Mathematical Explanation of the Multivariate Auto-Regressive State-Space (MARSS) Model Used in the Study of Barton Springs Salamander Populations

Aaron Richter, EIT
Environmental Resources Management Division
Water Resources Evaluation Secton

SR-12-02
December, 2011

Abstract

The Barton Spring Salamander (Eurycea sosorum) is a Federally listed endangered species found only in springs in Zilker Park in central Austin, Texas. Subterranean movement of the salamander population makes estimating the true population a difficult task. In order to estimate the population and track any temporal trends, City of Austin scientists conduct monthly count surveys of each spring. Data collected from 2004 to 2011 was used in the MARSS (Multivariate Auto-Regressive State-Space) package for the R statistical computing language to obtain population estimates and determine density-dependence. MARSS iteratively solved a set of differential equations with Gaussian errors using a Kalman Filter to estimate the population of E. sosorum. The derivation and explanation of equations used in the program are briefly described within this report. Results from this analysis are documented in a subsequent report (Bendik and Turner 2011).

Introduction

Population modeling has been used in ecology to estimate trends in growth and extinction of endangered species for many years. There are many types of models that can be used in order to perform this task. Auto-regressive models tend to be used when data consists of a time series of counts. These models have been effective because the count data matches the auto-regressive process which is an attempt to model a current value of a variable as a weighted linear sum of its previous values (Kutner *et. al.* 2005, Clark 2007). Auto-regressive models can be used for a single data set or can be used on multiple time series at once. When more than one time series is analyzed it is known as a multivariate auto-regressive model. A further extension of this type of model is the multivariate auto-regressive state-space model. This allows for the effective analysis of multiple time series while accommodating for multiple forms of error. State-space models incorporate a state variable and an observation variable that are related through a set of differential equations and make use of an observation error and a process error (Clark 2007). The state variable is a non-measured variable represented by the observed variable. In ecology, the population density of a species is considered the state variable while the observed counts is considered the observation variable. The observation error is the variation in the relationship between the true population size and the observed count (Dennis *et al.* 2006). Process error is the unexplained variation in the changes in the true population size, and represents environmental variability (Dennis *et al.* 1991). By accounting for both types of error, the MARSS model would be expected to produce a better estimate for a population trend than a normal regression model.

The City of Austin has used the multivariate auto-regressive state-space model (MARSS) developed for R (a statistical computing language and environment) to estimate population trends and test for density-dependence of the Barton Springs salamander (*Eurycea sosorum*) in multiple spring locations in Austin, Texas (Bendik and Turner 2011). The program is used to estimate the parameters of linear MARSS models with Gaussian errors (Holmes & Ward 2011). A set of differential equations with Gaussian errors can be represented as:

$$\mathbf{x_t = Bx_{t-1} + u + w_t \text{ where } w_t \sim MVN(0,Q)}$$
$$\mathbf{y_t = Zx_t + a + v_t \text{ where } v_t \sim MVN(0,R)}$$
$$\mathbf{x_o \sim MVN(\xi,\Lambda)}$$

This is the basic problem that the MARSS program solves where $\mathbf{y_t}$ is the observed count and $\mathbf{x_t}$ is the actual population of the salamander at time t. In order to document the general methodology used, we explain how the MARSS package solves for $\mathbf{y_t}$, $\mathbf{x_t}$, **B, u, Q, Z, a, R,** ξ, and Λ in order to estimate population trends and density-dependence. Population trends for the Barton Springs Salamander using two different statistical methods including the MARSS methodology and Generalized Linear Regression Model (GLM) are detailed in Bendik and Turner, 2011.

Results

In order to solve the original problem, MARSS uses the following algorithm:

$$\boldsymbol{\theta_{j+1} = \text{arg max } (\theta) \int_x \int_y \text{logL}(x,y\,;\,\theta)\, f(x,y \mid Y(1) = y(1), \theta_j)\, dx\, dy}$$

In the above equation **Y(1)** indicates the set of all **Y** that has an observation while **y(1)** denotes the actual observation. The $\boldsymbol{\theta}$ is the set of parameters that MARSS is estimating in the algorithm. In most cases of the MARSS model $\boldsymbol{\theta}$ will consist of **B, u, Q, Z, a, R,** ξ, and Λ. Solving the equation above equates to finding the set of parameters that maximizes the expected joint log likelihood of **X** and **Y**, as the integral portion of the equation is the expected joint log likelihood. This is an expectation-maximization (EM) algorithm. The EM algorithm used in MARSS works in two separate steps for every iteration. The first step is to solve the integration process. This is known as the expectation step. The second step is to find a new parameter set $(\boldsymbol{\theta_{j+1}})$ that maximizes the expected log likelihood that was computed in step one. MARSS accomplishes this on a parameter-by-parameter basis, where parameters that are not being maximized are set to their iteration ***j*** values. This two step process continues until the likelihood no longer increases by a set tolerance level.

To simplify things, MARSS uses a set of update equations for each parameter in $\boldsymbol{\theta}$. These update equations successfully solve and maximize the expected log likelihood for the specified parameter, giving the iteration ***j+1*** value. Combining all ***j+1*** parameter estimates will produce $\boldsymbol{\theta_{j+1}}$. To formulate each of the update equations we first expand the expected log likelihood of **X,Y** for the original set of equations discussed. If $\boldsymbol{\Psi}$ represents E[logL(Y,X ; θ) | Y(1) = y(1),

θ_j] which is the expected log likelihood conditioned on **(1),** the random values of **Y(1)** equal the observed values **y(1)** and **(2)** the parameter set $\boldsymbol{\theta_j}$.
Then we can write:

$$\Psi = -(1/2)*\sum(E[Y_t^T R^{-1} Y_t] - E[Y_t^T R^{-1} ZX_t] - E[(ZX_t)^T R^{-1} Y_t] - E[a^T R^{-1} Y_t] -$$
$$E[Y_t^T R^{-1} a] + E[(ZX_t)^T R^{-1} ZX_t] + E[a^T R^{-1} ZX_t] + E[(ZX_t)^T R^{-1} a] + E[a^T R^{-1} a]) -$$
$$(T/2)*\log|R| - (1/2)*\sum(E[X_t^T Q^{-1} X_t] - E[X_t^T Q^{-1} BX_{t-1}] - E[(BX_{t-1})^T Q^{-1} X_t] -$$
$$E[u^T Q^{-1} X_t] - E[X_t^T Q^{-1} u] + E[(BX_{t-1})^T Q^{-1} BX_{t-1}] + E[u^T Q^{-1} BX_{t-1}] +$$
$$E[(BX_{t-1})^T Q^{-1} u] + u^T Q^{-1} u) - (T/2)*\log|Q| -$$
$$(1/2)*(E[X_o^T V_o^{-1} X_o] - E[\xi^T \Lambda^{-1} X_o] - E[X_o^T \Lambda^{-1} \xi] + \xi^T \Lambda^{-1} \xi) - (1/2)*\log|\Lambda| - (n/2)*\log \pi$$

A superscript T denotes the transpose of either a vector or a matrix while $\mathbf{R}^{-1}$ would imply the inverse of **R**. The above expansion is correct as long as $\mathbf{x_o}$ is considered a stochastic variable (known distribution). The other option is to consider $\mathbf{x_o}$ as a fixed but unknown variable, in which case the bottom line of the above equation would be dropped and $\mathbf{X_o} \equiv \boldsymbol{\xi}$ and $\boldsymbol{\Lambda}$ would be zero. The update equations are then the partial derivative of Ψ with respect to a parameter of $\boldsymbol{\theta}$. To maximize the set $\boldsymbol{\theta}$, each partial derivative is set equal to 0 and a new value for the parameter is obtained. It should be noted that there is a difference in computing the update equations for each parameter matrix based on if the elements in the parameter are constrained or unconstrained. The unconstrained scenario is less complex mathematically so these equations are described first. As an extensive example we solve for $\mathbf{u_{j+1}}$. Developing the partial derivative of $\boldsymbol{\Psi}$ with respect to **u**:

$$\partial\Psi/\partial u = - (1/2)*\sum(- \partial(E[u^T Q^{-1} X_t])/\partial u - \partial(E[X_t^T Q^{-1} u])/\partial u + \partial(E[u^T Q^{-1} BX_{t-1}])/\partial u$$
$$+ \partial(E[(BX_{t-1})^T Q^{-1} u])/\partial u + \partial(u^T Q^{-1} u)/\partial u)$$

It is important to note that the derivative of a constant is 0. Any term in $\boldsymbol{\Psi}$ that does not involve u can be considered as a constant and is not included in the above partial derivative because the terms are equal to 0. The following rules of vector derivatives will be used in the simplification of this partial derivative:

1) $\partial(a^Tc)/\partial a = c^T$
2) $\partial(a^TCa)/\partial a = 2a^TC$

Taking the partial derivative and moving parameters out of the expected value terms leaves one with the following:

$$\partial\Psi/\partial u = -(1/2)*\sum(-(Q^{-1}E[X_t])^T - E[X_t]^TQ^{-1} + (Q^{-1}BE[X_{t-1}])^T + (BE[X_{t-1}])^TQ^{-1} + 2u^TQ^{-1})$$

Using the transpose rule for matrices that states $(ABC)^T = C^TB^TA^T$ the partial derivative can be simplified to:

$$\partial\Psi/\partial u = -(1/2)*\sum(-E[X_t]^T(Q^{-1})^T - E[X_t]^TQ^{-1} + E[X_{t-1}]^TB^T(Q^{-1})^T + E[X_{t-1}]^TB^TQ^{-1} + 2u^TQ^{-1})$$

Using the fact that Q is symmetric we know that $\mathbf{Q^{-1} = (Q^{-1})^T}$ and can further simplify the partial derivative.

$$\partial\Psi/\partial\mathbf{u} = -(1/2)*\sum(-E[X_t]^T Q^{-1} - E[X_t]^T Q^{-1} + E[X_{t-1}]^T B^T Q^{-1} + E[X_{t-1}]^T B^T Q^{-1} + 2u^T Q^{-1})$$

Combine like terms and multiply through by –(1/2) to get:

$$\partial\Psi/\partial\mathbf{u} = \sum(E[X_t]^T Q^{-1} - E[X_{t-1}]^T B^T Q^{-1} - u^T Q^{-1})$$

Now we can set $\partial\Psi/\partial u$ to 0 in order to maximize the equation. In the same step right multiply both sides by **Q**. **As $Q^{-1}Q = I$** where I is the identity matrix, right multiplying by Q will negate all Q^{-1} values to the equation.

$$0 = \sum(E[X_t]^T - E[X_{t-1}]^T B^T - u^T)$$

Take the transpose of the entire equation to get:

$$0 = \sum(E[X_t] - BE[X_{t-1}] - u)$$

From here one simply removes u from the summation statement. As u is not changing from t = 1 to t = T, one is simply subtracting u a total of T times or subtracting Tu.

$$0 = \sum(E[X_t] - BE[X_{t-1}]) - Tu$$

$$Tu = \sum(E[X_t] - BE[X_{t-1}])$$

$$u_{j+1} = (1/T) * \sum(E[X_t] - BE[X_{t-1}])$$

This is the update equation for the u parameter when the vector is unconstrained (all elements are estimated and allowed to be different). The following is a list of the other parameter update equations; however, for a more rigorous derivation of each update equation see Holmes (2011).

- $B_{j+1} = (\sum(E[X_t X_{t-1}^T] - uE[X_{t-1}^T])) * (\sum E[X_t X_{t-1}^T])^{-1}$
- $Q_{j+1} = (1/T) * \sum(E[X_t X_t^T] - E[X_t X_{t-1}^T]B^T - BE[X_{t-1}X_t^T] - E[X_t]u^T - uE[X_t^T] + BE[X_{t-1}X_{t-1}^T]B^T + BE[X_{t-1}]u^T + uE[X_{t-1}^T]B^T + uu^T)$
- $a_{j+1} = (1/T) * \sum(E[Y_t] - ZE[X_t])$
- $Z_{j+1} = (\sum(E[Y_t X_t^T] - aE[X_t^T])) * (\sum E[X_t X_t^T])^{-1}$
- $R_{j+1} = (1/T) * \sum(E[Y_t Y_t^T] - E[Y_t X_t^T]Z^T - ZE[X_t Y_t^T] - E[Y_t]a^T - aE[Y_t^T] + ZE[X_t X_t^T]Z^T + ZE[X_t]a^T + aE[X_t^T]Z^T + aa^T)$
- $\xi_{j+1} = E[X_o]$ **(stochastic initial value)**
- $\xi_{j+1} = (B^T Q^{-1} B)^{-1} B^T Q^{-1}(E[X_t] - u)$ **(fixed x_o)**
- $\Lambda_{j+1} = var[X_o]$ **(stochastic initial value)**
- **Λ_{j+1} does not exist** **(fixed x_o)**

In order to solve the initial problem, all that is needed now is to solve for the expected values listed in the equations. Before reaching this point, the update equations should be explained when the parameters are constrained (elements of a matrix are fixed or linear combinations of other elements). A matrix that is composed of fixed and shared values can be broken down into two different matrices. One matrix would contain all of the fixed elements while the other would contain all free elements. As an example, consider the following break down of a 1x4 matrix.

$$\mathbf{[a\ 2\ a{+}3c{+}8\ 7] = [0\ 2\ 8\ 7] + [a\ 0\ a{+}3c\ 0]}$$

MARSS computes the ***vec*** of each parameter matrix which is simply stacking the columns of a matrix on top of each other so that there is only a single column when the process is finished. So every parameter matrix is transformed into a column vector. This column vector is then separated into the FIXED matrix, denoted as f, and the FREE matrix, denoted $\boldsymbol{vec}(\mathbf{M}_{\mathbf{free}})$. The FREE matrix is then further separated into a column matrix, denoted as **m**, consisting of the free values (**a** and **c** in the above example) and a design matrix, denoted as **D**. The matrix **D** is formed in such a way that the equation **f+Dm** will equal the ***vec*** of the original matrix. Finally the m which sets the derivative of **Ψ** with respect to **m** to zero is found through similar algebraic computations as the unconstrained solutions. Considering the computation of $\mathbf{u}_{\mathbf{j+1}}$ in an extensive example:

$$\partial\mathbf{\Psi}/\partial\mathbf{u} = \mathbf{-\ (1/2)^*\textstyle\sum(\ -\ \partial(E[u^T\ Q^{-1}\ X_t])/\partial u - \partial(E[X_t^T\ Q^{-1}\ u])/\partial u + \partial(E[u^T\ Q^{-1}\ BX_{t-1}])/\partial u}$$
$$\mathbf{+\ \partial(E[(BX_{t-1})^T\ Q^{-1}\ u])/\partial u + \partial(u^T\ Q^{-1}\ u)/\partial u)}$$

However, this time let **u = f + Dv** and take the derivative of **Ψ** with respect to **v**.

$$\partial\mathbf{\Psi}/\partial\mathbf{v} = \mathbf{-\ (1/2)^*\textstyle\sum(\ -\ \partial(E[(f + Dv)^T\ Q^{-1}\ X_t])/\partial v - \partial(E[X_t^T\ Q^{-1}\ (f + Dv])/\partial v}$$
$$\mathbf{+\ \partial(E[(f + Dv)^T\ Q^{-1}\ BX_{t-1}])/\partial v + \partial(E[(BX_{t-1})^T\ Q^{-1}\ (f + Dv)])/\partial v}$$
$$\mathbf{+\ \partial((f + Dv)^T\ Q^{-1}\ (f + Dv))/\partial v)}$$

Drop all the terms without v in them to get:

$$\partial\mathbf{\Psi}/\partial\mathbf{v} = \mathbf{-\ (1/2)^*\textstyle\sum(\ -\ \partial(E[(Dv)^T\ Q^{-1}\ X_t])/\partial v - \partial(E[X_t^T\ Q^{-1}\ Dv])/\partial v}$$
$$\mathbf{+\ \partial(E[(Dv)^T\ Q^{-1}\ BX_{t-1}])/\partial v + \partial(E[(BX_{t-1})^T\ Q^{-1}\ Dv])/\partial v}$$
$$\mathbf{+\ \partial((f + Dv)^T\ Q^{-1}\ (f + Dv))/\partial v)}$$

Expanding the last term and taking the transpose of certain terms results in the following:

$$\partial\mathbf{\Psi}/\partial\mathbf{v} = \mathbf{-\ (1/2)^*\textstyle\sum(\ -\ \partial(E[(v^T\ D^T\ Q^{-1}\ X_t])/\partial v - \partial(E[X_t^T\ Q^{-1}\ Dv])/\partial v}$$
$$\mathbf{+\ \partial(E[v^T\ D^T\ Q^{-1}\ BX_{t-1}])/\partial v + \partial(E[(BX_{t-1})^T\ Q^{-1}\ Dv])/\partial v}$$
$$\mathbf{+\ \partial(f^TQ^{-1}f + f^TQ^{-1}Dv + (Dv)^TQ^{-1}f + (Dv)^TQ^{-1}Dv)/\partial v)}$$

Compute the derivatives and set the left side to 0 to get:

$$\mathbf{0 = -\ (1/2)^*\textstyle\sum(\ -E[(D^T\ Q^{-1}\ X_t)^T] - E[X_t^T\ Q^{-1}\ D] + E[(D^T\ Q^{-1}\ BX_{t-1})^T] + E[(BX_{t-1})^T\ Q^{-1}\ D]}$$
$$\mathbf{+\ f^TQ^{-1}D + (D^TQ^{-1}f)^T + 2v^TD^TQ^{-1}D)}$$
$$\mathbf{0 = -\ (1/2)^*\textstyle\sum(\ -E[X_t^T\ Q^{-1}\ D] - E[X_t^T\ Q^{-1}\ D] + E[(BX_{t-1})^T\ Q^{-1}\ D] + E[(BX_{t-1})^T\ Q^{-1}\ D]}$$
$$\mathbf{+\ f^TQ^{-1}D + f^TQ^{-1}D + 2v^TD^TQ^{-1}D)}$$

Combining like terms and multiplying through by –(1/2) gives:

$$\mathbf{0 = \sum(E[X_t^T Q^{-1} D] - E[(BX_{t-1})^T Q^{-1} D] - f^T Q^{-1} D - v^T D^T Q^{-1} D)}$$

Next take the transpose of both sides and move terms out of the expected values.

$$\mathbf{0 = \sum(D^T Q^{-1} E[X_t] - D^T Q^{-1} B E[X_{t-1}] - D^T Q^{-1} f - D^T Q^{-1} D v)}$$

Then remove $D^T Q^{-1}$ from the summation term as well as $D^T Q^{-1} Dv$.

$$\mathbf{0 = D^T Q^{-1} \sum(E[X_t] - BE[X_{t-1}] - f) - TD^T Q^{-1} Dv}$$

$$\mathbf{TD^T Q^{-1} Dv = D^T Q^{-1} \sum(E[X_t] - BE[X_{t-1}] - f)}$$

$$\mathbf{v_{j+1} = (1/T) * (D^T Q^{-1} D)^{-1} D^T Q^{-1} \sum(E[X_t] - BE[X_{t-1}] - f)}$$

$$\mathbf{u_{j+1} = f + Dv_{j+1}}$$

This is the general update equation for u when the vector is constrained. The following is a list of the update equations for the other constrained parameters in the MARSS model. For a more rigorous derivation of the equations see Holmes (2011).

- $\alpha_{j+1} = (1/T)(D^T R^{-1} D)^{-1} D^T R^{-1} \mathbf{(}\sum(E[Y_t] - ZE[X_t] - f)\mathbf{)}$ $\qquad$ $\mathbf{a_{j+1} = f + D\alpha_{j+1}}$
- $q_{j+1} = (1/T)(D^T D)^{-1} D^T vec\mathbf{(}\sum(E[X_t X_t^T] - E[X_t X_{t-1}^T]B^T - BE[X_{t-1} X_t^T] - E[X_t]u^T$ $uE[X_t^T] + BE[X_{t-1} X_{t-1}^T]B^T + BE[X_{t-1}]u^T + uE[X_{t-1}^T]B^T + uu^T))$ $\qquad$ $\boldsymbol{vec}\mathbf{(Q_{j+1}) = f + Dq_{j+1}}$
- $\beta_{j+1} = \mathbf{(}\sum D^T(E[X_{t-1} X_{t-1}^T] \times Q^{-1})D\mathbf{)}^{-1} * D^T\mathbf{(}\sum(vec(Q^{-1}E[X_t X_{t-1}^T]) - (E[X_{t-1} X_{t-1}^T] \times Q^{-1})f -$ $vec(Q^{-1}uE[X_{t-1}^T]))\mathbf{)}$ $\qquad$ $\boldsymbol{vec}\mathbf{(B_{j+1}) = f + D}\boldsymbol{\beta}_{\mathbf{j+1}}$
- $\zeta_{j+1} = \mathbf{(}\sum(D^T(E[X_t X_t^T] \times R^{-1})D)\mathbf{)}^{-1} D^T\mathbf{(}\sum(vec(R^{-1}E[Y_t X_t^T]) - (E[X_t X_t^T] \times R^{-1}\mathbf{)}f -$ $vec(R^1 aE[X_t^T]))\mathbf{)}$ $\qquad$ $\boldsymbol{vec}\mathbf{(Z_{j+1}) = f + D}\boldsymbol{\zeta}_{\mathbf{j+1}}$
- $\rho_{j+1} = (1/T)(D^T D)^{-1} D^T vec\mathbf{(}\sum(E[Y_t Y_t^T] - E[Y_t X_t^T]Z^T - ZE[X_t Y_t^T] - E[Y_t]a^T - aE[Y_t^T] +$ $ZE[X_t X_t^T]Z^T + ZE[X_t]a^T + aE[X_t^T]Z^T + aa^T))$ $\qquad$ $\boldsymbol{vec}\mathbf{(R_{j+1}) = f + D}\boldsymbol{\rho}_{\mathbf{j+1}}$
- $p_{j+1} = (D^T \Lambda^{-1} D)^{-1} D^T \Lambda^{-1}(E[X_o] - f)$ $\qquad$ $\boldsymbol{\xi`}_{\mathbf{j+1}} \mathbf{= f + Dp_{j+1}}$ **(stochastic initial value)**
- $p_{j+1} = (D^T B^T Q^{-1} BD)^{-1} D^T B^T Q^{-1}(E[X_t] - u - Bf)$ $\qquad$ $\boldsymbol{\xi}_{\mathbf{j+1}} \mathbf{= f + Dp_{j+1}}$ **(fixed $\mathbf{x_o}$)**
- $\Lambda_{j+1} = var[X_o]$ $\qquad$ **(stochastic initial value)**
- Λ_{j+1} does not exist $\qquad$ **(fixed $\mathbf{x_o}$)**

The symbol **×** denotes the Kronecker product in the above equations. Notice that the parameters **a, u,** and **ξ** are column vectors already and do not need to be transformed using the ***vec*** function. The only step that remains in solving the initial problem is the computation of the expected values of **X** and **Y** in the MARSS model.

The expected values and variances of **X** are computed using the Kalman Filter (default) or the Kalman Smoother in the MARSS package. The Kalman filter is itself a recursion equation that uses data up to time t for each iteration. The smoother uses all available data to calculate each expected value.

$$\mathbf{E[X_t] = C_t[BC_{t-1}B^T + Q)^{-1}(BE[X_{t-1}] + u_t) + Z^T R^{-1} y_t}$$

$$\mathbf{C_t = \{(BC_{t-1}B^T + Q)^{-1} + ZR^T Z\}^{-1}}$$

From these values calculated using the Kalman filter the rest of the expected value components are calculated in MARSS using the following set of equations:

$$\mathbf{E[X_t X_t^T] = var[X_t] + E[X_t]E[X_t^T]}$$

$$\mathbf{E[X_t X_{t-1}^T] = cov[X_t, X_{t-1}] + E[X_t]E[X_{t-1}^T]}$$

$$\mathbf{E[Y_t] = y_t - \Delta_t(y_t - ZE[X_t] - a)}$$

$$\mathbf{E[Y_t Y_t^T] = I_t^{(2)}(\Delta_t R + \Delta_t Z{*}var[X_t]Z^T\Delta_t^T)I_t^{(2)} + E[Y_t]E[Y_t^T]}$$

$$\mathbf{E[Y_t X_t^T] = \Delta_t Z{*}var[X_t] + E[Y_t]E[X_t^T]}$$

$$\mathbf{E[Y_t X_{t-1}^T] = \Delta_t Z{*}cov[X_t, X_{t-1}] + E[Y_t]E[X_{t-1}^T]}$$

$$\mathbf{where\ \Delta_t = I - R(\Omega_t^{(1)})^T\,(\Omega_t^{(1)}R(\Omega_t^{(1)})^T)^{-1}\,\Omega_t^{(1)}}$$

$$\mathbf{and\ I_t^{(2)} = (\Omega_t^{(2)})^T\,\Omega_t^{(2)}}$$

In the above equations $\mathbf{\Omega_t^{(1)}}$ denotes the matrix which pulls only the actual data $\mathbf{(y_t)}$ from **Y** where there is non-missing data while $\mathbf{\Omega_t^{(2)}}$ denotes the matrix that pulls only data $\mathbf{(y_t)}$ from **Y** where there is missing data. These expected values are input to update equations and new parameters are found until the maximum likelihood no longer increases by some set value.

Conclusion

The application of current and innovative methods of analysis is expected in the field of endangered species management. The standard of acceptance for Habitat Conservation Plans and Recovery Plans submitted to USFWS for Endangered Species Act compliance is the "best available science". In order to meet that standard, City of Austin scientists have investigated a number of methods to assess the population trends of the Barton Springs Salamander. The use of surface habitat counts when subterranean refugia exists make this analysis challenging. Application of MARSS to obtain population estimates and determine density-dependence is an attempt to resolve the complications and provide supportable population trends to habitat managers and regulatory agencies.

References

Bendik, N. and M. Turner. 2011. Estimating population trends for the Barton Springs Salamander using two different statistical methods. City of Austin, Watershed Protection Department, Environmental Resource Management: SR-12-14.

Clark, J.S. 2007. Models for Ecological Data. New Jersey: Princeton University Press.

Dennis, B., P.L. Munholland, J.M. Scott. 1991. Estimation of Growth and Extinction Parameters for Endangered Species. Ecological Monographs 61:115-143.

Dennis, B., J.M. Ponciano, S.R. Lele, M.L. Taper, and D.F. Staples. 2006. Estimating density dependence, process noise, and observation error. Ecological Monographs 76:323–341.

Holmes, E.E. 2011. Derivation of the EM algorithm for constrained and unconstrained multivariate autoregressive state-space (MARSS) models. NOAA Fisheries, Northewest Fisheries Science Center, 2725 Montlake Blvd E., Seattle, WA 98112.

Holmes, E.E., E.J. Ward. 2011. Analysis of multivariate time-series using the MARSS package, version 2.3. NOAA Fisheries, Northwest Fisheries Science Center, 2725 Montlake Blvd E., Seattle, WA 98112.

Kutner, M.H., C.J. Nachtsheim, J. Neter, W. Li. 2005. Applied Linear Statistical Models. Fifth Edition. New York: McGraw-Hill Irwin.