Estimation of the VCZAR model for irregularly sampled time series

Granville Tunnicliffe Wilson*

July 29, 2019

Keywords: Irregularly sampled time series, VCZAR model, Gaussian estimation

1 Models and data

In this document we explain the usage of the function `VCZARestimate.m` for estimating the parameters of a VCZAR model for irregularly sampled multivariate time series as presented in Tunnicliffe Wilson et al. (2015). The VCZAR model for a zero-mean stationary multivariate continous time series $x(t)$ may be expressed in two different but equivalent forms. These two forms differ in their interpretation and parametrization so that one form may have advantages over the other in different contexts. Apart from this, the estimation scheme is the same for both forms of model.

The predictive form of the model is given using operator notation in (7.30) and (7.31) of the book, which are reproduced here:

$$s\,x(t) = \xi(Z)\,x(t) + e(t) \tag{1}$$

where

$$\xi(Z) = \xi_1 + \xi_2\,Z + \cdots + \xi_p\,Z^{p-1}. \tag{2}$$

We refer to $e(t)$ as the innovation term, the formal derivative of $W(t)$, a multivariate Wiener process with incremental properties $\mathrm{Var}\{dW(t)\} = V_e\,dt$. The parameters of this model are therefore the matrix coefficients $\xi_1, \ldots, \xi_p$ and the innovation variance matrix V_e.

The natural form of the model is given using operator notation in (7.32), (7.33) and (7.34) of the book, which are reproduced here:

$$\varphi(Z)x(t) = n(t), \tag{3}$$

where

$$\varphi(Z) = I - \varphi_1 Z - \varphi_2 Z^2 - \cdots - \varphi_p Z^p, \tag{4}$$

and the *disturbance* term $n(t)$ follows a CAR(1) process with scalar coefficient κ:

$$(s+\kappa)n(t) = \varepsilon(t). \tag{5}$$

*Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, UK.

The innovation term $\varepsilon(t)$ is the formal derivative of a multivariate Wiener process with differential variance $V_\varepsilon dt$ related to the variance V_n of $n(t)$ by $V_\varepsilon = 2\,\kappa\,V_n$. The parameters of this model are therefore the matrix coefficients $\varphi_1, \ldots, \varphi_p$ and the innovation variance matrix V_ε.

In both models the rate parameter κ, which is used to define the continuous time operator Z, besides appearing explicitly in the disturbance model (5), is specified, and not estimated by the function.

The function fits the specified model to observed values y_j related to the component series $x_i(t)$ at the sampling time t_j, with the possible inclusion of additive regressors and observation noise, by

$$y_j = x_i(t_j) + \sum_{k=1}^{K_i} \alpha_{i,k} A_{i,k}(t_j) + w_{i,j}. \tag{6}$$

We here use j to uniquely index the observations. This allows for two distinct observations y_j and $y_{j'}$ with $j \neq j'$ to be made at the same times $t_j = t_{j'}$. The regressors are also here assumed to be continuous functions $A_{i,k}(t)$ of time t, and may differ for the different series $x_i(t)$.

The observation noise terms $w_{i,j}$ are uncorrelated at different times. Correlation is only considered between noise terms of distinct series observed at the same time. The noise variances and covariances are specified by the matrix V_w.

In the next section we specify the inputs and outputs of `VCZARestimate.m`, relating them to the quantities described above, and to the iterative estimation scheme described in the technical document *GaussianEstimation.pdf* with title *Non-linear Gaussian estimation from orthogonal residuals*. In subsequent sections we document the subsidiary functions which are called within `VCZARestimate.m` and describe the calling structure. Finally we describe the modifications in the closely related function `VCZARestimate_reml.m` which uses the residual maximum likelihood criterion for estimation in the presence of regressors.

2 Input and output quantities of `VCZARestimate.m`

The usage of the function is

```
results=VCZARestimate...
(y,p,kappa,parind,obsnoise,xreg,struc,initpar,constraints,controlpar,strategy)
```

where the output `results` is a cell structure with 17 elements including the parameter estimates and related quantities. We will first describe the input arguments, presenting the MATLAB help information on each, together with any further explanatory details. We then, similarly, describe the elements of the output structure.

2.1 The input quantities

The irregularly sampled time series

```
y       := continuous m-vector time series dataset
           this is a structure of m cells, each successive cell containing
           a continuous time series composed of two columns:
           the observation time, and the value observed at that time.
```

Simple example:

```
x=[0.3 213.9
   2.7 159.6
   3.6 176.2]
z=[0.0 5.46
   1.9 3.32
   3.1 4.64
   5.1 2.98]
y={x,z}
```

The model order p

```
p       := CZAR model order
```

The model decay rate parameter κ

```
kappa   := CZAR model decay rate
```

The indicator of the model form

```
parind  : parameterisation indicator: 1 for canonical, 2 for direct
```

The indicator of the observation noise variance structure and presence

```
obsnoise:= one for diagonal form, two for full, zero to omit
```

Note: the indicator is numeric; we use text here to avoid confusing zero and the letter O.

The values of the regressors at the observation times

```
xreg    := explanatory regression variables
           this is a cell structure, each successive cell corresponding
           to those of the time series, and containing a regression matrix
           of explanatory variables for that time series, at the same
           observation times. Set {} if none.
```

A simple example relating to that of the time series above, with trend regression for `x` and a constant level for `z`:

```
xtrend=...
  [1 213.9
   1 159.6
   1 176.2]
const=...
  [1
   1
   1
   1 ]
xreg={x,z}
```

The specification of the error variance with the option of a structural model form

```
struc   := zero to omit, one for structural error, two for full structural model
```

Note: the indicator is numeric; we use text here to avoid confusing zero and the letter O.
Explanation:
If `struc=0` the error variance matrices V_e and V_ε are estimated for, respectively, the predictive and natural forms of the model.
If `struc=1` the predictive model error in (1) is modified as

$$\xi_0\, e(t) = e_S(t) \tag{7}$$

where the structural error matrix ξ_0 has unit diagonals and the variance matrix V_{e_S} of the structural error $e_S(t)$ is constrained to be diagonal.
If `struc=2` the predictive model (1) is modified as

$$s\,\xi_0\, x(t) = \xi(Z)\, x(t) + e_S(t) \tag{8}$$

where the structural error matrix ξ_0 has unit diagonals and the variance matrix V_{e_S} of the structural error $e_S(t)$ is constrained to be diagonal.
For the purpose of model estimation the likelihoods of the structural forms are found by mapping them to the standard form by dividing through by ξ_0. Thus

$$V_e = \xi_0^{-1} V_{e_S} \left(\xi_0^{-1}\right)' \quad \text{and} \quad \xi_0^{-1}\xi_i \to \xi_i. \tag{9}$$

Similarly for the natural form of model, if `struc=1` the error term is modified as

$$\varphi_0\, \varepsilon(t) = \varepsilon_S(t). \tag{10}$$

where φ_0 has unit diagonals and the variance matrix V_{ε_S} of $\varepsilon_S(t)$ is constrained to be diagonal.
And for the natural form of model, if `struc=2`, the model operator becomes

$$\varphi(Z) = \varphi_0 - \varphi_1 Z - \varphi_2 Z^2 - \cdots - \varphi_p Z^p, \tag{11}$$

where φ_0 has unit diagonals and the variance matrix V_{ε_S} of $\varepsilon_S(t)$ is constrained to be diagonal.
For the purpose of model estimation the likelihoods of the structural forms are found by mapping them to the standard form by dividing through by φ_0. Thus

$$V_\varepsilon = \varphi_0^{-1} V_{\varepsilon_S} \left(\varphi_0^{-1}\right)' \quad \text{and} \quad \varphi_0^{-1}\varphi_i \to \varphi_i. \tag{12}$$

For the structural model forms the further parameters ξ_0 and φ_0 are estimated for the predictive and natural model forms respectively. The estimated structural variance parameters V_{e_S} and V_{ε_S} have zero-valued off-diagonal matrix elements.

The initial model parameter values which may be set for iterative estimation

```
initpar := values to initialise non-linear model parameters. Include only the
       parameter groups which are to be estimated, giving a structure of two,
       three, four or five cells described below and in that order: if set to {}
       or any cell set to [], the following defaults are used as appropriate:
       VCZARcoeffinit : inital values of VCZAR coefficient array
                         if [], set to zeros(m,m,p) (parameter group I)
       VCZARSiginit : initial values of disturbance variance matrix of the model
                      if struc>0 all off-diagaonls are reset to zero
                      if [], set to a diagonal function of y (parameter group II)
       NoiseSiginit : initial values of the vector of the diagonal observation noise
                      variance matrix if obsnoise==1 or the full variance matrix if
                      obsnoise==2; if [], set to zero vector or matrix.
                      Omit if obsnoise==0. (parameter group III)
       xregcoeffinit: initial values of regression coefficients as single vector
                      If [] set to zeros (parameter group IV)
                      Not set if no regressors present
       Strucoeffinit: initial values of the structural model matrix. Diagonals are
                      reset to ones. If [] set to unit diagonal matrix.
                      Omit if struc==0. (parameter group V)
```

A simple example for a model of dimension 2 and order 2:

```
VCZARcoeffinit(:,:,1)=...
[0.5, 0.2
 0.0, 0.8]
VCZARcoeffinit(:,:,2)=...
[0.1, 0.0
 0.0, 0.0]
VCZARSiginit=...
[4.3  0.0
 0.0  2.5]
initpar={VCZARcoeffinit,VCZARSiginit}
```

The matrix element `VCZARcoeffinit(i,j,k)` corresponds to the model coefficient element $(\xi_k)_{ij}$ or $(\varphi_k)_{ij}$ for the predictive and natural model forms respectively.

The matrix element `VCZARSiginit(i,j)` corresponds to the model disturbance term variance element $(V_e)_{ij}$ or $(V_\varepsilon)_{ij}$ for the predictive and natural model forms respectively, except for both structural model forms they correspond to $(V_{e_S})_{ij}$ or $(V_{\varepsilon_S})_{ij}$

With the setting of the indicator `obsnoise=0`, `NoiseSiginit` is not included in the structure `initpar`.

With the setting of the indicator `obsnoise=2`, the matrix element `NoiseSiginit(i,j)` corresponds to the observation error model variance element $(V_w)_{ij}$

With the setting of the indicator `obsnoise=1`, the vector element `NoiseSiginit(i)` corresponds to the observation error model diagonal variance matrix element $(V_w)_{ii}$.

With the parameter setting `xreg={}`, `xregcoeffinit` is not included in the structure `initpar`, else it is a vector with elements that are the initial values of the coefficients of all the regression vectors, in sequence, in the structure `xreg`.

With the parameter settings `struc=0`, `Strucoeffinit` is not included in the structure `initpar`.

With the parameter settings `struc=1` and `struc=2`, the matrix element `Strucoeffinit(i,j)` corresponds to the structural model coefficient element $(\xi_0)_{ij}$ or $(\varphi_0)_{ij}$ for the predictive and natural model forms respectively.

The specification of constraints that coefficients retain their initial settings

```
constraints  := cell structure comprising a parameter constraint vector
                for each group of model parameters I, II, III, IV, V
                value of one to estimate, of zero to retain initial setting.
                The vector applies to matrix elements row by row
                (last index varying most rapidly), but only
                to the upper half of a general variance matrix, to the
                diagonal elements of a structural disturbance variance
                matrix, and to the non-diagonal elements of the
                structural dependence matrix.
                If set to {} or any cell set to [] the default interpretation is used
                as appropriate. That is: all parameters are estimated except that
                if struc>0 the default is that only the structural model matrix upper
                triangular elements are estimated and the diagonals of the
                disturbance variance matrix.
```

Simple example of Constraint Vectors corresponding to the model above with regressors but no observation noise

```
VCZARcoeffCV=[1,1,1,0,0,0,1,1]
VCZARSigCV=[1,0,1]
xregcoeffCV=[]
constraints={VCZARcoeffCV,VCZARSigCV,xregcoeffCV}
```

This feature is useful for effectively removing terms from the model by constraining the corresponding coefficients to retain the initial values of zero.

An optional feature to tune various general aspects of the iterative estimation

```
controlpar  := search control parameter vector with elements as below;
         if truncated, defaults are set:
         maxiter  : the maximum number of iterations
         convcrit : the convergence criterion for deviance reduction,
         monind   : monitoring indicator:
                  : 0 for no monitoring
                  : 1 for dev at main iteration
                  : 2 for dev at each iteration
                  : 3 for dev and lam
         facup    : for search control - increase in constraint lam,
         facdn    : for search control - decrease in constraint lam,
         laminit  : for search control - initial value of lam
         lamlow   : for search control - lower limit on lam,
```

```
lammax   : for search control - upper limit on lam,
delta    : parameter perturbation for numerical derivatives
maxinn   : maximum for inner loop of search,
eps      : singularity criterion.
```

On setting `controlpar=[]` these control parameters are given default values. It is possible to set a truncated selection as in the following example:

```
maxiter=60
convcrit=0.0000001
controlpar=[maxiter,convcrit]
```

The search constraint parameter `lam` is adjusted according to the progress of the deviance minimization. Large values lead to short steps with greater assurance of a reduction, but a smaller magnitude of reduction. Small values of `lam` lead to longer steps which can sometimes result in a greater magnitude of reduction but can also lead to an increase if the deviance function departs substantially from quadratic in the coefficients. Longer steps may also lead to inadmissible parameter values, such non-positive variance matrices. If such an increase in the deviance or an inadmissible model occurs, `lam` is increased by `facup` and a new, shorter, parameter step is trialled. If the step leads to a deviance reduction, `lam` is reduced by `facdn`so that a longer step is trialled at the next iteration.

The perturbation parameter `delta` is used to control the magnitude of parameter perturbations in construction of numerical derivatives of the residuals with respect to the parameters. The perturbation is suitably scaled for the different model parameters. A perturbation may take a parameter into an inadmissible model region, in which case the sign of the perturbation is reversed.

The parameter `maxinn` limits the number of attempts to find a lower deviance by trial steps from a particular set of model parameters. The main-step iteration count is the number of distinct steps to reduced deviance values, the number of attempts to find a lower deviance within a main iteration contributes to the total iteration count.

The parameter `eps` should be set to a small positive value. It is used to detect near singularity of the equations for the parameter step, which indicates co-linearity in the model parameters.

The default values of `controlpar` are:

```
maxiter =50;
convcrit=0.0001;
monind  = 2;
facup   = 5.0;
facdn   = 5.0;
laminit = 0.01;
lamlow  = 0.0001;
lammax  = 10000;
delta   = 1.e-7;
maxinn  = 20;
eps     = 1.e-9;
```

The sequence of estimation of the different parameter groups

```
strategy: matrix specifying an initial loop pattern of strategic search
         and a continuing loop pattern of strategic search.
         each row consists of five 0 or 1 indicators followed by a positive integer (r)
         the indicators specify which of the parameter groups (as defined for initpar above)
         are to be estimated in that iterative substep which is repeated r times
         a zero row separates the initial from the continuing pattern
         the initial pattern is used once to start the estimation process
         the continuing pattern is repeated within each subsequent main step of the iterations
         If set to {} the default used is
        [ 0, 0, 0, 1, 0, 1    % explanatory regression coefficients only
          1, 0, 0, 0, 1, 2    % autoregressive and structural coefficients
          0, 1, 1, 0, 0, 2    % disturbance and observation noise terms
          0, 0, 0, 0, 0, 0    % separation row
          1, 1, 1, 1, 1, 1    % all parameters simultaneously
          1, 0, 0, 0, 1, 1    % autoregressive and structural coefficients
          0, 1, 1, 0, 0, 1 ]; %disturbance and observation noise terms
         The matrix is internally amended to incorporate parameter constraints and options
------------------------------------------------------------------------
```

The five parameter groups defined under the setting of `initpar` differ in nature to sufficient extent that minimization of the deviance can be accelerated by initially estimating them separately in a particular sequence. For example, any regression parameters in paramter group IV can be robustly estimated with respect to any of the other groups, whereas failure to estimate any regression can badly bias initial steps in the estimation of the CZAR model parameters in group I. Also, the autoregressive (AR) parameters in group I can be initially estimated without precise estimates of the disturbance variance parameters in group II. A default strategy is to initially estimate these three groups in sequence, allowing respectively 1, 2 and 2 iteration steps. Then a sequence of 3 steps is repeated, in which firstly all groups are estimated simultaneously, then groups I and V (AR and structural) together, and thirdly the variance parameters in groups II and III together. This strategy has been found to result in good progress to minimization of the deviance, but may be modified by the user.

2.2 The output quantities

The output structure

The output is a cell structure which can be used to access the parameter estimates and related quantities as in the example:

```
VCZARcoeff = results{1}
VCZARSig = results{2}
```

We list the function help information relating to the output and then describe some of the results further:

```
OUTPUTS: results is a cell structure with the following elements

  1: VCZARcoeff   := estimated values of CZAR coefficient array
  2: VCZARSig     := estimated variance matrix of the VCZAR model
```

```
  3: NoiseSig     := estimated observation noise variance, diagonal or full matrix
  4: xregcoeff    := estimated regression coefficients as vector
  5: Strucoeff    := estimated values of structural model matrix
  6: parstderr    := standard error array of the ordered parameter estimates
  7: parcorrmx    := correlation matrix of the ordered parameter estimates
  8: dev          := deviance of the fitted model; refer differences in dev to chi-squared
  9: pdev         := psuedo-deviance analogous to residual SS (AIC is N*log(pdev)+2*m^2*p)
 10: df           := residual degrees of freedom
 11: stanperr     := vector of standardized prediction errors
 12: predsdev     := vector of standard errors of predictions
 13: yRef         := structured multivariate time series data
               first  column = series number
               second column = original index number of that series
               third column  = time value of that series
               fourth column = series value
 14: totmodpar    := total number of model parameters
 15: subindex     := an index of locations of estimated parameters
 16: xregtot      := total number of regressors
 17: FailInd      := vector of indicators of failure conditions
                Failind(1)=1: equations for new estimates singular
                Failind(2)=1; VCZARsig non-positive if struc==0; negative if struc>0
                Failind(3)=1; NoiseSig negative or non-positive resp obsnoise==1,2
                Failind(4)=1; lack of model stability
                Failind(5)=1; Deviance non-reduction
                Failind(6)=1; Failure to find Deviance reduction
--------------------------------------------------------------------------
```

The first 5 elements of `results` supply the final values of the model parameters, i.e. their estimates or initial constrained values, as follows:

The matrix element `VCZARcoeff(i,j,k)` is the final value of the model coefficient element $(\xi_k)_{ij}$ or $(\varphi_k)_{ij}$ for the predictive and natural model forms respectively.

The matrix element `VCZARSig(i,j)` is the final value of the model disturbance term variance element $(V_e)_{ij}$ or $(V_\varepsilon)_{ij}$ for the predictive and natural model forms respectively, except for both structural model forms it is the estimate of $(V_{e_S})_{ij}$ or $(V_{\varepsilon_S})_{ij}$

With the setting of the indicator `obsnoise=0`, `NoiseSig=[]`.

With the setting of the indicator `obsnoise=2`, the matrix element `NoiseSig(i,j)` is the final value of the observation error model variance element $(V_w)_{ij}$

With the setting of the indicator `obsnoise=1`, the vector element `NoiseSig(i)` is the final value of the observation error model diagonal variance matrix element $(V_w)_{ii}$.

With the parameter setting `xreg=`{}, `xregcoeff=[]`, else it is a vector with elements that are the final values of the coefficients of all the regression vectors, in sequence, in the structure `xreg`.

With the parameter settings `struc=0`, `Strucoeff=[]`.

With the parameter settings `struc=1` and `struc=2`, the matrix element `Strucoeff(i,j)` is the final value of the structural model coefficient element $(\xi_0)_{ij}$ or $(\varphi_0)_{ij}$ for the predictive and natural model forms respectively.

The vector `parstderr` contains the standard errors of the estimates of the foregoing model parameters. They are in the sequence of the parameter groups I to V with array elements in the order by which their last index (if more than 1) varies most rapidly, but including only the upper elements of symmetric variance matrices, and only the diagonal elements of a diagonal matrix. Parameters constrained to be fixed at initial values are not estimated and therefore not included.

The symmetric matrix `parcorrmx` is the correlation matrix of the estimates of the foregoing model parameters, in the same sequence as listed in the vector `parstderr`.

`dev` and `pdev` are the estimated model deviance and pseudo-deviance as defined in the technical document *GaussianEstimation.pdf* with title *Non-linear Gaussian estimation from orthogonal residuals*. The quantity `N` in the help information for `pdev` above corresponds to the number T of orthogonal residuals in that document.

`df` is the number of residual degrees of freedom of the model, which is the length N of the vector `stanperr` below, minus the length of `parstderr`.

`stanperr` is the vector of standardized prediction errors of the values of the multivariate times series as listed in the combined time series structure `yRef` below.

`predsdev` is the vector of standard errors of the predictions of the values of the multivariate times series as listed in the combined time series structure `yRef` below. These are the linear predictions based on all previous values of the series in the order listed in `yRef`.

`yRef` is an array with 4 columns of length equal to the total number of time series values in the input structure `y`. Each time series value `y{i}[t]` is listed in column 4 in the order of its time `t` which is listed in column 3 of the same row. Column 1 lists the series number `i` and column 2 lists the index (row number) of this value in its original series. The original multivariate series can therefore be recovered from `yRef`.

`totmodpar` is the total number of model parameters in those groups included in the model, whether or not constrained to initial values, but counting only the upper elements of symmetric matrices.

`subindex` is a vector of total length equal to the number of estimated parameters. The number `subindex(i)` is the position of the estimated parameter `i` in the list of all the model parameters in groups I to V (including only the upper half of symmetric variance matrices). It is used with function `parvec2true`, described below, to map the vector of estimated parameters to their positions in the parameter structures. This is useful when displaying the model parameter standard deviations and constructing error limits on various properties of the estimated model.

`xregtot` is the total number of regressorsfor all the series in the model.

`FailInd` is a vector of 6 elements. If all these are zero the iterative estimation is assumed to have concluded successfully.

Unit values of elements 2, 3 and 4 indicate various aspects of model inadmissibility. The estimation procedure attempts to avoid these at successive iterations, but the failure can arise from inadmissible initial values.

A unit value of element 1 indicates that the equations for the new step in parameter values are singular. The estimation procedure may have converged, but to a point where the model parameters are collinear.

A unit value of element 5 indicates that the procedure always steps to an inadmissible model when attempting deviance reduction, suggesting that it has reached a boundary of the admissible parameter space. It may be possible to apply constraints to initial values to avoid this failure.

A unit value of element 6 indicates that the procedure cannot find a parameter step which achieves a reduction in deviance. It is still possible that estimation has converged to the actual deviance minimum, with the deviance reduction to that point having been too great to trigger the convergence criterion.

3 The calling structure of `VCZARestimate`

There are functions that are specifically called by `VCZARestimate`. These are:

```
VCZARlikfun
scoreconstruct
search
parvec2true
subparindex
```

These functions are described in subsequent sections. Here we show the simple calling tree for these functions within `VCZARestimate`.

`VCZARestimate` *called and initial parameters mapped to parameter vector*
  `VCZARlikfun` *evaluates initial deviance and checks initial model admissible*
    `parvec2true` *maps from parameter vector to model structural parameters*
  main iterative loop starts
    `subparindex` *determines index of estimation parameters given strategy and constraints*
    `scoreconstruct` *forms linearized least squares equations for given parameters*
      `VCZARlikfun` *forms residuals for perturbed parameters for numerical derivatives*
        `parvec2true` *maps from parameter vector to model structural parameters*
    `search` *finds new parameter vector with reduced deviance*
    `VCZARlikfun` *evaluates deviance at proposed new parameters*
      `parvec2true` *maps from parameter vector to model structural parameters*
    convergence checks applied
  end of main iteration loop
  `VCZARlikfun` *for final deviance and residuals*
    `parvec2true` *maps from parameter vector to model structural parameters*
  `subparindex` *sets index of all estimated parameters subject only to constraints*
  `scoreconstruct` *to form final parameter variance matrix of estimated parameters*
    `VCZARlikfun` *residuals for perturbed parameters for numerical derivatives*
      `parvec2true` *maps from parameter vector to model structural parameters*
return from call of `VCZARestimate`

4 `VCZARlikfun`

The usage and help lines of this function are

```
[stanperr, predsdev, dev, stab, DSig, NSig] =...
    VCZARlikfun(yRef,m,p,kappa,obsnoise,xreg,xregnum,parvec,parind,struc)
  Uses State Space formulation of the Continuous Vector ZAR model to
  calculate likelihood quantities
```

All the input arguments of the function, except `parvec`, have been explained in the main section on `VCZARestimate`.

The argument `parvec` is a vector in which the values of all the model parameter, whether they are to be estimated or constrained, are listed. They are in the sequence of the parameter groups I to V with array elements in the order by which their last index (if more than 1) varies most rapidly, but including only the upper elements of symmetric variance matrices, and only the diagonal elements of a diagonal matrix. This vector form of parameter values is convenient for the matrix manipulations relating to the linearized least squares equations.

The first three output quantities are also explained in the main section on `VCZARestimate`, the standardized prediction errors, their related standard deviations and the model deviance. These are formed within `VCZARlikfun` for the specified VCZAR model and its parameters using state-space computations. The remaining three output parameters are diagnostic quantities: `stab=0` indicates that the model is stable, else `stab=1`; `DSig=0` indicates that the disturbance noise variance matrix `VCZARSig` is positive, else `DSig=1`; `NSig=0` indicates that the observation noise variance matrix `NoiseSig` is positive, else `NSig=1`. If any of these are non-zero the model is not admissible and the first three output quantities are all set to zero.

5 parvec2true

The usage and help lines of this function are

```
[VCZARcoeff,VCZARSig,NoiseSig,xregcoeff,Strucoeff]=parvec2true(parvec,m,p,obsnoise,xregtot,struc)
  maps vector parvec to the model parameter structures
```

The function copies the vector of parameter values held in `parvec` into the model structures `VCZARcoeff`, `VCZARSig`, `NoiseSig`, `xregcoeff` and `Strucoeff`. The sequence in which they are copied is described in the previous section on `VCZARlikfun`

`m` and `p` are the model dimension and order and `obsnoise`, `struc` are options described in the input arguments to `VCZARestimate.m`. `xregtot` is described in the output `results` from `VCZARestimate.m` and is the total number of regressors.

6 subparindex

The usage and help lines of this function are

```
[subindex,lensubpar]=subparindex(rstrategy,constraints)
  index for mapping from subparvec to parvec
```

The function produces a vector `subindex` which lists the indices in the vector `parvec`, of the model parameters that are to be estimated in the current search step. These are determined firstly by the parameter groups specified by the current row vector `rstrategy` of the search matrix `strategy`, and secondly by the parameter restrictions defined in the structure `constraints`. These are described under the input arguments of `VCZARestimate.m`. `lensubpar` is the length of `subindex` and therefore the number of parameters to be estimated in the current search step.

7 scoreconstruct

The usage and help lines of this function are

```
[y,e,X,Z,scFail,stab,DSig,NSig]=scoreconstruct(yRef,m,p,kappa,obsnoise,xreg,xregnum,...
    subindex,lensubpar,parvec,parind,struc,pertscale,stanperr,predsdev,delta)
  constructs numerical derivatives X, Z of residuals and their se's for active parameters
  evaluated at current parameter values
  also saves modified responses Y, e for parameter correction estimation
```

This function implements a central computation of the iterative estimation scheme described in the technical document *GaussianEstimation.pdf* with title *Non-linear Gaussian estimation from orthogonal residuals.*

Almost all the input arguments have been described above in sections on the input and output quantities of `VCZARestimate.m`, `VCZARlikfun` or other subsidiary functions. The exception is the vector `pertscale` which is of the same length as `parvec`. It is formed within `VCZARestimate.m` from the specified model and data. It provides a scaling of the parameter perturbation magnitude `delta` (an element of `controlpar`) that is appropriate for each parameter in the model.

The output quantity `y` corresponds to the vector of standardized prediction errors $f_t = e_t/\sigma_t$ which appear in equation (17) of section 3 of *GaussianEstimation.pdf.* It is used there to define part of the linearized response vector Y, which explains the variable name, when it is, in fact, identical to the input argument `stanperr`. The output quantity `e` corresponds to the vector of actual prediction errors e_t which appear in the same equation.

The output quantity `X` is a matrix equal to A_e in equation (15) of *GaussianEstimation.pdf*, with elements defined in (18). It is part of the design matrix A of the linearized least squares equation. The output quantity `Z` corresponds to another part of the design matrix A. When multiplied by $\sqrt{2}$ it is equal to A_σ in equation (15), with elements also defined in (18).

`stab`, `DSig` and `NSig` are indicators of admissible or inadmissible model properties, as described in the section above on `VCZARlikfun`, but only following perturbation of the parameters, because the unperturbed parameters are checked to be admissible on entry to the function. `scFail` takes the value 1 if any of these fail, else is zero, so indicates that the numerical derivatives cannot be calculated. This is a fatal failure condition.

8 search

The usage and help lines of this function are

```
[newsubparvec,newstanperr,newpredsdev,newdev,Iprogress,newlam]=...
```

```
search(y,e,X,Z,subindex,lensubpar,oldparvec,parind,struc,olddev,oldlam,...
yRef,m,p,kappa,obsnoise,xreg,xregnum,controlpar)
attempts to find a new set of parameter values with improved deviance
```

The input arguments of this function are all described in previous sections, with the proviso that some are prefixed with `old` to indicate that they correspond to the latest model on which an improvement is sought. The function uses the first four arguments, produced by `scoreconstruct` described in the previous section, to construct and solve the linearized least squares equations as in (19) and (20) of *GaussianEstimation.pdf*. As outlined in section 3 of that document, a constraint is placed on the step length to new parameter values and if a reduced value of the model deviance is not found, the step length is reduced with the aim of finding such a reduction. The parameter `lam` is used internally to constrain the step length.

The output quantities have similarly been previous described, with the proviso that they are prefixed with `new`, to indicate that they relate to the new model. The exception is the indicator vector `Iprogress` whose elements correspond to those of the indicator vector `FailInd` which is returned by `VCZARestimate` and described above.

9 Modifications applied to `VCZARestimate_reml.m`

This function estimates the model using the criterion of residual maximum likelihood (REML) which can lead to more accurate estimates of the model parameters when regressors are included. The criterion is described in Tunnicliffe Wilson (1989) where the term marginal likelihood is used rather than residual likelihood. The modification is implemented by including in the state space representation of the model, additional states that are the coefficients of the regressors. These are treated as random variables with initially, before the first observation, very low precision (large variances). Finally, after updating the states with the last observation, the mean and variance of these states are those of the estimated regression coefficients. The likelihood computations are otherwise the same but yielding somewhat different values for the prediction residuals, their standard errors and the deviance. The REML estimates of regression coefficients are conditional upon the remaining final model parameter estimates.

The usage of this function is

```
results=...
  VCZARestimate_reml(y,p,kappa,parind,obsnoise,xreg,struc,initpar,constraints,controlpar,strategy)
```

Compared with `VCZARestimate` the main difference in the usage of the function is that the previous parameter group IV of regression coefficients is omitted from the initial parameter values structure `initpar`. Parameter constraints are also no longer allowed for the regression coefficients. And the `strategy` setting has no column for these coefficients. The regression variables must still be provided in the structure `xreg`.

The output structure `result` is the same as for `VCZARestimate` except for the addition of the variance matrix `xregvarmx` of the vector of estimated regression coefficients `xregcoeff`. This appears as the 6th item in the cell structure. The numbering of the remaining results is therefore incremented by one, compared with those of `VCZARestimate`.

Another point to note is that the vectors `stanperr` of standardized prediction errors and `predsdev` of the standard deviations of series predictions have lengths less than the total number of series values. This is because estimation of the initial regression coefficient states, which are unconstrained due to their initial low precision, leads to near exact prediction of several initial series values, equal in number to the total number of regressors. The corresponding residuals are near zero and prediction standard deviations large. These are not included in the computation of the model deviance or psuedo-deviance. The estimated parameters from `VCZARestimate_reml.m`, including the final regression coefficients, may be used to generate the conventional residuals using these as fixed coefficients throughout the series. This is done by setting them as initial parameters for `VCZARestimate` and calling it with the control paramameter `maxiter=-1`, which supresses any parameter change.

The help information relating to the output is:

```
OUTPUTS: results is a cell structure with the following elements

   1: VCZARcoeff   := estimated values of CZAR coefficient array
   2: VCZARSig     := estimated variance matrix of the VCZAR model
   3: NoiseSig     := estimated observation noise variance, diagonal or full matrix
   4: xregcoeff    := estimated regression coefficients as vector
   5: xregvarmx    := conditional variance matrix of regression coefficients
   6: Strucoeff    := estimated values of structural model matrix
   7: parstderr    := standard error array of the ordered parameter estimates excluding regression
   8: parcorrmx    := correlation matrix of the ordered parameter estimates excluding regression
   9: dev          := deviance of the fitted model; refer differences in dev to chi-squared
  10: pdev         := psuedo-deviance analagous to residual SS (AIC is N*log(pdev)+2*m^2*p)
  11: df           := residual degrees of freedom
  12: stanperr     := vector of standardized prediction errors
  13: predsdev     := vector of standard errors of predictions
  14: yRef         := structured multivariate time series data
                first  column = series number
                second column = original index number of that series
                third column  = time value of that series
                fourth column = series value
  15: totmodpar    := total number of model parameters
  16: subindex     := an index of locations of estimated parameters excluding regression
  17: xregtot      := total number of regressors
  18: FailInd      := vector of indicators of failure conditions
                 Failind(1)=1: equations for new estimates non-singular
                 Failind(2)=1; VCZARsig non-positive if struc==0; negative if struc>0
                 Failind(3)=1; NoiseSig negative or non-positive resp obsnoise==1,2
                 Failind(4)=1; lack of model stability
                 Failind(5)=1; Deviance non-reduction
                 Failind(6)=1; Failure to find Deviance reduction
```

There are 5 subsidiary functions that are also modified for calling by `VCZARestimate_reml.m`. We list here their usage and help lines. Their functionality is essentially the same as for the corresponding functions called by `VCZARestimate`.

9.1 VCZARlikfun_reml.m

```
[stanperr, predsdev, dev, stab, DSig, NSig, xregcoeff, xregvarmx] =...
```

```
  VCZARlikfun_reml(yRef,m,p,kappa,obsnoise,xreg,xregnum,parvec,parind,struc)
 Uses State Space formulation of the Continuous Vector ZAR model to
 calculate likelihood quantities
 Regressors are included as extra state variables for reml computation
 Final time regression coefficients and conditional variances returned
```

9.2 scoreconstruct_reml.m

```
[y,e,X,Z,scFail,stab,DSig,NSig]=scoreconstruct_reml(yRef,m,p,kappa,obsnoise,xreg,xregnum,...
  subindex,lensubpar,parvec,parind,struc,pertscale,stanperr,predsdev,delta)
 constructs numerical derivatives X, Z of residuals and se's for active parameters
 evaluated at current parameter values
 also saves modified responses Y, e for parameter correction estimation
```

9.3 search_reml.m

```
[newsubparvec,newstanperr,newpredsdev,newdev,Iprogress,newlam,xregcoeff,xregvarmx]=...
  search_reml(y,e,X,Z,subindex,lensubpar,oldparvec,parind,struc,olddev,oldlam,...
  yRef,m,p,kappa,obsnoise,xreg,xregnum,controlpar)
 attempts to find a new set of parameter values with improved deviance
```

9.4 parvec2true_reml.m

```
[VCZARcoeff,VCZARSig,NoiseSig,Strucoeff]=parvec2true_reml(parvec,m,p,obsnoise,struc)
 maps vector parvec to the model parameter structures - omitting regression coefficients for reml
```

9.5 subparindex_reml.m

```
[subindex,lensubpar]=subparindex_reml(rstrategy,constraints)
 index for mapping from subparvec to parvec for parameters currently selected in strategy
```

References

G. Tunnicliffe Wilson. On the use of marginal likelihood in time series model estimation. *Journal of the Royal Statistical Society: Series B*, 51:15–27, 1989.

G. Tunnicliffe Wilson, M. Reale, and J. Haywood. *Models for dependent time series.* New York, CRC Press, 2015.