# Higher-order statistics for DSGE models

**ONLINE APPENDIX**

**NOT FOR PUBLICATION**

---

## Contents

1. **Exact expressions for pruned state-space representation**
   1.1 State-space system of first-order approximation ........................................... 2
   1.2 State-space system of second-order approximation and pruning ........................ 3
   1.3 State-space system of third-order approximation and pruning ........................... 3

2. **Computation of product moments for extended innovations**
   2.1 First-order approximation .................................................................................. 4
   2.2 Second-order approximation .............................................................................. 5
   2.3 Third-order approximation .................................................................................. 7

---

*Preprint submitted to Econometrics and Statistics*  
*August 29, 2016*
1. Exact expressions for pruned state-space representation

This is based on the technical appendix of [Andreasen et al. (2014)].

First we derive some additional expressions:

\begin{align}
\dot{y}_{t+1}^r &= g_x \dot{x}_t^r + g_u u_{t+1} \\
\dot{y}_{t+1}^d &= g_x \dot{x}_t^d + 2G_{xu} (\dot{x}_t^r \otimes \dot{x}_t^r) + G_{xu} (u_{t+1} \otimes u_{t+1}) + g_{xu} \sigma^2 \\
\dot{x}_{t+1}^r &= h_x \dot{x}_t^r + \frac{1}{2} \left[ H_{xx} (\dot{x}_t^r \otimes \dot{x}_t^r) + 2H_{xu} (\dot{x}_t^r \otimes u_{t+1}) + H_{uu} (u_{t+1} \otimes u_{t+1}) + h_{xu} \sigma^2 \right] \\
\dot{x}_{t+1}^d &= h_x \dot{x}_t^d + H_{xx} (\dot{x}_t^r \otimes \dot{x}_t^r) + H_{xu} (\dot{x}_t^r \otimes u_{t+1}) + h_{xu} \sigma^2 \\
\dot{\xi}_{t+1}^r &= h_x \dot{\xi}_t^r + h_u \dot{u}_{t+1} \\
\dot{\xi}_{t+1}^d &= h_x \dot{\xi}_t^d + \frac{1}{2} \left[ H_{xx} (\dot{\xi}_t^r \otimes \dot{\xi}_t^r) + 2H_{xu} (\dot{\xi}_t^r \otimes u_{t+1}) + H_{uu} (u_{t+1} \otimes u_{t+1}) + h_{xu} \sigma^2 \right] \\
\dot{\xi}_{t+1}^d &= h_x \dot{\xi}_t^d + H_{xx} (\dot{\xi}_t^r \otimes \dot{\xi}_t^r) + H_{xu} (\dot{\xi}_t^r \otimes u_{t+1}) + h_{xu} \sigma^2 \\
\dot{x}_{t+1}^r &\otimes \dot{x}_{t+1}^r = (h_x \otimes h_u) (\dot{x}_t^r \otimes \dot{x}_t^r) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1} - \Gamma_{2u} + \Gamma_{2u}) \\
\dot{x}_{t+1}^d &\otimes \dot{x}_{t+1}^d = (h_x \otimes h_u) (\dot{x}_t^d \otimes \dot{x}_t^d) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1}) \\
\dot{\xi}_{t+1}^r &\otimes \dot{\xi}_{t+1}^r = (h_x \otimes h_u) (\dot{\xi}_t^r \otimes \dot{\xi}_t^r) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1} - \Gamma_{3u} + \Gamma_{3u}) \\
\dot{\xi}_{t+1}^d &\otimes \dot{\xi}_{t+1}^d = (h_x \otimes h_u) (\dot{\xi}_t^d \otimes \dot{\xi}_t^d) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1}) \\
\dot{\xi}_{t+1}^r &\otimes \dot{\xi}_{t+1}^d = (h_x \otimes h_u) (\dot{\xi}_t^r \otimes \dot{\xi}_t^d) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1}) \\
\dot{\xi}_{t+1}^d &\otimes \dot{\xi}_{t+1}^r = (h_x \otimes h_u) (\dot{\xi}_t^d \otimes \dot{\xi}_t^r) + (h_u \otimes h_u) (u_{t+1} \otimes u_{t+1}) \\
\end{align}

1.1. State-space system of first-order approximation

In a first-order approximation the system dynamics are captured by equations (1) and (4), we are therefore already working in a linear state-space system. That is, define \( z_t := \dot{x}_t^r \), \( y_t := \dot{y}_t + \ddot{y}_t \), \( \xi_{t+1} := u_{t+1} \), \( c := 0 \), \( d := 0 \), \( A := h_x \), \( B := h_u \), \( C := g_x \), and \( D := g_u \), then the equations can be rewritten as

\begin{align}
z_{t+1} &= c + Az_t + B\xi_{t+1} \\
y_{t+1} &= \ddot{y}_t + d + Cz_t + D\xi_{t+1}
\end{align}

Note that if \( u_t \) is Gaussian, \( \xi_t \) is clearly Gaussian as well.
1.2. State-space system of second-order approximation and pruning

In a second-order approximation the system dynamics are captured by equations (1), (2), (4), (5) and (7). To set up the pruned state-space system we define

\[
y_t = \hat{y}_t + \hat{y}_t^f + \tilde{y}, \quad z_t := \begin{pmatrix} \hat{x}_t^f \\ \hat{x}_t \\ \tilde{x}_t \\ \hat{x}_t^d \\ \tilde{x}_t^d \end{pmatrix}, \quad \xi_{t+1} := \begin{pmatrix} u_{t+1} \\ u_{t+1} \otimes u_{t+1} - \Gamma_{2u} \\ \hat{x}_t^f \\ \hat{x}_t \otimes \hat{x}_t \\ \tilde{x}_t \otimes \tilde{x}_t \\ \tilde{x}_t^d \otimes \tilde{x}_t^d \end{pmatrix}
\]

and

\[
A := \begin{pmatrix} h_x & 0 & 0 & 0 & 0 & 0 \\ 0 & h_x & 1 \frac{1}{2} H_{xx} & 0 & 0 & 0 \\ 0 & 0 & h_x \otimes h_x & 0 & 0 & 0 \\ \frac{1}{2} \frac{1}{2} H_{xx} & 0 & 0 & h_x & H_{xx} & \frac{1}{2} H_{xx} \\ h_x \otimes \frac{1}{2} h_x \sigma^2 & 0 & 0 & h_x \otimes h_x & h_x \otimes \frac{1}{2} H_{xx} & 0 \\ 0 & 0 & 0 & 0 & h_x \otimes h_x & h_x \otimes h_x \end{pmatrix}, \quad B := \begin{pmatrix} h_u & 0 & 0 & 0 & 0, \hline 0 & \frac{1}{2} H_{uu} & 0 & 0 & 0, \hline 0 & h_u \otimes h_u & h_x \otimes h_u & h_u \otimes h_x \end{pmatrix}, \quad D := \begin{pmatrix} h_x & \frac{1}{2} G_{uu} & \Sigma_{G} \\ 0 & G_{uu} & 0 \end{pmatrix}
\]

The system can thus be rewritten as a linear state-space representation

\[
z_{t+1} = c + Az_t + Bu_{t+1} + \xi_{t+1}
\]

\[
y_{t+1} = \hat{y} + d + Cz_t + D\xi_{t+1}
\]

Note that even if \( u_t \) is Gaussian, \( \xi_t \) is clearly non-Gaussian.

1.3. State-space system of third-order approximation and pruning

In a third-order approximation the system dynamics are captured by equations (1), (2), (3), (4), (5), (6), (7), (8) and (9). To set up the pruned state-space system we define

\[
y_t = \hat{y}_t^f + \hat{y}_t^g + \tilde{y}_d + \tilde{y}, \quad z_t := \begin{pmatrix} \hat{x}_t^f \\ \hat{x}_t^g \\ \hat{x}_t^d \\ \tilde{x}_t^d \end{pmatrix}, \quad \xi_{t+1} := \begin{pmatrix} u_{t+1} \\ u_{t+1} \otimes u_{t+1} - \Gamma_{2u} \\ \hat{x}_t^f \\ \hat{x}_t^g \\ \hat{x}_t^d \\ \tilde{x}_t^d \otimes \tilde{x}_t^d \end{pmatrix}
\]

and

\[
A := \begin{pmatrix} h_x & 0 & 0 & 0 & 0 & 0 \\ 0 & h_x & \frac{1}{2} H_{xx} & 0 & 0 & 0 \\ 0 & 0 & h_x \otimes h_x & 0 & 0 & 0 \\ \frac{1}{2} \frac{1}{2} H_{xx} & 0 & 0 & h_x & H_{xx} & \frac{1}{2} H_{xx} \\ h_x \otimes \frac{1}{2} h_x \sigma^2 & 0 & 0 & h_x \otimes h_x & h_x \otimes \frac{1}{2} H_{xx} & 0 \\ 0 & 0 & 0 & 0 & h_x \otimes h_x & h_x \otimes h_x \end{pmatrix};
\]
and even if $u_2$.

2.1. First-order approximation

duplicate elements. Denote with $n$ the duplication matrix $D_P$ $T_P$ $\theta$ independent of $(2005)$ for an example and more details.

Given the number of shocks $n$, the system can thus be rewritten as a linear state-space representation

To compute the product-moments of $\xi$ symbolically we therefore use the following procedure in Matlab given the number of shocks $n_u$ and the order of product moments $k=2,3,4$.

1. Define $u_{t+1} = (u_{t+1,1}, \ldots, u_{t+1,n_u})'$ and $\Sigma_u = [\text{sig}_{ij}]_{nu \times nu}$ symbolically with $i, j = 1, \ldots, n_u$.
2. Get all integer permutations of $[i_1, i_2, \ldots, i_{n_\xi}]$ that sum up to $k$, with $i_j = 1, \ldots, k$ and $j = 1, \ldots, n_\xi$.

Sort them in the ordering of Meijer (2005).

3. For each permutation $[i_1, i_2, \ldots, i_{n_\xi}]$, evaluate symbolically

$$E \left[ (\xi_{1,t})^{i_1} \cdot (\xi_{2,t})^{i_2} \cdot \cdots \cdot (\xi_{n_\xi,t})^{i_{n_\xi}} \right]$$

and store it in the vector $\bar{M}_{k,\xi}$. 

Note that even if $u_t$ is Gaussian, $\xi_t$ is clearly non-Gaussian, since it’s higher-order cumulants are nonzero.

2. Computation of product moments for extended innovations

2.1. First-order approximation

Given a first-order approximation, the innovations are defined as the $n_\xi \times 1$ vector $\xi_{t+1} = u_{t+1}$ with $n_\xi = n_u$ elements. We are interested in product moments $M_{2,\xi} := E(\xi_t \otimes \xi_t)$, $M_{3,\xi} := E(\xi_t \otimes \xi_t \otimes \xi_t)$ and $M_{4,\xi} := E(\xi_t \otimes \xi_t \otimes \xi_t \otimes \xi_t)$ with $n_\xi^2$, $n_\xi^3$ and $n_\xi^4$ elements, respectively. These, however, contain many duplicate elements. Denote with $\bar{M}_{k,\xi}$ the unique elements of $M_{k,\xi}$, we have the following relationships:

$$M_{2,\xi} = D_{P_{n_\xi}} \cdot \bar{M}_{2,\xi}, \quad M_{3,\xi} = T_{P_{n_\xi}} \cdot \bar{M}_{3,\xi}, \quad M_{4,\xi} = Q_{P_{n_\xi}} \cdot \bar{M}_{4,\xi},$$

with the duplication matrix $D_{P_{n_\xi}}$ defined by Magnus & Neudecker (1999), and the triplication matrix $T_{P_{n_\xi}}$ and quadruplication matrix $Q_{P_{n_\xi}}$ similarly defined by Meijer (2005). Note that these matrices are independent of $\theta$ and their Moore-Penrose-Inverse always exists, e.g. $(Q_{P_{n_\xi}} P_{n_\xi})^{-1} Q_{P_{n_\xi}} \cdot M_{4,\xi} = \bar{M}_{4,\xi}$. Further, $D_{P_{n_\xi}}$, $T_{P_{n_\xi}}$ and $Q_{P_{n_\xi}}$ are constructed such that there is a unique ordering in $\bar{M}_{k,\xi}$, see Meijer (2005) for an example and more details.
The expressions we get in step 3 contain terms of the form
\[ \text{const} \cdot E[(u_{1,t+1})^{i_1} \cdot (u_{2,t+1})^{i_2} \cdots (u_{n_u,t+1})^{i_{n_u}}], \]
that is joint product moments of the elements of \( u_{t+1} \). Given a function that evaluates the moment structure of \( u_{t+1} \) either analytically or numerically, we are able to calculate these terms individually and save them into script files. Note, that these computations need only to be done once for a model, after that we simply evaluate the script files numerically given model parameters \( \theta \). Our code can evaluate product moments from the Gaussian as well as Student-t distribution.

**Normal distribution.** In the case that \( u_t \) is normally distributed, the joint product moments are functions of the variances and covariances in \( \Sigma \) and can be computed analytically. To this end, we use the very efficient method and Matlab function of Kan (2008) to derive these joint product moments symbolically. The cumulants can then be computed as outlined in the paper.

**Student’s t distribution.** In the case that \( u_t \) is Student-t distributed with \( v \) degrees of freedom, we rewrite \( u_t \) in terms of a Inverse-Gamma distributed variable \( W = v^{-1/2} \sim IGAM(v/2, v/2) \), and a normally distributed variable \( \varepsilon_t \sim N(0, \Sigma), u_t = v^{-1/2} \varepsilon_t \) (similar to Kotz & Nadarajah (2004) or Roth (2013)). Since \( W \) and \( \varepsilon_t \) are independent, we have \( E(u_t u'_j) = E(W)|E(\varepsilon_t \varepsilon'_i) = v^{-1} \Sigma \). Whereas all odd product moments of \( u_t \) are zero, the even product moments \((n = \sum_{j=1}^{n_u} i_u) \) is an even number) are given by
\[ E[(u_{1,t})^{i_1} \cdot (u_{2,t})^{i_2} \cdots (u_{n_u,t})^{i_{n_u}}] = E[W^2] \cdot E[(\varepsilon_{1,t})^{i_1} \cdot (\varepsilon_{2,t})^{i_2} \cdots (\varepsilon_{n_u,t})^{i_{n_u}}]. \]
The first term is equal to \( E[W^k] = (v/2)^k \frac{(v/2-1)!}{(v/2-k)!} \) and since \( \varepsilon_t \) is multivariate normal, we can use Kan (2008)’s procedure and Matlab function for the second product. The cumulants can then be computed as outlined in the paper.

2.2. Second-order approximation

Given a second-order approximation, the innovations are defined as the \( n_\xi \times 1 \) vector
\[ \xi_{t+1} = (u'_{t+1} \cdot (u_{t+1} \otimes u_{t+1} - \text{vec}(\Sigma))' \cdot (x'_t \otimes u_{t+1})' \cdot (u_{t+1} \otimes x'_t)'), \]
with \( n_\xi = n_u + n_u^2 + 2 n_u n_v \) elements. We are interested in product moments \( M_{2,\xi} := E(\xi_t \otimes \xi_t), M_{3,\xi} := E(\xi_t \otimes \xi_t \otimes \xi_t) \) and \( M_{4,\xi} := E(\xi_t \otimes \xi_t \otimes \xi_t \otimes \xi_t) \) with \( n_\xi^2, n_\xi^3 \) and \( n_\xi^4 \) elements, respectively. In order to compute these objects efficiently, we first reduce the dimension of \( \xi_t \), since it has some duplicate elements. That is, we compute product-moments for the \( n_\xi = n_u + n_u(n_u + 1)/2 + n_u n_v \) vector
\[ \bar{\xi}_{t+1} := (u'_{t+1} \cdot (DP^+_{n_u}(u_{t+1} \otimes u_{t+1} - \text{vec}(\Sigma))' \cdot (x'_t \otimes u_{t+1})')), \]
since
\[ \xi_t = \begin{pmatrix} I & 0 & 0 \\ 0 & DP_{n_u} & 0 \\ 0 & 0 & I \end{pmatrix} \bar{\xi}_t := F_{k \xi} \cdot \bar{\xi}_t, \]
with \( DP^+_{n_u} \) being the Moore-Penrose-Inverse of the duplication matrix \( DP_{n_u} \) and \( K_{n_u,n_u} \) the commutation matrix such that \( K_{n_u,n_u}(x'_t \otimes u_{t+1}) = (u_{t+1} \otimes x'_t) \). Then we have
\[ M_{k,\xi} := [\otimes_{j=1}^{k} F_{k \xi}] \cdot M_{k,\bar{\xi}} \]
denoting the k-th (k=2,3,4)-order product moment of \( \tilde{\xi} \). Since \( \otimes_{j=1}^{k} F_{\xi} \) does not change with \( \theta \), we can focus on \( M_{k,\tilde{\xi}} \). \( M_{k,\tilde{\xi}} \), however, contains also many duplicate elements. Denote with \( \tilde{M}_{k,\tilde{\xi}} \) the unique elements of \( M_{k,\tilde{\xi}} \), we have the following relationships:

\[
M_{2,\tilde{\xi}} = DP_{n_\xi} \cdot \tilde{M}_{2,\tilde{\xi}}, \quad M_{3,\tilde{\xi}} = TP_{n_\xi} \cdot \tilde{M}_{3,\tilde{\xi}}, \quad M_{4,\tilde{\xi}} = QP_{n_\xi} \cdot \tilde{M}_{4,\tilde{\xi}},
\]

with the duplication matrix \( DP_{n_\xi} \) defined by [Magnus & Neudecker (1999)], and the triplication matrix \( TP_{n_\xi} \) and quadruplication matrix \( QP_{n_\xi} \) similarly defined by [Meijer (2005)]. Note that these matrices are independent of \( \theta \) and their Moore-Penrose-Inverse always exists, e.g. \((QP_{n_\xi} QP_{n_\xi})^{-1} QP_{n_\xi} \cdot M_{4,\tilde{\xi}} = \tilde{M}_{4,\tilde{\xi}}\). Further, \( DP_{n_\xi} \), \( TP_{n_\xi} \) and \( QP_{n_\xi} \) are constructed such that there is a unique ordering in \( M_{k,\tilde{\xi}} \), see [Meijer (2005)] for an example and more details.

To compute the product-moments of \( \tilde{\xi} \) symbolically we therefore use the following procedure in Matlab given the number of shocks \( n_u \), the number of state variables \( n_x \) and the order of product moments \( k=2,3,4 \).

1. Define \( u_{t+1} = (u_{t+1,1}, \ldots u_{t+1,n_u})' \), \( x_{t} = (x_{t,1}, \ldots x_{t,n_x})' \) and \( \Sigma_u = [\text{sig}_{ij}]_{nu \times nu} \) symbolically with \( i, j = 1, \ldots, n_u \). Set up

\[
\tilde{\xi}_t = (u_{t,1}DP_{n_\xi}^{-1}(u_{t+1} \otimes u_{t+1} - \text{vec}(\Sigma)))', (x_{t} \otimes u_{t+1})'.
\]

2. Get all integer permutations of \([i_1, i_2, \ldots, i_{n_\xi}]\) that sum up to \( k \), with \( i_j = 1, \ldots, k \) and \( j = 1, \ldots, n_\xi \). Sort them in the ordering of [Meijer (2005)].

3. For each permutation \([i_1, i_2, \ldots, i_{n_\xi}]\) evaluate symbolically

\[
E[(\tilde{\xi}_{1,t})^{i_1} \cdot (\tilde{\xi}_{2,t})^{i_2} \cdots (\tilde{\xi}_{n_\xi,t})^{i_{n_\xi}}]
\]

and store it in the vector \( \tilde{M}_{k,\xi} \).

4. Optionally: Use Matlab’s \texttt{unique} function to further reduce the dimension of \( \tilde{M}_{k,\xi} \).

The expressions we get in step 3 contain terms of the form

\[
\text{const} \cdot E[(u_{1,t+1})^{i_1} \cdot (u_{2,t+1})^{i_2} \cdots (u_{n_u,t+1})^{i_{nu}}] \cdot E[(x_{1,t}^{i_1} \cdot (x_{2,t}^{i_2})^{i_2} \cdots (x_{n_x,t}^{i_{n_x}})^{i_{n_x}}],
\]

that is joint product moments of the elements of \( u_{t+1} \) and \( x_{t} \) (keeping in mind that \( x_{t} \) and \( u_{t+1} \) are independent due to the temporal independence of \( u_t \)). For instance, for \( n_u = n_x = 1 \), the third-order product moment of \( \tilde{\xi}_t \) is equal to

\[
\tilde{M}_{3,\xi} = \text{vec} \left( E \left[ \begin{array}{c}
u^3 \\ u^3 \\ x u^2 \\ -\sigma_u^2 u^2 \\ \sigma_u^2 u - 2 \sigma_u^2 u^3 + u^5 \\ -u^2 + 3 \sigma_u^4 u^2 - 3 \sigma_u^2 u^4 + u^6 \\ \sigma_u^4 u - 2 \sigma_u^2 u^3 + x u^5 \\ x \sigma_u^4 u - 2 x \sigma_u^2 u^3 + x u^5 \\
\end{array} \right] \right)
\]

where we dropped sub- and superscripts and \( E(u^2) = \sigma_u^2 \). Given a function that evaluates the moment structure of \( x_t \) and \( u_{t+1} \) either analytically or numerically, we are able to calculate these terms individually and save them into script files. Note, that these computations need only to be done once for a model, after that we simply evaluate the script files numerically given model parameters \( \theta \). Our code can evaluate product moments from the Gaussian as well as Student-t distribution.

\footnote{Actually \( \tilde{M}_{k,\xi} \) has some further duplicate terms for \( n_u, n_x > 1 \) due to higher-order cross terms of \( u_{t+1} \) and \( x_{t} \), which we can further reduce using indices from the unique function of Matlab.}
Normal distribution. In the case that \( u_t \) is normally distributed, \( x^f_t \) is also Gaussian with covariance matrix \( \Sigma_x \). Therefore,

\[
\begin{pmatrix}
  u_{t+1} \\
  x^f_t 
\end{pmatrix} 
\sim \mathcal{N}\left( \begin{pmatrix} 0 \\ \Sigma 
\end{pmatrix}, \begin{pmatrix} 0 & \Sigma_x \\
 0 & \Sigma_x 
\end{pmatrix} \right)
\]

is multivariate normal. All joint product moments are therefore functions of the variances and covariances in \( \Sigma \) and \( \Sigma_x \) and can be computed analytically. To this end, we use the very efficient method and Matlab function of [Kan (2008)](Kan2008) to derive these joint product moments symbolically. For our example with \( n_u = n_x = 1 \) and Gaussian \( u_t \), we get the unique entries

\[
\begin{align*}
\tilde{M}_{2, 1} &= [\sigma^2_i, 0, 0, 2\sigma^4_i, 0, \sigma^6_i, \sigma^2_i] \\
\tilde{M}_{3, 1} &= [0, 2\sigma^4_i, 0, 0, 0, 8\sigma^6_i, 0, 2\sigma^4_i, \sigma^2_i] \\
\tilde{M}_{4, 1} &= [3\sigma^4_i, 0, 0, 10\sigma^6_i, 0, 3\sigma^4_i, \sigma^2_i, 0, 0, 0, 60\sigma^8_i, 0, 10\sigma^6_i, \sigma^2_i, 0, 9\sigma^4_i, \sigma^2_i] 
\end{align*}
\]

where \( E(x^{f^2}_t) = \sigma^2_x \). The cumulants can then be computed as outlined in the paper. Since the third-order cumulant of a Gaussian process must be zero, we now see, that \( \xi_t \) is clearly non-Gaussian, since its third-order cumulant is different from zero, even if the underlying distribution for \( u_t \) is Gaussian.

Student’s t distribution. In the case that \( u_t \) is Student-t distributed with \( v \) degrees of freedom, we rewrite \( u_t \) in terms of an Inverse-Gamma distributed variable \( W = v^{-1/2} \sim IGAM(v/2, v/2) \), and a normally distributed variable \( \varepsilon_t \sim N(0, \Sigma) \), \( u_t = v^{-1/2} \varepsilon_t \) (similar to [Kotz & Nadarajah (2004)](Kotz2004) or [Roth (2013)](Roth2013)). Since \( W \) and \( \varepsilon_t \) are independent, we have \( E(u_t u'_t) = E(W)E(\varepsilon_t \varepsilon'_t) = \frac{v}{v-2} \Sigma \). Whereas all odd product moments of \( u_t \) are zero, the even product moments \( (n = \sum_{j=1}^{\infty} i_{uj} \) is an even number) are given by

\[
E[(u_{1,t})^{i_{u,1}} \cdot (u_{2,t})^{i_{u,2}} \cdots (u_{n_u,t})^{i_{u,n_u}}] = E[W^{\frac{v}{2}}] \cdot E[(\varepsilon_{1,t})^{i_{u,1}} \cdot (\varepsilon_{2,t})^{i_{u,2}} \cdots (\varepsilon_{n_u,t})^{i_{u,n_u}}].
\]

The first term is equal to \( E[W^k] = (v/2)^k \frac{v/2-1}{v/2-k} \cdots \frac{v/2-k}{v/2-2} \) and since \( \varepsilon_t \) is multivariate normal, we can use [Kan (2008)](Kan2008)’s procedure and Matlab function for the second product. Similar arguments apply to the product moments of \( x^f_t \), for instance the variance is given by

\[
\text{vec}(\Sigma_x) = E[x^f_t \otimes x^f_t] = E[W] \cdot (I_n \otimes h_x)^{-1} (h_u \otimes h_u) \cdot E[\varepsilon_t \otimes \varepsilon_t].
\]

Thus, odd product moments are also zero, whereas even product moments can also be computed symbolically by [Kan (2008)](Kan2008)’s procedure and Matlab function, however, adjusted for \( E[W^{n/2}] \). The cumulants can then be computed as outlined in the paper.

2.3. Third-order approximation

Given a third-order approximation, the innovations are defined as the \( n_x \times 1 \) vector

\[
\xi_{t+1} :=
\begin{pmatrix}
  u_{t+1} \\
  u_{t+1} \otimes u_{t+1} - \Gamma_{2u} \\
  \dot{x}^f_t \otimes u_{t+1} \\
  u_{t+1} \otimes \dot{x}^f_t \\
  \dot{x}^f_t \otimes \dot{x}^f_t \\
  \dot{x}^f_t \otimes u_{t+1} \otimes \dot{x}^f_t \\
  \dot{x}^f_t \otimes u_{t+1} \otimes \dot{x}^f_t \\
  u_{t+1} \otimes \dot{x}^f_t \otimes u_{t+1} \\
  u_{t+1} \otimes \dot{x}^f_t \otimes u_{t+1} \\
  u_{t+1} \otimes u_{t+1} \otimes \dot{x}^f_t \\
  u_{t+1} \otimes u_{t+1} \otimes \dot{x}^f_t \\
  u_{t+1} \otimes u_{t+1} \otimes u_{t+1} \\
  u_{t+1} \otimes u_{t+1} \otimes u_{t+1} \\
  u_{t+1} \otimes u_{t+1} \otimes u_{t+1} \\
  u_{t+1} \otimes u_{t+1} \otimes u_{t+1} + \Gamma_{3u}
\end{pmatrix}
\]
with \( n_\xi = n_u + n_u^2 + 2n_xn_u + 2n_xn_u + 3n_x^2n_u + 3n_xn_u^2 + n_u^2 \) elements. We are interested in product moments 
\( M_{2,\xi} := E(\xi_2 \otimes \xi_2) \), \( M_{3,\xi} := E(\xi_3 \otimes \xi_2 \otimes \xi_2) \) and 
\( M_{4,\xi} := E(\xi_4 \otimes \xi_2 \otimes \xi_2 \otimes \xi_2) \) with \( n_\xi^2 \), \( n_\xi^3 \) and \( n_\xi^4 \) elements, respectively. In order to compute these objects efficiently, we first reduce the dimension of \( \xi \) with \( \bar{\xi} = \xi + n \xi^2 + n \xi^3 + n(n + 1)(n + 2) / 6 \) vector

\[
\hat{\xi}_{t+1} := \begin{pmatrix}
\frac{u_t + 1}{u_t + 1} & DP_{n_u}(u_t + 1 \otimes u_t + 1 - \Gamma_{2u}) \\
\bar{x}_n \otimes u_t + 1 & \bar{x}_n \otimes u_t + 1 \\
\bar{x}_n \otimes u_t + 1 & \bar{x}_n \otimes u_t + 1 \\
TP_{n_u}(u_t + 1 \otimes u_t + 1 \otimes u_t + 1 - \Gamma_{3u})
\end{pmatrix}
\]

given that

\[
F_\xi = \begin{bmatrix}
I_u & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & DP_u & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_{ru} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & K_{ux} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I_{ru} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & K_{ux} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & I_{ux} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & DP_x \otimes I_u \\
0 & 0 & 0 & 0 & (I_x \otimes K_{ux})(DP_x \otimes I_u) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & (K_{ux} \otimes I_x)(I_x \otimes K_{ux})(DP_x \otimes I_u) & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & (K_{ux} \otimes I_x)(I_x \otimes K_{ux})(DP_x \otimes I_u) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & I_u \otimes K_{ux}(I_x \otimes DP_u) & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & (I_u \otimes K_{ux})(I_x \otimes DP_u) \otimes I_u & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & TP_u
\end{bmatrix}
\]

since

\[
\xi_t = F_\xi \cdot \hat{\xi}_t
\]

and with \( DP_{n_u}^+ \) being the Moore-Penrose-Inverse of the duplication matrix \( DP_{n_u} \), \( TP_{n_u}^+ \) being the Moore-Penrose-Inverse of the triplication matrix \( TP_{n_u} \) and \( K_{ux,n_u} \) the commutation matrix such that \( K_{ux,n_u}(x_t \otimes u_t + 1) = (u_t + 1 \otimes x_t) \). Then we have

\[
M_{k,\xi} := [\otimes_{j=1}^{k-1} F_\xi] \cdot M_{k,\xi}
\]

denoting the k-th \((k=2,3,4)\)-order product moment of \( \hat{\xi}_t \). Since \([\otimes_{j=1}^{k-1} F_\xi] \) does not change with \( \theta \), we can focus on \( M_{k,\xi} \). \( M_{k,\xi} \) however, contains also many duplicate elements. Denote with \( \bar{\xi}_k, \xi_k \) the unique elements of \( M_{k,\xi} \); we have the following relationships:

\[
M_{2,\xi} = DP_{n_\xi} \cdot \bar{\xi}_2, \quad M_{3,\xi} = TP_{n_\xi} \cdot \bar{\xi}_3, \quad M_{4,\xi} = QP_{n_\xi} \cdot \bar{\xi}_4
\]

with the duplication matrix \( DP_{n_\xi} \) defined by \cite{Magnus1999}, and the triplication matrix \( TP_{n_\xi} \) and quadruplication matrix \( QP_{n_\xi} \) similarly defined by \cite{Meijer2005}. Note that these matrices are 2Actually \( \bar{\xi}_k \) has some further duplicate terms for \( n_u, n_x > 1 \) due to higher-order cross terms of \( u_t + 1 \) and \( x_t \), which we can further reduce using indices from the unique function of Matlab.
independent of \( \theta \) and their Moore-Penrose-Inverse always exists, e.g. \((QP_n \tilde{\xi} QP_n \tilde{\xi})^{-1} QP_n \tilde{\xi} \cdot M_{4,\tilde{\xi}}^g = \tilde{M}_{4,\tilde{\xi}}\). Further, \(DP_n \tilde{\xi}, TP_n \tilde{\xi}\) and \(QP_n \tilde{\xi}\) are constructed such that there is a unique ordering in \(\tilde{M}_{k,\tilde{\xi}}\); see (Meijer, 2005) for an example and more details.

The product-moments of \(\tilde{\xi}_t\) can thus be computed symbolically as outlined in the second-order approximation.

**References**


