My divergence arguments are "cone" arguments inspired by the one from Ian Mitchell's M.Sc. thesis. Section 1 shows that all trajectories starting from inside a cone whose apex is the metastable point escape the metastable region (except, of course, for the trajectory that starts at the metastable point). This isn't enough to complete the proof of correctness of the Rambus oscillator – there could be trajectories that start in an arbitrarily small neighbourhood of the metastable point but that are not in the cone. Section 2 presents the generalization to show that the oscillator starts almost surely.

1 Simple Analysis

In the remainder of this section, subsection 1.1 presents the model of the oscillator dynamics in a region that contains the metastable point; subsection 1.2 shows that the differential component of any trajectory in this cone diverges from the metastable point; subsection 1.3 shows that trajectories on the boundary of the cone are driven into the interior of the cone. This shows that all trajectories that start in the cone stay in the cone, and that they diverge from the metastable point.

1.1 Model

Though out this analysis, I'll use the quadratic model that Chao provided. I don't believe that my approach is particularly sensitive to the numerical values involved, and I'll point out how the analysis could be generalized as I go along.

The oscillator in question is a two-stage, Rambus ring oscillator. It has four state variables, $x_1, \ldots, x_4$. The chain of forward inverters goes from x_1 to x_2 to x_3 to x_4 and back to x_1. There is a pair of cross-coupled inverters coupling x_1 and x_3 and another pair that couples x_2 and x_4. Let $[w_1; w_2; w_3; w_3]$ be the metastable equilibrium of the oscillator. I'll now do a change of variables:

$$\begin{array}{rcl} d_1 & = & \frac{1}{\sqrt{2}}\left((x_1 - w_1) - (x_3 - w_3)\right) \\ d_2 & = & \frac{1}{\sqrt{2}}\left((x_2 - w_2) - (x_4 - w_4)\right) \\ s_1 & = & \frac{1}{\sqrt{2}}\left((x_1 - w_1) + (x_3 - w_3)\right) \\ s_2 & = & \frac{1}{\sqrt{2}}\left((x_2 - w_2) + (x_4 - w_4)\right) \end{array} \tag{1}$$

Note that if all of the forward inverters are the same size and all of the cross-coupled inverters are of the same size (but the forwards and cross-coupled inverters may be different sizes), then $w_1 = w_2 = w_3 = w_4$, and Equation 1 can be simplified accordingly. I'll write

$$y \quad = \quad [y_1; y_2; y_3; y_4] \quad = \quad [d_1; d_2; s_1; s_2]$$

when it's convenient to have all of these components as a single vector.

Chao wrote matlab function C = meta_matrix(symflag, s) (see Section **??**) that produces a cell-array of matrices, C{1}, ..., C{4} where if symflag is true, the computations are performed symbolically and numerically otherwise; s is the ratio of the size of the cross-coupling inverters to the forward inverters; and

$$\dot{y}_i \quad = \quad [y;1]^T\, \mathsf{C}\{\mathsf{i}\}\, [y;1], \quad \text{for } i \in \{1, \ldots, 4\} \tag{2}$$

The C{i} matrices generated by meta_matrix(false, 1) are shown in Section **??**.

In the following, it's convenient to treat the constant, linear, and quadratic terms of $\dot{y}$ separately. Accordingly, I'll define:

$$\begin{array}{rcll} A_0(i) & = & \mathsf{C}\{\mathsf{i}\}(5,5), & 1 \leq i \leq 4 \\ A_1(i,j) & = & \mathsf{C}\{\mathsf{i}\}(j,5) + \mathsf{C}\{\mathsf{i}\}(5,j), & 1 \leq i,j \leq 4 \\ A_2(i+4*(k-1),j) & = & \mathsf{C}\{\mathsf{k}\}(i,j), & 1 \leq i,j,k \leq 4 \end{array} \tag{3}$$

This yields

$$\dot{y} = A_0 + A_1 y + (I_4 \otimes y^T) A_2 y \quad (4)$$

Where I_4 is the 4×4 identity matrix, and $\otimes$ denotes the Kronecker product. See Section **??** for examples of these matrices with all the numbers plugged in.

Section **??** presents the matlab code for a function that converts a cell array (such as C above) to a struct with fields for A_0, A_1, and A_2. Section **??** presents the matlab code for qeval, a function that evaluates quadratic functions given as matrices with fields for A_0, A_1, and A_2.

Looking at the A_1 matrix from Section **??**, I was tempted to try completing the diagonalization. Looking at the 2×2 block consisting of the first two rows and first two columns, it's clear that $(6.4672 \pm 7.2526) \cdot 10^{-3}$ are the eigenvalues for diverging from the metastable point, and their eigenvectors span the space spanned by $[+1; -1; 0; 0]$ and $[0; 0; +1; -1]$. Diagonalizing the 2×2 block consisting of the last two rows and last two columns, reveals that the other two eigenvalues are $-0.7854 \cdot 10^{-3}$ and $-15.2906 \cdot 10^{-3}$. The corresponding eigenvectors are $[+1; +1; -1; -1]$ and $[1; 1; 1; 1]$. I tried converting the quadratic model to the coordinate system spanned by the eigenvectors of A_1, but I find any advantages or simplifications by making such transformations. So, I'll continue with the coordinates of Chao's C matrices and the A_i matrices described above.

I'll leave this topic with one final observation. The eigenvalue of $-15.2906 \cdot 10^{-3}$ has a magnitude more than twice that of the next largest eigenvalues, $(6.4672 \pm 7.2526) \cdot 10^{-3}$. This means that if all four nodes of the oscillator are perturbed by the same amount in the the same direction, such a perturbation will die out rather quickly. On the other hand, the eigenvalue of $0.7854 \cdot 10^{-3}$ has about one eight the magnitude of the diverging vectors. This means that if the oscillator state is perturbed by slightly increasing the voltages on both outputs of one stage, and slightly decreaseing the voltages on the outputs of the other stage, the resulting perturbation will die out, but significantly slower than the other time-constants in the system. are the eigenvalues for diverging from the metastable point, and their eigenvectors span the space spanned by $[+1; -1; 0; 0]$ and $[0; 0; +1; -1]$.

1.2 Divergence of the Differential Component

I will consider a fixed cone whose apex is at the metastable point. I'll show that all trajectories that start (or, equivalently, pass through) any point in this cone other than the apex must eventually diverge from the a small neighbourhood around the metastable point. To start, we need a cone, and a neighbourhood. A cone is define by the inequality:

$$y_1^2 + y_2^2 \geq c^2(y_3^2 + y_4^2), \quad c > 0. \quad (5)$$

The boundary of the cone is the set of points where equality of the left and right sides of Equation 5 holds. The neighbourhood of the metastable point that I'll consider is the set of all points where each component has a magnitude of at most r:

$$\|y_1; y_2\|, \|y_3, y_4\| \leq r, \quad r > 0. \quad (6)$$

For this section, I'll choose, for simplicity, $c = 1$ and determine upper bounds for r.

Let $y = [y_1; y_2; y_3; y_4] = [d_1; d_2; s_1; s_2$ be any point in the cone; let $d = [d_1; d_2]$ and $s = [s_1; s_2]$. I'll show that $\frac{\|\dot{d}\|}{\|d\|} > \alpha$ for some $\alpha > 0$. This shows that trajectories in the cone are diverging (by the d-metric) from the metastable point. To do, this, I'll treat the linear and quadratic parts of the derivative function separately, and that involves defining a few more quantities. Let

$$\begin{aligned} D &= \begin{bmatrix} A_1(1,1) & A_1(1,2) \\ A_1(2,1) & A_1(2,2) \end{bmatrix} \\ S &= \begin{bmatrix} A_1(3,3) & A_1(3,4) \\ A_1(4,3) & A_1(4,4) \end{bmatrix} \quad (7) \\ Z_2 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

Thus,

$$A_1 = \begin{bmatrix} D & Z_2 \\ Z_2 & S \end{bmatrix} \quad (8)$$

Let $\dot{d} = \delta_1 + \delta_2$, where δ_1 is the contribution from the linear (i.e. A_1) part of the derivitive function, and δ_2 is the contribution from the quadratic part. For the model under consideration (see Section **??**,

$$D \quad = \quad 10^{-3} \ * \ \begin{bmatrix} 6.4672 & 7.2526 \\ -7.2526 & 6.4672 \end{bmatrix}$$

Thus,

$$\delta_1 \quad = \quad 6.4672 \cdot 10^{-3} d \ . \tag{9}$$

I'll now bound δ_2. A simple approach is to note that

$$\begin{aligned} |\delta_2| \quad &\leq \quad c\|d\|\|s\| \sum_{i=1}^{8} \sum_{j=1}^{4} |A_2(i,j)| \\ &\approx \quad 1.8102 \cdot 10^{-3} r \|d\| \end{aligned} \tag{10}$$

where the numerical bound uses the choice of $c = 1$ and the values of the elements of the A_2 matrix from Section **??**. Accordingly, if $r < 3.5$, then $\frac{d^T}{\|d\|}\dot{d} > 10^{-4} d$ which shows divergence in the d-plane from the metastable point. For smaller values of r, faster divergence can be shown. For example, if $r < 1$, then $\frac{d^T}{\|d\|}\dot{d} > 4.6 \cdot 10^{-3} d$.

Combining the linear and quadratic terms yields

$$\dot{d} \quad > \quad 4.6 \cdot 10^{-3} d \tag{11}$$

which shows divergence of trajectories that are in the cone.

1.3 Convergence of the Common Mode Component

I'll now show that trajectories that start in the cone stay in the cone. This requires showing that trajectories on the boundary have an inward flow. Let $\overline{y} = [\overline{y}_1; \overline{y}_2; \overline{y}_3; \overline{y}_4]$ be a point on the boundary of the cone. Equation 5 yields

$$\overline{y}_1^2 + \overline{y}_2^2 - c^2\overline{y}_3^2 - c^2\overline{y}_4^2 \quad = \quad 0$$

and differentiating yields $\xi(\overline{y}) = [-\overline{y}_1; -\overline{y}_2; c\overline{y}_3; c\overline{y}_4]$ as an inward, normal vector to the cone at the point $\overline{y}$. Thus, we want to show that $\xi(\overline{y})^T\dot{\overline{y}} > 0$. Let $\sigma(\overline{y}) = \xi(\overline{y})^T\dot{\overline{y}}$.

Taking the same general approach as in Section 1.2, we first consider the linear part of the derivative function, $A_1\overline{y}$. Let σ_1 denote this linear component:

$$\begin{aligned} \sigma_1 \quad &= \quad \xi(\overline{y})^T A_1 \overline{y} \\ &= \quad [-\overline{y}_1, -\overline{y}_2] D [\overline{y}_1; \overline{y}_2] + c[\overline{y}_3, \overline{y}_4] S [\overline{y}_3; \overline{y}_4] \\ &= \quad -6.4672 \cdot 10^{-3}\|d\|^2 + c[\overline{y}_3, \overline{y}_4] S [\overline{y}_3; \overline{y}_4] \end{aligned}$$

where I'm again using the matrix shown in Section **??** for numerical values for the elements of the various matrices. With these values, S is symmetric and negative-definite; it's eigenvalues are $-0.7854 \cdot 10^{-3}$ and $-15.2906 \cdot 10^{-3}$. Thus,

$$c[\overline{y}_3, \overline{y}_4] S [\overline{y}_3; \overline{y}_4] \quad \leq \quad -7.854 \cdot 10^{-4} c\|s\|^2$$

On the boundary of the cone, $\|d\|^2 = c^2\|s\|^2$; therefore,

$$\sigma_1 \quad \leq \quad -(6.4672 \cdot 10^{-3} + 7.854 \cdot 10^{-4}/c)\|d\|^2$$

Setting $c = 1$ yields

$$\sigma_1 \quad \leq \quad -7.2526 \cdot 10^{-3}\|d\|^2$$

The next step is to show that flows remain inward on the boundary of the cone when the quadratic term of $\dot{y}$ is included. Let σ_2 denote the component of $\xi(\overline{y})^T\dot{y}$ arising from the quadratic component of $\dot{y}$: Using the trivial bounds that $|\overline{y}_1|, |\overline{y}_2| < c\|s\|$ and $|\overline{y}_3|, |\overline{y}_4| < \|s\|$ yields:

$$\sigma_2 \quad \leq \quad \|s\|^3 \left(\begin{array}{cl} & c^3 \left(\sum_{i\in\{1,2,5,6\}} \sum_{j\in\{1,2\}} |A_2(i,j)| \right) \\ + & c^2 \left(\sum_{i\in\{1,2,5,6\}} \sum_{j\in\{3,4\}} |A_2(i,j)| + \sum_{i\in\{3,4,7,8,9,10,13,14\}} \sum_{j\in\{1,2\}} |A_2(i,j)| \right) \\ + & c \left(+ \sum_{i\in\{3,4,7,8,9,10,13,14\}} \sum_{j\in\{3,4\}} |A_2(i,j)| + \sum_{i\in\{11,12,15,16\}} \sum_{j\in\{1,2\}} |A_2(i,j)| \right) \\ + & \left(+ \sum_{i\in\{11,12,15,16\}} \sum_{j\in\{3,4\}} |A_2(i,j)| \right) \end{array} \right) \tag{12}$$

For the A_2 matrix from Section **??** and $c = 1$, this yields the bound:

$$\begin{array}{rcl} \sigma_2 & \leq & 7.0711 \cdot 10^{-4} + 1.1031 \cdot 10^{-3} c^2 \|s\|^3 \\ & = & 7.0711 \cdot 10^{-4} + 1.1031 \cdot 10^{-3} \|d\|^3 / c \\ & = & 1.8102 \cdot 10^{-3} \|d\|^3 \end{array}$$

Combining with the bound for σ_1 computed above yields

$$\xi(\overline{y}) \quad \leq \quad -7.2526 \cdot 10^{-3}\|d\|^2 + 1.8102 \cdot 10^{-3}\|d\|^3$$

Thus, trajectories on the boundary of the cone flow inward for all points where $\|d\| < 4$. This means that a bound of $r < 4$ is adequate to show inward flow on the cone boundary. Recall that to show divergence of the differential components, we derived a bound of $r < 3.5$. Thus, $r < 3.5$ remains the tightest bound, but we'll tighten that some more in the next section.

2 The RBO Starts Almost Surely

Now we consider trajectories that start outside of the cone that was considered in the previous section. Clearly, we can't hope to show that all of these trajectories escape from the metastable region – some of them must be on the stable manifold of the metastable equilibrium. Instead, we show that for any point y in a small neighbourhood (see Equation 6), we can define a cone whose apex is at y and such that all trajectories that start in the interior of this cone diverger from y in the differential coordinates, and that all such trajectories stay inside a code whose apex is translated according to the trajectory that started at y. Likewise, I'll show that all trajectories on the boundary of the cone flow inward. This shows that for any plane with defined by a fixed choice of $[s_1; s_2]$, at most one trajectory starting on that plane and near the metastable point stays near the metastable point. This establishes the desired almost-surely divergence result.

To show this divergence, I'll first show how to translate a quadratic function to a new origin. Then, the arguments are similar to those of the previous section with a few new details to account for some new non-zeros in the matrices.

Let $\hat{y}$ be any point with $\|\hat{y}_1; \hat{y}_2\|, \|\hat{y}_1; \hat{y}_2\| \leq r$. Let $P(y) = Q(\hat{y} + y) - Q(\hat{y})$. P is a quadratic function and a little bit of algebra shows:

$$\begin{array}{lrcl} & P.A_0 & = & 0, \\ & P.A_1 & = & Q.A_1 + W(\hat{y}), \\ \text{and} & P.A_2 & = & Q.A_2, \\ \text{where} & W(\hat{y})(i,j) & = & \sum_{k=1}^{4}(Q.A_2(4*(i-1)+j,k) + Q.A_2(4*(i-1)+k,j))\hat{y}_k, \quad 1 \leq i,j \end{array} \tag{13}$$

Let

$$V(i,j) \quad = \quad \sum_{k=1}^{4} |Q.A_2(4*(i-1)+j,k) + Q.A_2(4*(i-1)+k,j)|, \quad 1 \leq i,j \tag{14}$$

Noting that $\forall i \in [1 \dots 4].\ \hat{y}_i \leq r$ Then, it follows that $W(\hat{y}) \leq rV$ where the comparison is element-wise. The V matrix corresponding to the quadratic model from Section **??** is shown in Section **??**.

2.1 Divergence of the Differential Component

The analysis is very similar to that from Section 1.2. There are two key differences. First, a point in the cone can have a distance from the apex of up to $2r$ in any dimension. To see this, note that the apex could be at one extreme point of the neighbourhood of the metastable point, and the point in the cone could be at a "diagonally opposite" extreme point. Given the constraints on the neighbourhood, the apex and a cone point can't differ by $2r$ in all dimensions at the same time, but I'll use that as a simple upper bound. This increase the value of σ_2. Second, the contributions of the $W(\hat{y})$ matrix to $\dot{y}$ must be taken into account. I'll write δ_3 to denote the the contribution of the $w(\hat{y})$ component to the differential part of the derivative. We now have:

$$\begin{array}{rcll}
\delta_1 & = & 6.4672 \cdot 10^{-3} d, & \text{Eq. 9} \\
|\delta_2| & \leq & 1.8102 \cdot 10^{-3}(2r)\|d\|, & \text{Eq. 10} \\
 & = & 3.6204 \cdot 10^{-3} r\|d\| & \\
|\delta_3| & \leq & rVd & \\
 & \leq & r\left(\sum_{i\in\{1,2\}}\sum_{j\in\{1\dots4\}} V(i,j)\right)\ell_\infty(d) & \\
 & \leq & 7.2408 \cdot 10^{-3} r\|d\| &
\end{array} \tag{15}$$

Noting that $\delta_2 + \delta_3 \leq 1.086 \cdot 10^{-2} r\|d\|$, divergence of the differential component is guaranteed as long as $r < \dfrac{6.4672 \cdot 10^{-3}}{1.086 \cdot 10^{-2}} \approx 0.59$. So, $r < 0.5$ is a comfortable bound to ensure divergence of the differential component.

2.2 Convergence of the Common Mode Component

The analysis is very similar to that from Section 1.3. There is only one difference. I use σ_3 to denote the the contribution of the $W(\hat{y})$ matrix to $\dot{y}$. We have

$$\sigma_3 \quad = \quad \xi(\overline{y})^T W \overline{y} \tag{16}$$

$$\begin{aligned}
 = \quad & -(W(1,1)\overline{y}_1^2 + W(2,2)\overline{y}_2^2) + c(W(3,3)\overline{y}_3^2 + W(4,4)\overline{y}_4^2) \\
 & -(W(1,2) + W(2,1))\overline{y}_1\overline{y}_2 + c(W(3,4) + W(4,3))\overline{y}_3\overline{y}_4 \\
 & -(W(1,3)\overline{y}_1\overline{y}_3 + W(1,4)\overline{y}_1\overline{y}_4 + W(2,3)\overline{y}_2\overline{y}_3 + W(2,4)\overline{y}_2\overline{y}_4) \\
 & +c(W(3,1)\overline{y}_1\overline{y}_3 + W(3,2)\overline{y}_2\overline{y}_3 + W(4,1)\overline{y}_1\overline{y}_4 + W(4,2)\overline{y}_2\overline{y}_4)
\end{aligned} \tag{17}$$

Using the bounds that $|\overline{y}_1|, |\overline{y}_2| \leq ||\overline{d}||$ and $|\overline{y}_3|, |\overline{y}_4| \leq ||\overline{s}|| = ||\overline{d}||/c$:

$$\begin{aligned}
|\sigma_3|/r \quad \leq \quad & (V(1,1)\overline{y}_1^2 + V(2,2)\overline{y}_2^2) + c(V(3,3)\overline{y}_3^2 + V(4,4)\overline{y}_4^2) \\
& +(V(1,2) + V(2,1))||\overline{d}||^2 + c(V(3,4) + V(4,3))||\overline{s}||^2 \\
& +\sum_{i=1}^{2}\sum_{j=3}^{4} V(i,j)||\overline{s}|| \cdot ||\overline{d}|| + c\sum_{i=3}^{4}\sum_{j=1}^{2} V(i,j)||\overline{s}|| \cdot ||\overline{d}||
\end{aligned} \tag{18}$$

$$\begin{aligned}
= \quad & (V(1,1)\overline{y}_1^2 + V(2,2)\overline{y}_2^2) + c(V(3,3)\overline{y}_3^2 + V(4,4)\overline{y}_4^2) \\
& + \left((V(1,2) + V(2,1) + \sum_{i=3}^{4}\sum_{j=1}^{2} V(i,j)) + \frac{V(3,4) + V(4,3) + \sum_{i=1}^{2}\sum_{j=3}^{4} V(i,j)}{c}\right)||\overline{d}||^2
\end{aligned} \tag{19}$$

Noting $V(1,1) = V(2,2)$ and $V(3,3) = V(4,4)$, we have

$$
\begin{aligned}
|\sigma_3|/r &\leq V(1,1)||\overline{d}||^2 + cV(3,3)||\overline{s}||^2 \\
&\quad + \left((V(1,2) + V(2,1) + \sum_{i=3}^{4}\sum_{j=1}^{2} V(i,j)) + \frac{V(3,4) + V(4,3) + \sum_{i=1}^{2}\sum_{j=3}^{4} V(i,j)}{c} \right) ||\overline{d}||^2 \qquad (20) \\
&= \left(V(1,1) + (V(1,2) + V(2,1) + \sum_{i=3}^{4}\sum_{j=1}^{2} V(i,j)) + \frac{V(3,3) + V(3,4) + V(4,3) + \sum_{i=1}^{2}\sum_{j=3}^{4} V(i,j)}{c} \right) ||\overline{d}||^2 \\
&= 5.5720 \cdot 10^{-3}||\overline{d}||^2 \qquad (21)
\end{aligned}
$$

Because $||\overline{d}|| \leq 2r$,

$$
\begin{aligned}
\sigma &\leq ||\overline{d}||^2(-7.2526 \cdot 10^{-3} + 5.5720 \cdot 10^{-3} \cdot r + 1.8102 \cdot 10^{-3}||\overline{d}||) \\
&\leq ||\overline{d}||^2(-7.2526 \cdot 10^{-3} + 5.5720 \cdot 10^{-3} \cdot r + 1.8102 \cdot 10^{-3} \cdot 2 \cdot r) \\
&= ||\overline{d}||^2(-7.2526 \cdot 10^{-3} + 9.1924 \cdot 10^{-3} \cdot r) \qquad (22)
\end{aligned}
$$

This requires $r \leq 0.7890$. Therefore, $r \leq 0.5$ is a valid bound.

3 Matlab code, etc.

3.1 meta_matrix

```
function [C,Af,Ac,Bf,Bc,Cf,Cc] = meta_matrix(symbolic,s)
  % This script is used to find a prove for metastability problem of a 2-stage Rambus ring osci
  % Assume the forward inverters are of unit length and the cross-couple inverters are of lengt
  % Assume the capacitance of unit width inverters is f.
  % The result ignore the /(1+s) term

  if(nargin<1||isempty(symbolic))
    symbolic = true;
  end
  if(nargin<2||isempty(s))
    if(symbolic)
      syms('s','positive');
    else
      s = 1;
    end
  end

  n = 2; N = 2*n;


  % First, get the quadratic model of the current function of inverters.
  %   ids([Vin,Vout]) = [V;1]'*Q*[V;1];
  % Around the metastable point, the ids function is using relative position (V-0.9)/0.1
  % ids(vin,vout) = [(V-0.9)/0.1;1]'*Qr*[(V-0.9)/0.1;1];
  if(~symbolic)
    Qr = [0.0015,   0.0009,  -0.3618;
       0.0009,  -0.0008,  -0.0196;
      -0.3618,  -0.0196,  -0.2644]*1e-3;
    w = 0.1; c0 = repmat(0.9,2,1);
    Qr11 = Qr(1:2,1:2); Qr10 = Qr(1:2,3); Qr01 = Qr(3,1:2); Qr00 = Qr(3,3);
    Q11 = Qr11/w^2; Q10 = Qr10/w-Qr11*c0/w^2; Q01 = Qr01/w-c0'*Qr11/w^2;
    Q00 = c0'*Qr11*c0/w^2 -c0'*Qr10/w -Qr01*c0/w +Qr00;
  else
    syms('q11','q12','q10','q21','q22','q20','q01','q02','q00','real');
    Q11 = [q11,q12;q21,q22]; Q10 = [q10;q20]; Q01 = [q01,q02]; Q00 = [q00];
  end
  Q = [Q11,Q10;Q01,Q00];
  a = sum(sum(Q11)); b = sum(Q10)+sum(Q01); c = Q00;
  if(~symbolic)
    x_eq = (-b-sqrt(b^2-4*a*c))/(2*a); % + or - ?
  else
    syms('x_eq','positive');
  end
```

```
% function meta_matrix --- continued

  % Let’s use X to denote the voltage of each nodes, we can compute the currents to each nodes.
  % The current to each node is
  % ids(xi) = ids([x_{i-1},xi])+r*ids([x_i,x_{i+n}])
  %       = [X;1]’*(A_{i-1,i}+r*A_{i-n,i})*[X;1];
  %         = [X;1]’*A_fwd*[X;1] + [X;1]’*A_cc*[X;1]
  % Where A_{i,j} is a (N+1)x(N+1) matrix whose submatrix is Q: A([i,j,end],[i,j,end]) = Q.
  % X_dot = Ids ./ ((1+s)*f). We ignore the constant term $(1+s)*f$.
  %  X_dot = Xf_dot + s * Xc_dot
  %   Xf_dot(i) = Xv’*Afi*Xv;   Xc_dot(i) = Xv’*Aci*Xv;
  %  X = [x1;x2;x3;x4];  Xv = [X;1];
  Af = cell(N,1); Ac = cell(N,1);
  for i=1:N
    Afi = zeros(N+1,N+1); Aci = zeros(N+1,N+1);
    if(symbolic)
      Afi = sym(Afi); Aci = sym(Aci);
    end
    indf = [imod(i-1,N),i,N+1]; indc = [imod(i-n,N),i,N+1];
    Afi(indf,indf) = Q; Aci(indc,indc) = Q;
    endf{i,1} = simplify(Afi); Ac{i,1} = simplify(Aci);
  end

  % Let work in U coordinate where U = M’*X
  %   U_dot = Uf_dot + s * Uc_dot
  %   Uf_dot(i) = Uv’*Bfi*Uv;   Uc_dot(i) = Xv’*Bci*Xv
  %  U = M’*X;    Uv = [U;1];
  M = trans_matrix(n); % coordinate transformation matrix
  Mv = [M,zeros(N,1); zeros(1,N),1]; % Uv = Mv’*Xv
  Bf = cell(N,1); Bc = cell(N+1);
  for i=1:N
    Bfi = zeros(N+1,N+1); Bci = zeros(N+1,N+1);
    if(symbolic)
      Bfi = sym(Bfi); Bci = sym(Bci);
    end
    for j=1:N
      bfj = Mv’*Af{j}*Mv; bcj = Mv’*Ac{j}*Mv;
      Bfi = Bfi + M(j,i)*bfj; Bci = Bci + M(j,i)*bcj;
    end
    if(symbolic)
      Bfi = simplify(Bfi); Bci = simplify(Bci);
    end
    Bf{i,1} = Bfi; Bc{i,1} = Bci;
  end
```

```
% function meta_matrix --- continued

  % Shift the equlibrium point to the origin: W = U-Ueq
  %   W_dot = U_dot = Wf_dot + Wc_dot
  %  Wf_dot(i) = Wv'*Cfi*Wv;   Wc_dot(i) = Wv'*Cfi*Wv;
  %   W = U-Ueq;   Wv = [W;1]
  Xeq = repmat(x_eq,N,1); Ueq = M'*Xeq;
  Cf = cell(N,1); Cc = cell(N+1);
  for i=1:N
    Bfi = Bf{i}; Bfi11 = Bfi(1:N,1:N); Bci = Bc{i}; Bci11 = Bci(1:N,1:N);
    Cfi = Bfi; Cci = Bci;
    Cfi(N+1,N+1) = 0; Cci(N+1,N+1) = 0;
    Cfi(1:N,N+1) = Cfi(1:N,N+1) + Bfi11*Ueq; Cci(1:N,N+1) = Cci(1:N,N+1) + Bci11*Ueq;
    Cfi(N+1,1:N) = Cfi(N+1,1:N) + Ueq'*Bfi11; Cci(N+1,1:N) = Cci(N+1,1:N) + Ueq'*Bci11;
    if(symbolic)
      Cfi = simplify(Cfi);
      Cci = simplify(Cci);
    end
    Cf{i,1} = Cfi; Cc{i,1} = Cci;
  end

  % combine W_dot(i) = Wv'*C*Wv
  C = cell(N,1);
  for i=1:N
    Ci  = Cf{i}+s*Cc{i}; % ignore (1+s)
    if(symbolic)
      Ci = simplify(Ci);
    end
    C{i,1} = Ci;
  end
end % meta_matrix
```

```
%% compute ldot
%syms('w1','w2','w3','w4','real');
%%H = w1^2+w2^2+w3^2+w4^2; L = w1^2+w2^2-(w3^2+w4^2);
%Wv = [w1;w2;w3;w4;1];
%Hdot = sym(zeros(1,1)); Ldot = sym(zeros(1,1));
%for i=1:N
%  Hdot = simplify(Hdot + (Wv'*C{i}*Wv)*Wv(i));
%  Ldot = simplify(Ldot + (-1)^(i>2)*(Wv'*C{i}*Wv)*Wv(i));
%end
%Hdot = 2*Hdot; % see H_dot in meta_result.txt
%Ldot = 2*Ldot; % see L_dot in meta_result.txt
%%R = simplify(Ldot/H);
%%R = R/(1+s); % does the (1+s) term matter?



function i = imod(i,N)
  i = mod(i,N);
  if(i==0)
    i = N;
  end
end % imod

function u = trans_matrix(n)
  u = zeros(2*n,2*n);
  for i=1:n
    u([i,n+i],i) = [1,-1]; u([i,n+i],n+i) = [1,+1];
  end;
  u = u/sqrt(2);
end % trans_matrix
```

3.2 The $\mathsf{C}\{\mathsf{i}\}$ matrices

Chao told me that there's some unspecified scaling factor in front of the C matrix. I'm using the numbers generated by meta_matrix, hence the factors of 10^{-3}. This should give us "convenient" bounds on the diameter of the region for which we can show divergence; so, I won't bother to change the scaling.

$$
\begin{aligned}
\mathsf{C1} &= 10^{-3} * \begin{bmatrix} 0 & 0 & -0.2192 & 0.0636 & 3.2336 \\ 0 & 0 & -0.0636 & -0.1061 & 3.6263 \\ -0.2192 & -0.0636 & 0 & 0 & 0 \\ 0.0636 & -0.1061 & 0 & 0 & 0 \\ 3.2336 & 3.6263 & 0 & 0 & 0 \end{bmatrix} \\
\mathsf{C2} &= 10^{-3} * \begin{bmatrix} 0 & 0 & 0.1061 & 0.0636 & -3.6263 \\ 0 & 0 & 0.0636 & -0.2192 & 3.2336 \\ 0.1061 & 0.0636 & 0 & 0 & 0 \\ 0.0636 & -0.2192 & 0 & 0 & 0 \\ -3.6263 & 3.2336 & 0 & 0 & 0 \end{bmatrix} \\
\mathsf{C3} &= 10^{-3} * \begin{bmatrix} -0.1344 & -0.0636 & 0 & 0 & 0 \\ -0.0636 & 0.1061 & 0 & 0 & 0 \\ 0 & 0 & 0.1202 & 0.0636 & -4.0190 \\ 0 & 0 & 0.0636 & 0.1061 & -3.6263 \\ 0 & 0 & -4.0190 & -3.6263 & 0 \end{bmatrix} \\
\mathsf{C4} &= 10^{-3} * \begin{bmatrix} 0.1061 & 0.0636 & 0 & 00 & \\ 0.0636 & -0.1344 & 0 & 00 & \\ 0 & 0 & 0.1061 & 0.0636-3 & .6263 \\ 0 & 0 & 0.0636 & 0.1202-4 & .0190 \\ 0 & 0 & -3.6263 & -4.01900 & \end{bmatrix}
\end{aligned}
$$

3.3 The A matrices

Chao did a change of coordinates so that the metastable point is is at the origin. Thus, the constant term is 0:

$$
A_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

A_1 is the linear term. Note how the differential (first two rows/columns) and common-mode blocks are distinct. This shows that they correspond to the eigenvalues of the Jacobian of $\dot{y}$ at the metastable point. If we modified the oscillator circuit so that it wasn't perfectly symmetrical, we'd end up with a different Jacobian. That's OK. We could replace the simple coordinate transformation described in Section 1.1 with the one that puts the Jacobian into block-Jordan form.

$$
A_1 = 10^{-3} * \begin{bmatrix} 6.4672 & 7.2526 & 0 & 0 \\ -7.2526 & 6.4672 & 0 & 0 \\ 0 & 0 & -8.0380 & -7.2526 \\ 0 & 0 & -7.2526 & -8.0380 \end{bmatrix}
$$

A_2 is the quadratic term. I added the dashed lines to make it easier to read. The first four rows are the quadratic terms for computing $\dot{y}_1$, then next four are for $\dot{y}_2$, and so on.

$$A_2 \quad = \quad 10^{-3} * \left[\begin{array}{cccc} 0 & 0 & -0.21920 & .0636 \\ 0 & 0 & -0.0636{-}0 & .1061 \\ -0.2192 & -0.0636 & 00 & \\ 0.0636 & -0.1061 & 00 & \\ \hdashline 0 & 0 & 0.10610 & .0636 \\ 0 & 0 & 0.0636{-}0 & .2192 \\ 0.1061 & 0.0636 & 00 & \\ 0.0636 & -0.2192 & 00 & \\ \hdashline -0.1344 & -0.0636 & 00 & \\ -0.0636 & 0.1061 & 00 & \\ 0 & 0 & 0.12020 & .0636 \\ 0 & 0 & 0.06360 & .1061 \\ \hdashline 0.1061 & 0.0636 & 00 & \\ 0.0636 & -0.1344 & 00 & \\ 0 & 0 & 0.10610 & .0636 \\ 0 & 0 & 0.06360 & .1202 \end{array}\right]$$

In Section 1.1, I used the Kronecker product, $\otimes$. Of course, you can read all about it at:
http://en.wikipedia.org/wiki/Kronecker_product
. Here's an example. Suppose that $y^T = [0.8, -0.3, 0.73, -0.265]$. Then

$$I_4 \otimes y^T \quad = \quad \left[\begin{array}{cccccccc} 0.8{-}0.3 & 0.73{-}0.267 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 0.8{-}0.3 & 0.73{-}0.267 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 0.8{-}0.3 & 0.73{-}0.267 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 0.8{-}0.3 & 0.73{-}0.267 \end{array}\right]$$

We also get

$$A_2 * y \quad = \quad 10^{-3} \left[\begin{array}{c} -0.1769 \\ -0.0183 \\ -0.1563 \\ 0.0827 \\ \hdashline 0.0606 \\ 0.1045 \\ 0.0658 \\ 0.1167 \\ \hdashline -0.0884 \\ -0.0827 \\ 0.0709 \\ 0.0183 \\ \hdashline 0.0658 \\ 0.0912 \\ 0.0606 \\ 0.0146 \end{array}\right]$$

Again, I added the dashed lines to show which values go to computing which elements of $\dot{y}$. Putting it all together, we get:

$$\dot{y} \quad = \quad A_0 + A_1 y + (I_4 \otimes y^T) A_2 y \quad = \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 10^{-3} \begin{bmatrix} 2.9980 \\ -7.7423 \\ -3.9458 \\ -3.1643 \end{bmatrix} + 10^{-3} \begin{bmatrix} -0.2720 \\ 0.0342 \\ 0.0010 \\ 0.0656 \end{bmatrix} \quad = \quad 10^{-3} \begin{bmatrix} 2.7260 \\ -7.7081 \\ -3.9448 \\ -3.0988 \end{bmatrix}$$

Note that although the quadratic term makes a small contribution to $\dot{y}$, the linear term is definitely dominant (for y with this magnitude). We'll exploit this more formally to prove the divergence result that we need for the oscillator.

3.4 The chaos_cells_to_marks_matrices function

```
% Q = chaos_cells_to_marks_matrices(C)
%  Convert a model for a quadratic function from a cell-array of
%  matrices (as provided by Chao Yan's meta_matrix function) to
%  a struct with separate matrices for the constant, linear, and
%  quadratic components (as used my Mark Greenstreet's qeval
%  function, etc.).
%
%  Parameter:
%    C: A cell array with n, (m+1)-by-(m+1) matrices.
%         C denotes a quadratic function from R^m to R^n.
%         Given x in R^m, the corresponding y in R^n has:
%           y(i) = [x; 1]' C{i} [x; 1]
%
%  Result:
%    Q: A struct with three fields, A0, A1, and A2.
%       A0 is the constant term.
%       A1 is the linear term.
%       A2 is the quadratic term.
%       see qeval for a more detailed description.
%
%  See also:
%    meta_matrix, qeval.

function Q = chaos_cells_to_marks_matrices(C)
  if(nargin < 1)
    X = load('C');
    C = X.C;
  end;
  n = length(C);
  m = size(C{1},1)-1;
  Q.A0 = zeros(n,1);
  Q.A1 = zeros(m,n);
  Q.A2 = zeros(m*n,m);
  for i = 1:n
    Q.A0(i) = C{i}(m+1,m+1);
    Q.A1(i,:) = C{i}(m+1,1:m) + C{i}(1:m,m+1)';
    Q.A2((1:m)+(m*(i-1)),1:m) = C{i}(1:m,1:m);
  end % for i
end % chaos_cells_to_marks_matrices
```

3.5 The qeval function

```
% y = qeval(x, Q)
%   Evaluate a quadratic function
%   Parameters:
%     x:  a m-by-1 vector.
%     Q: a struct that holds the coefficients of the quadratic.
%       Q.A0 is a n-by-1 vector, the constant term of the quadratic.
%       Q.A1 is a n-by-m matrix, the linear term of the quadratic.
%       Q.A2 is a (m*n)-by-m matrix, the quadratic term of the quadratic.
```

```
%
%   Result: y is a n-by-1 vector.
%   Let A2i be the i^th block of Q.A2:
%     A2i = Q.A2(m*(i-1)+(1:m),:)
%   Then
%     y(i) = Q.A0(i) + Q.A1(i,:)*x + ...
%              sum_{j=1}^m sum_{k=1}^m A2i(j,k) * x(j) * x(i)
%   Equivalently,
%     dy_dt = A0 + A1*y + kron(eye(m), y')*A2*y
%
%   Vectorization:
%     If size(x) == (m,k), then size(y) will be (n,k).
%     The j^th column of y is the value for evaluating Q on the j^th
%     column of x.
%
%   See also:
%     chaos_cells_to_marks_matrices

function y = qeval(x, Q)
  % Rather than trying to make the kronecker-product method work
  % when x has multiple columns, we just do the equivalent with
  % a repmat() and couple of reshape() calls and a .* operation.
  m = size(x,1); k = size(x,2); n = size(Q.A0,1);
  y0 = repmat(Q.A0, 1, k);  % constant term
  y1 = Q.A1*x;  % linear term

  % Let y2 be the quadratic term.
  % To compute y2(i,j), let A2i be the i^th block of A2:
  %   A2i = A2(m*(i-1)+(1:m),:)
  % y2(i,j) = x(:,j)' * (A2i * x(:,j)).
  %
  % Let u1i = A2i*x.
  % The j^th column of u1i is A2i * x(:,j).
  % Let u1 be the matrix whose i^th block is u1i:
  %   u1((i-1)*m + (1:m), :) = u1i
  u1 = Q.A2 * x;

  % Note that
  %   x(:,j)' * (A2i * x(:,j)) = sum(x(:,j) .* (A2i * (x(:,j))), 1)
  % Thus,
  %   y2(i,:) = sum(x .* (A2i * x), 1) = x .* u1i
  % Let u2i = x' .* (A2i * x),
  % and let u2 be the matrix whose i^th block is u2i:
  %   u2((i-1)*m + (1:m), :) = u2i
  u2 = repmat(x, m, 1) .* u1;

  % u2 has n columns of m^2 elements.  For each column, we need to compute
  % m sums of m consecutive elements.  We do this by reshaping the matrix
  % to have n*m columns of m elements, taking the sum along each column,
  % and then reshaping again to get the m-by-n matrix we need for y2.
  y2 = reshape(sum(reshape(u2, m, n*k), 1), n, k);  % the quadratic term

  % Finally, combine the constant, linear and quadratic terms.
  y = y0 + y1 + y2;
end % qeval
```

3.6 The V matrix

$$V = 10^{-4} * \begin{bmatrix} 5.6569 & 3.3941 & 5.6569 & 3.3941 \\ 3.3941 & 5.6569 & 3.3941 & 5.6569 \\ 3.9598 & 3.3941 & 3.6770 & 3.3941 \\ 3.3941 & 3.9598 & 3.3941 & 3.6770 \end{bmatrix}$$