MARSS Quick Start Guide

A MARSS model is a linear multivariate autoregressive state-space model, also called dynamic linear models (DLMs) or variable autoregressive (VAR) state-space models. The default MARSS model used in the MARSS model (`form="marxss"`) is written as follows (using $m = 2$ and $n = 3$):

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_t = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}_t \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t-1} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}_t \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}_t + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_t, \quad \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_t \sim MVN\left(\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_t, \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}_t \right)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_t = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ z_{31} & z_{32} \end{bmatrix}_t \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_t + \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \\ d_{31} & d_{32} \end{bmatrix}_t \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}_t + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_t, \quad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_t \sim MVN\left(\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_t, \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}_t \right)$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 \sim MVN\left(\begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}, \begin{bmatrix} \nu_{11} & \nu_{12} \\ \nu_{21} & \nu_{22} \end{bmatrix} \right) \quad or \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_1 \sim MVN\left(\begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix}, \begin{bmatrix} \nu_{11} & \nu_{12} \\ \nu_{21} & \nu_{22} \end{bmatrix} \right)$$

c and **d** are inputs (aka, exogenous variables or covariates or indicator variables) and should have no missing values. They are not treated as 'data' in the likelihood but as inputs.

First write your model in the form above. Many of the parameter elements will be fixed or zero and others will be shared (equal) within a matrix. The following shows an example of a mean-reverting random walk model with three observation time series:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_t = \begin{bmatrix} b & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t-1} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_t, \quad \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_t \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \right), \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 \sim MVN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_t + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_t, \quad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_t \sim MVN\left(\begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r_{11} & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \right)$$

The model specification for the MARSS package has a one-to-one relationship to the model written on paper, i.e. you specify the form of the parameter matrices using matrices in R. Matrices that combine fixed and estimated values are specified using a list matrix with numerical values for fixed values and character names for the estimated values. The following shows how to specify the model above.

```
B1=matrix(list("b",0,0,"b"),2,2)
U1=matrix(0,2,1)
Q1=matrix(c("q11","q12","q12","q22"),2,2)
Z1=matrix(c(1,0,1,1,1,0),3,2)
A1=matrix(list("a1",0,0),3,1)
R1=matrix(list("r11",0,0,0,"r",0,0,0,"r"),3,3)
pi1=matrix(0,2,1); V1=diag(1,2)
model.list=list(B=B1,U=U1,Q=Q1,Z=Z1,A=A1,R=R1,x0=pi1,V0=V1,tinitx=0)
```

For `form="marxss"` (the default), matrix names in the model list must be B, U, C, c, Q, Z, A, D, d, R, x0, and V0. The tinitx element tells MARSS whether the initial state for x is at $t = 1$ (`tinitx=1`) or $t = 0$ (`tinitx=0`). The data must be entered as a $n \times T$ matrix; a dataframe is not a matrix nor is a vector. There are many text shortcuts for common parameter forms, such as “diagonal and unequal”; see the User Guide for the possible shortcuts. You can leave off matrix names and the defaults will be used. Type ?MARSS.marxss to see the defaults for `form="marxss"`.

The call to MARSS then becomes (type `?MARSS` at the command line for more info).

```
fit=MARSS(data, model=model.list)
```

The R, Q and V0 variances can be set to zero to specify partially deterministic systems. This allows you to write MAR-p models in MARSS form for example. See the User Guide for examples.

Specification of a properly constrained model with a unique solution is the responsibility of the user because MARSS has no way to tell if you have specified an insufficiently constrained model.

Time-varying parameters and inputs

The default model form (`form="marxss"`) allows you to pass in an array of T matrices for a time-varying parameter (T is the number of time-steps in your data):

$$\begin{gathered}
\boldsymbol{x}_t = \mathbf{B}_t\boldsymbol{x}_{t-1} + \mathbf{u}_t + \mathbf{C}_t\mathbf{c}_t + \mathbf{w}_t, \quad \mathbf{W}_t \sim \text{MVN}(0, \mathbf{Q}_t) \\
\boldsymbol{y}_t = \mathbf{Z}_t\boldsymbol{x}_t + \mathbf{a}_t + \mathbf{D}_t\mathbf{d}_t + \mathbf{v}_t, \quad \mathbf{V}_t \sim \text{MVN}(0, \mathbf{R}_t) \\
\boldsymbol{x}_{t_0} \sim \text{MVN}(\boldsymbol{\xi}, \boldsymbol{\Lambda})
\end{gathered} \tag{1}$$

Zeros are allowed on the diagonals of $\mathbf{Q}$, $\mathbf{R}$ and $\boldsymbol{\Lambda}$.

The length of the 3rd dimension must be the same as your data. For example, say in your mean-reverting random walk model (the example on the first page) you wanted $\mathbf{B}(2,2)$ to be one value before $t = 20$ and another value after but $\mathbf{B}(1,1)$ to be time constant. You can pass in the following:

```
TT=dim(data)[2]
B1=array(list(),dim=c(2,2,TT))
B1[,,1:20]=matrix(list("b",0,0,"b_1"),2,2)
B1[,,21:TT]=   matrix(list("b",0,0,"b_2"),2,2)
```

Notice the specification is one-to-one to how you would write your $\mathbf{B}_t$ matrices on paper.

Inputs are specified in exactly the same manner. $\mathbf{C}$ and $\mathbf{D}$ are the estimated parameters and $\mathbf{c}$ and $\mathbf{d}$ are the inputs. Let's say you have temperature data and you want to include a linear effect of temperature that is different for each $\boldsymbol{x}$ time series:

```
C1=matrix(c("temp1","temp2"),2,1)
model.list=list(B=B1,U=U1,C=C1,c=temp,Q=Q1,Z=Z1,A=A1,R=R1,x0=pi1,V0=V1,tinitx=0)
```

If you want a factor effect, then you'll need to recode your factor as a matrix with TT columns and a row for each factor. Then you have 0 or 1 if that factor applies in time period t. $\mathbf{C}$ then has a column for each estimated factor effect.

Printing output

There are print and summary functions for marssMLE and marssm objects. marssMLE objects are what is output by a call to `MARSS()`. A marssm object is the model object in MARSS. A `print` call can also print other useful output such as the parameters, the estimated states, state standard errors, etc. Type `?print.MARSS` to see the various output that can be printed from a `print` call.

The full time-varying model used in the MARSS EM algorithm

Expectation-Maximization algorithms for unconstrained MARSS models have been around for many years. What makes the EM algorithm in MARSS different is that it is a constrained algorithm. In mathematical form, the MARSS model with linear constraints that is being fit with the package is

$$\begin{gathered}\boldsymbol{x}_t = (\boldsymbol{x}_{t-1}^\top \otimes \mathbf{I}_m)\,\text{vec}(\mathbf{B}_t) + (\mathbf{u}_t^\top \otimes \mathbf{I}_m)\,\text{vec}(\mathbf{U}_t) + \mathbf{w}_t, \quad \mathbf{W}_t \sim \text{MVN}(0, \mathbf{Q}_t)\\ \boldsymbol{y}_t = (\boldsymbol{x}_t^\top \otimes \mathbf{I}_n)\,\text{vec}(\mathbf{Z}_t) + (\mathbf{a}_t^\top \otimes \mathbf{I}_n)\,\text{vec}(\mathbf{A}_t) + \mathbf{v}_t, \quad \mathbf{V}_t \sim \text{MVN}(0, \mathbf{R}_t)\\ \boldsymbol{x}_{t_0} = \boldsymbol{\xi} + \mathbf{F}\mathbf{l}, \mathbf{L} \sim \text{MVN}(0, \boldsymbol{\Lambda})\end{gathered} \tag{2}$$

Each model parameter matrix, $\mathbf{B}_t$, $\mathbf{U}_t$, $\mathbf{Q}_t$, $\mathbf{Z}_t$, $\mathbf{A}_t$, and $\mathbf{R}_t$, is written as a time-varying linear model, $\mathbf{f}_t + \mathbf{D}_t\mathbf{m}$, where $\mathbf{f}$ and $\mathbf{D}$ are fully-known (not estimated and no missing values) and $\mathbf{m}$ is a column vector of the estimates elements of the parameter matrix:

$$\begin{gathered}\text{vec}(\mathbf{B}_t) = \mathbf{f}_{t,b} + \mathbf{D}_{t,b}\boldsymbol{\beta} \quad \text{vec}(\mathbf{U}_t) = \mathbf{f}_{t,u} + \mathbf{D}_{t,u}\boldsymbol{\upsilon} \quad \text{vec}(\mathbf{Q}_t) = \mathbf{f}_{t,q} + \mathbf{D}_{t,q}\mathbf{q}\\ \text{vec}(\mathbf{Z}_t) = \mathbf{f}_{t,z} + \mathbf{D}_{t,z}\boldsymbol{\zeta} \quad \text{vec}(\mathbf{A}_t) = \mathbf{f}_{t,a} + \mathbf{D}_{t,a}\boldsymbol{\alpha} \quad \text{vec}(\mathbf{R}_t) = \mathbf{f}_{t,r} + \mathbf{D}_{t,r}\mathbf{r}\\ \text{vec}(\boldsymbol{\Lambda}) = \mathbf{f}_\lambda + \mathbf{D}_\lambda\boldsymbol{\lambda} \quad \text{vec}(\boldsymbol{\xi}) = \mathbf{f}_\xi + \mathbf{D}_\xi\mathbf{p}\end{gathered}$$

The internal MARSS model specification (marssm objects) is just a list with the $\mathbf{f}_t$ and $\mathbf{D}_t$ matrices for each parameter. The output from fitting are the vectors, $\boldsymbol{\beta}$, $\boldsymbol{\upsilon}$, etc. The trick is to rewrite the user's linear multivariate problem into the general form (equation 2). MARSS does this using functions that take more familiar arguments as input and then constructs the $\mathbf{f}_t$ and $\mathbf{D}_t$ matrices. Because the $\mathbf{f}_t$ and $\mathbf{D}_t$ can be whatever the user wants (assuming they are the right shape), this allows users to include covariates, trends (linear, sinusoidal, etc) or indicator variables in a variety of ways.