

# Bifurcations and EXIT charts for the Binary Erasure Channel

Christopher M. Kellett  
The Hamilton Institute  
National University of Ireland, Maynooth  
Maynooth, Co. Kildare, Ireland  
e-mail: chris.kellett@nuim.ie

Steven R. Weller  
School of Electrical Engineering and Computer Science  
University of Newcastle  
Callaghan, NSW 2308, Australia  
email: steven.weller@newcastle.edu.au

**Abstract**—In this paper we present an abstraction of the extrinsic information transfer (EXIT) chart as the interconnection of two nonlinear systems in feedback with each other. We present results on the stability of fixed points for such a dynamical system and use this framework to rederive the well-known stability condition, connecting this to the one-dimensional dynamical system describing the fractions of erasure for low-density parity-check (LDPC) codes on the binary erasure channel (BEC). We observe that the error threshold corresponds to a fixed point bifurcation for this one-dimensional system, and show that this information can be visualized using a well-known tool from control theory: the root locus plot. We further show that these bifurcations can be seen by examining the EXIT chart.

## I. INTRODUCTION

The dramatic success of turbo codes [3] and the subsequent rediscovery of low-density parity-check codes [6], [10] have made iterative decoding a subject of intense research over the past decade. A unifying framework for such codes is provided by the sum-product algorithm, operating on a suitably defined sparse graph representation of the code [8]. While it is now well known that the sum-product algorithm is exact (in the sense of generating the requisite a posteriori probabilities of the transmitted symbols) when the graph is a tree, explaining the empirical success of the algorithm on graphs which contain cycles remains a problem of fundamental interest [17].

One approach to studying the complicated behaviour of iterative decoding algorithms is to consider such algorithms as nonlinear dynamical systems. Understanding the fixed points of these algorithms then opens the way to a better understanding of their global dynamics. For the turbo code formulation of iterative decoding, such approaches have been considered by Richardson [12], Agrawal and Vardy [1], and Fu [5]. A common difficulty encountered in such approaches is the necessarily high order of the dynamical systems, and a range of solutions to this problem are suggested in [12], [1], [5].

A quite different means of analysing iterative decoding is the EXtrinsic Information Transfer (EXIT) chart method of ten Brink [16], in which a single scalar quantity (the extrinsic information between constituent decoders) is effectively used as a surrogate state variable. In this fashion, iterative decoding can be visualized as a trajectory in the plane, the axes of

which correspond to two successive time-steps in the iterative process.

EXIT chart-based methods have enjoyed immense practical success, and now form part of the toolbox for engineers analyzing and designing other (iteratively implemented) elements of the receiver chain, including multiuser detectors, equalizers, and space-time receivers. For iterative decoding, it is now known that the pragmatic basis of EXIT charts in fact rests on strong theoretical underpinnings [11], [2].

In [7] we employed tools of dynamical systems theory to study EXIT charts. Our starting point was the observation that the exchange of messages between constituent decoders can be seen as the feedback interconnection of two discrete-time dynamical systems. To this end, we consider the following two-dimensional system

$$\begin{aligned}x_{k+1} &= f(y_k) \\ y_{k+1} &= g(x_k)\end{aligned}\tag{1}$$

where  $f, g : [0, 1] \rightarrow [0, 1]$ , and both  $f(\cdot)$  and  $g(\cdot)$  are monotonically increasing on  $[0, 1]$ . We assume we are given initial conditions  $x_0, y_0 \in [0, 1]$ .

An EXIT chart is then the plot of  $x = f(y)$  and  $y = g(x)$ . Assuming the existence of an (at least local) inverse for the function  $f$ , fixed points of (1) consist of those points  $(x^*, y^*)$  satisfying

$$g(x^*) = f^{-1}(x^*), \quad y^* = g(x^*).\tag{2}$$

If we plot the two functions in the plane, we see that these points are the intersections of the graphs of  $y = g(x)$  and  $y = f^{-1}(x)$ . If both  $f$  and  $g$  are continuous on the square, Brouwer's fixed point theorem guarantees the existence of at least one fixed point. Obvious questions are: What can we say about the stability of these fixed points? What can we say about domains of attraction for fixed points?

In this paper we present results connecting the existence of fixed points for the dynamical system (1) and for the one-dimensional dynamical system describing the fraction of erasures at each iteration for low-density parity-check (LDPC) codes over the binary erasure channel (BEC) using a belief propagation decoder. In Section II we review results from [7] characterizing the stability of fixed points for the general feedback connection (1). In Section III, we show how the above

formulation gives rise to the well-known stability condition [14] and how this demonstrates that the EXIT chart provides a simple factorization of the one-dimensional dynamics. Finally, in Section IV we make novel use of the well-known root locus plot from control theory. This allows us to track the location of fixed points of the iterative decoding algorithm as a function of the channel erasure probability. We observe that the decoding threshold value corresponds to a bifurcation point for the one-dimensional dynamics describing the fraction of erasures. We also note that these bifurcations can be seen by examining the EXIT chart.

## II. FIXED POINTS

We first make precise our stability notions.

*Definition 1:* A fixed point  $x^*$  is said to be *stable* for  $x_{k+1} = f(x_k)$  if for every  $\varepsilon > 0$  there exists  $\delta > 0$  such that for all initial states satisfying  $|x_0 - x^*| < \delta$ , solutions satisfy  $|x_k - x^*| < \varepsilon$  for all  $k \in \mathbb{Z}_{\geq 0}$ . A fixed point that is not stable is said to be *unstable*.

*Definition 2:* A fixed point is said to be *asymptotically stable* if, in addition to being stable,  $|x_k - x^*| \rightarrow 0$  as  $k \rightarrow \infty$ .

For nonlinear systems, such as those described by equation (1), we may clearly have more than one equilibrium point. Consequently, we can modify the above definitions to hold locally around a fixed point. In essence, this means that there exists a neighbourhood around the fixed point wherein we have asymptotic stability. We refer to this as local asymptotic stability, and say that the fixed point is *locally asymptotically stable (LAS)*.

We will make use of the following terminology:

*Definition 3:* We call  $(x^*, y^*)$  a *stable crossing* if for some  $\varepsilon > 0$ , the graphs intersect with  $g(x) > f^{-1}(x)$  for  $x \in (x^* - \varepsilon, x^*)$  (i.e., when approaching from the left) and  $g(x) < f^{-1}(x)$  for  $x \in (x^*, x^* + \varepsilon)$  (i.e., when moving away from the fixed point to the right). We call  $(x^*, y^*)$  an *unstable crossing* if the graphs intersect with  $g(x) < f^{-1}(x)$  for  $x \in (x^* - \varepsilon, x^*)$  and  $g(x) > f^{-1}(x)$  for  $x \in (x^*, x^* + \varepsilon)$ .

The following was proved in [7]:

*Theorem 1:* Suppose  $f(\cdot)$ ,  $f^{-1}(\cdot)$ , and  $g(\cdot)$  are continuously differentiable in a neighbourhood of the fixed point  $(x^*, y^*)$ . If  $(x^*, y^*)$  is a stable crossing, then the fixed point is LAS. If, on the other hand,  $(x^*, y^*)$  is an unstable crossing, then the fixed point is unstable.

*Remark 1:* Note that, if the curves intersect, but do not cross, then it is possible to show that the fixed point is, in fact, a saddle point. ■

*Remark 2:* We observe that, so long as  $f$  and  $g$  are continuous in a neighbourhood of  $(1, 1)$ , then the rightmost fixed point must be either a stable crossing or at  $(1, 1)$ . This follows from the fact that  $f$  and  $g$  must be defined over the entirety of  $[0, 1]$ . ■

The previous result was local to the fixed points. Next, we take a more global view. We make two observations. First, if there is only one fixed point, then it is globally asymptotically stable; by which we mean that all initial conditions in the

square eventually converge to the fixed point. The following was proved in [7]:

*Lemma 1:* Suppose  $f$  and  $g$  are both continuous. If  $f \circ g(\cdot)$  has a unique fixed point  $x^*$ , then  $x^*$  is globally asymptotically stable for the difference equation  $x_{k+1} = f \circ g(x_k)$ .

Second, suppose we have multiple fixed points. In particular, consider two fixed points  $(x_1^*, y_1^*)$  and  $(x_2^*, y_2^*)$  ordered such that  $x_1^* < x_2^*$  and  $y_1^* < y_2^*$  and such that no fixed point lies between them. In [7], we showed that if  $f^{-1}(x) > g(x)$  for  $x \in (x_1^*, x_2^*)$ , then iterates of  $x_{k+1} = f \circ g(x_k)$  with  $x_0 \in (x_1^*, x_2^*)$  converge to  $x_1^*$ . If, on the other hand,  $f^{-1}(x) < g(x)$  for all  $x \in (x_1^*, x_2^*)$ , then iterates of  $x_{k+1} = f \circ g(x_k)$  with  $x_0 \in (x_1^*, x_2^*)$  converge to  $x_2^*$ .

Finally, we observe that the above statements hold for the case when a fixed point is the leftmost or rightmost fixed point in the square. This allows us to define a simple partition of the square which quickly gives the regions of attractions for the fixed points. For each unstable or saddle point, draw both a vertical and a horizontal line through the point. A LAS fixed point attracts all points in its (open) partition. Note that a saddle point will be at a corner. If  $f^{-1}(x) \geq g(x)$  around a saddle point, then it will lie on the bottom left corner of the partition containing its domain of attraction. If  $f^{-1}(x) \leq g(x)$  around a saddle point, then the saddle point will be on the top right corner of the partition containing its domain of attraction.

## III. LDPC CODES

We now turn our attention to LDPC codes transmitted over the BEC with erasure probability  $\epsilon > 0$ . We use the degree distribution polynomials

$$\lambda(x) = \sum_i \lambda_i x^{i-1}, \quad \rho(x) = \sum_i \rho_i x^{i-1} \quad (3)$$

as defined in [9]. In [15], it was shown that the EXIT function for the variable nodes is  $I_{\text{out}}^{\text{va}} = 1 - \epsilon\lambda(1 - I_{\text{in}}^{\text{va}})$ , while the EXIT function for the check nodes is  $I_{\text{out}}^{\text{ch}} = \rho(I_{\text{in}}^{\text{ch}})$ . In the dynamical system formulation of (1), this yields

$$\begin{aligned} x_{k+1} &= \rho(y_k) \\ y_{k+1} &= 1 - \epsilon\lambda(1 - x_k). \end{aligned} \quad (4)$$

For successful decoding it has been observed that a ‘‘convergence tunnel’’ allowing initial conditions at the origin to converge to the point  $(1, 1)$  is necessary [2]. Therefore, at a minimum, we require local asymptotic stability of the point  $(1, 1)$ . We therefore examine the linearization of (4) about  $(1, 1)$  and obtain

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{y}_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & \rho'(1) \\ \epsilon\lambda'(0) & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix}, \quad (5)$$

which has eigenvalues at  $\pm\sqrt{\epsilon\lambda'(0)\rho'(1)}$ . For local asymptotic stability, we require that both eigenvalues lie strictly within the unit circle, leading to the condition

$$\epsilon < \frac{1}{\lambda'(0)\rho'(1)}, \quad (6)$$

which we immediately recognize as the celebrated stability condition for the BEC [14].

Looking at (4), we consider the change of variables given by  $z = 1 - y$ ,  $x = x$ . This gives

$$\begin{aligned} x_{k+1} &= \rho(1 - z_k) \\ z_{k+1} &= \epsilon\lambda(1 - x_k). \end{aligned} \quad (7)$$

Consider the two-step mapping on  $z$ ; i.e.,

$$\begin{aligned} z_{k+2} &= 1 - y_{k+2} = 1 - (1 - \epsilon\lambda(1 - x_{k+1})) \\ &= \epsilon\lambda(1 - \rho(1 - z_k)) \end{aligned} \quad (8)$$

which is precisely the fraction of errors after each iteration of belief propagation decoding [14]. This is not surprising, of course, as it is this one-dimensional dynamical system that originally gave rise to the stability condition (6). Note that, for successful decoding, one wants the fraction of erasures to go to zero as the number of iterations goes to infinity. The change of variables bears this out as, for the EXIT chart we required local asymptotic stability of the point  $(x, y) = (1, 1)$ , whereas, for system (8), we require stability of  $z = 0 = 1 - y$ .

The above observation indicates that there is no fundamental difference between the one-dimensional system and the EXIT chart. However, the EXIT chart is significantly easier to understand. Consider, for example, the one-dimensional system describing the fraction of errors for the (3, 6)-regular LDPC code (system (8) with indices relabeled):

$$\begin{aligned} z_{k+1} &= \epsilon(25z_k^2 - 100z_k^3 + 200z_k^4 - 250z_k^5 + 210z_k^6 \\ &\quad - 120z_k^7 + 45z_k^8 - 10z_k^9 + z_k^{10}) \end{aligned} \quad (9)$$

and compare this to the dynamical system EXIT chart formulation:

$$\begin{aligned} x_{k+1} &= y_k^5 \\ y_{k+1} &= 1 - \epsilon(x_k^2 - 2x_k + 1). \end{aligned} \quad (10)$$

Clearly, the latter formulation is simpler.

For the (3, 6)-regular LDPC code, we observe that the stability condition is vacuous since  $\lambda'(0) = 0$ . In other words, for (8), the origin is LAS for all finite values of  $\epsilon$ . This motivates the following question: when are there fixed points of (8), other than the origin, in the range  $[0, 1]$ ?

To this end, we write down the condition for a fixed point of (8); i.e.,  $\epsilon\lambda(1 - \rho(1 - z)) = z$ , which is equivalent to

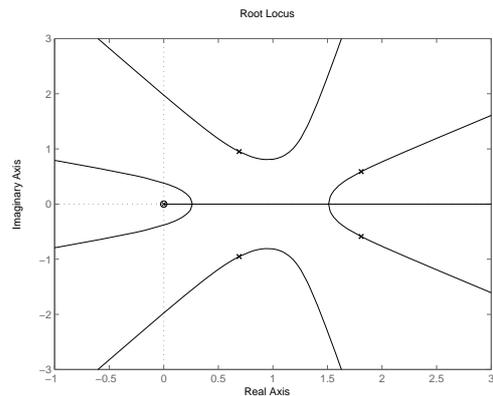
$$-\lambda(1 - \rho(1 - z)) + Kz = 0, \quad (11)$$

where  $K = \frac{1}{\epsilon}$ . Equation (11) is in so-called *root locus form*.

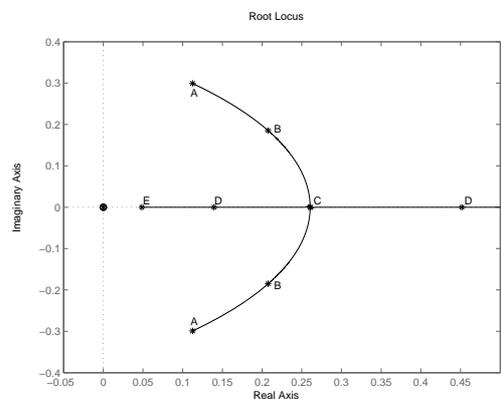
#### IV. ROOT LOCUS AND BIFURCATIONS

The root locus plot is a common graphical tool in control theory for use in designing controllers for single-input single-output linear systems (see, for example, [4, Chapter 5]). In particular, the root locus describes how the closed-loop poles of a linear feedback system vary as a function of feedback gain. In the context of the one-dimensional system described by (8), we can use the root locus plot to visualise how the fixed points vary as a function of the channel erasure probability  $\epsilon$ .

Figure 1(a) shows the root locus plot coming from (9) and (11). When  $K = 0$ , the roots are simply the roots of  $\lambda(1 - \rho(1 - z)) = 0$ , while when  $K = +\infty$  one root will be



(a)



(b)

Fig. 1. Root locus plot for (3, 6)-regular LDPC fixed points. (a) Root locus. (b) Locus for various values of  $\epsilon$ : (A)  $\epsilon = \frac{1}{5}$ ; (B)  $\epsilon = \frac{1}{3}$ ; (C)  $\epsilon = 0.4294$  (threshold value); (D)  $\epsilon = \frac{1}{2}$ ; (E)  $\epsilon = 1$ .

at the origin and the others will have an infinite modulus and be distributed symmetrically about a point on the real axis. The movement between these two extremes is continuous. Figure 1(a) shows the points at  $K = 0$  as x's. Note that, in this case, each x denotes a double root, including two at the origin.

Of course, our interest is not in  $K$ , but in  $\epsilon \in [0, 1]$ . This clearly corresponds to  $K \in [1, +\infty)$ . Furthermore, we are interested in varying  $\epsilon$  from zero to one; i.e., we wish to start with a zero probability of erasure and increase until we reach the threshold value. At  $\epsilon = 0$ , all roots, except the one at the origin, will have an infinite modulus. As  $\epsilon$  increases, the roots will move inward toward the x's. Since (8) describes the *fraction* of erasures at each iteration, our interest is in the situation when there are roots on the real axis between zero and one.

##### A. (3, 6)-regular LDPC Code

Figure 1(b) shows the relevant portion of the real axis and various values of  $\epsilon$ . As  $\epsilon$  increases from  $\frac{1}{5}$  to the threshold value of 0.4294 [13, §2.9.7], the roots move from points A

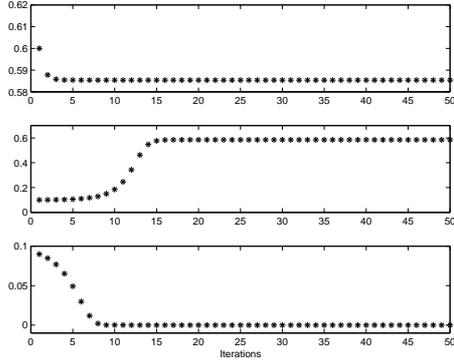


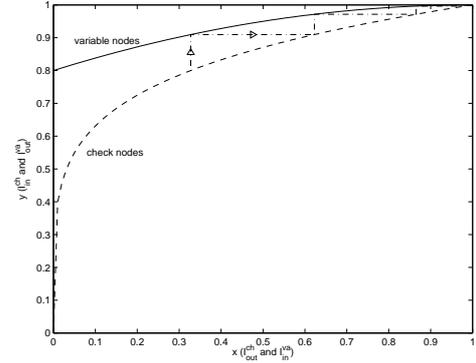
Fig. 2. Trajectories for (8) with  $\epsilon = 0.6$  from initial conditions 0.6, 0.1, and 0.09. Unstable fixed point at approximately 0.099.

to point C. At point C, a new fixed point is created. As  $\epsilon$  continues to increase, this fixed point bifurcates, with one fixed point moving in towards the origin, and another moving away from the origin. It is not difficult to show that the fixed point which moves in toward the origin is unstable and that which moves away from the origin is LAS. Figure 2 shows solutions from three different initial conditions ( $x_0 = 0.6, 0.1, 0.09$ ) when  $\epsilon = 0.6$ . Observe that there are LAS fixed points at the origin and approximately 0.585. There is also an unstable fixed point at approximately 0.099.

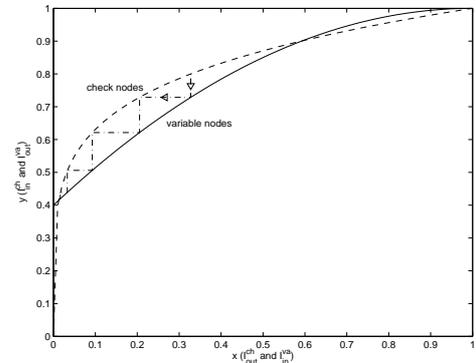
Recall that the origin is LAS for any value of  $\epsilon$ . This is consistent with the root locus plot in Figure 1(b) as, for any finite value of  $\epsilon$ , the unstable fixed point will be bounded away from the origin. Furthermore, we can observe a sizable gap between the origin and the unstable fixed point at point (E) corresponding to  $\epsilon = 1$ .

We observe that this bifurcation behavior is what we would also expect from our analysis of the EXIT chart in Section II. Specifically, Figures 3(a) and 3(b) show the EXIT chart for the (3, 6)-regular LDPC code for a value below the threshold ( $\epsilon = 0.2$  in Figure 3(a)) and a value above the threshold ( $\epsilon = 0.6$  in Figure 3(b)). We observe that, as the channel error probability increases, the variable node curve moves down until it first intersects the check node curve at approximately (0.22, 0.74), creating a fixed point. As the error probability continues to increase, the variable node curve continues to move down, causing the fixed point at (0.22, 0.74) to bifurcate into a LAS fixed point (that moves toward the origin) and an unstable fixed point (that moves toward (1, 1)). Note that (1, 1) remains a LAS fixed point. For the fixed value  $\epsilon = 0.6$ , Theorem 1 and Figure 3(b) immediately imply three fixed points; two LAS and one unstable. We also see this by examining the root locus plot in Figure 1(b), where there will be three roots on the real axis between zero and one.

Finally, we observe that the point where the root locus plot “breaks in” to the real axis between zero and one, in Figure 1(b) this is near 0.26, corresponds to the point where the convergence of the iterative decoding algorithm slows down when operating close to the threshold value. In



(a)



(b)

Fig. 3. EXIT chart for (3, 6)-regular LDPC code. (a)  $\epsilon = 0.2$  and initial condition  $(\rho(0.8), 0.8)$ . Note trajectory moves up and right. (b)  $\epsilon = 0.6$  and initial condition  $(\rho(0.8), 0.8)$ . Note trajectory moves down and left.

particular, Figure 4 show the trajectory of (8) for a channel error probability of  $\epsilon = 0.4293$ , just below the threshold value of  $\epsilon = 0.4294$ . Note that convergence requires over 250 iterations. We can see the same behavior in the EXIT chart as the “convergence tunnel” will be very nearly closed at this value of  $\epsilon$ .

### B. (2, 4)-regular LDPC Code

We now briefly turn our attention to a code that displays a qualitatively different bifurcation behavior: the (2, 4)-regular LDPC code. This qualitative difference is visible in both the root locus plot and the EXIT chart. Figure 5 shows the root locus corresponding to (11) with  $\lambda(z) = z$  and  $\rho(z) = z^3$ . Whereas for the (3, 6)-regular code we saw the locus intersect the real axis at about 0.26 when the erasure probability reached the threshold value (thereby creating a new fixed point), the locus for the (2, 4)-regular LDPC code moves from left to right on the real axis, crossing the imaginary axis at the threshold value of  $\epsilon = \frac{1}{3}$ . As  $\epsilon$  continues to increase, the original LAS fixed point at the origin bifurcates into an unstable fixed point at the origin and a LAS fixed point that moves away from the origin.

As for the (3, 6)-regular LDPC code, this behavior is

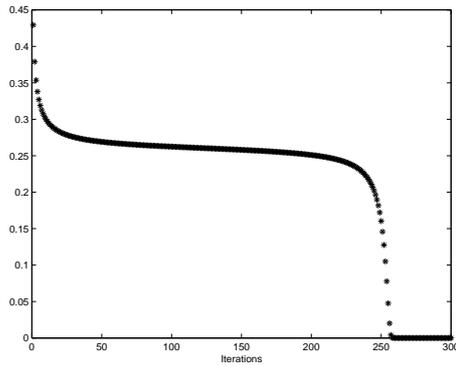


Fig. 4. Trajectory for (8) with both initial condition and  $\epsilon = 0.4293$ .

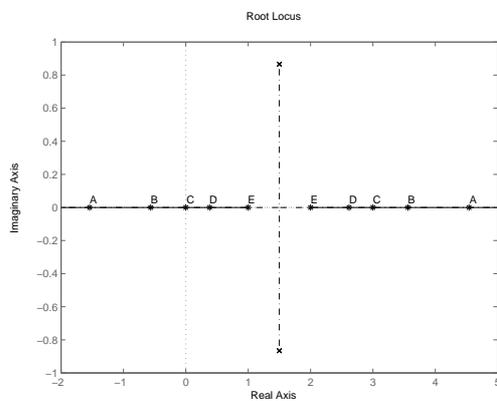
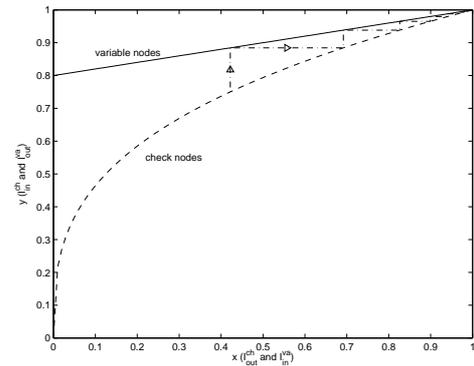


Fig. 5. Root locus for the  $(2, 4)$ -regular LDPC code. Solid line corresponds to  $\epsilon \in [0, 1]$ . (A)  $\epsilon = 0.1$ ; (B)  $\epsilon = 0.2$ ; (C)  $\epsilon = \frac{1}{3}$  (threshold value); (D)  $\epsilon = 0.5$ ; (E)  $\epsilon = 1$ .

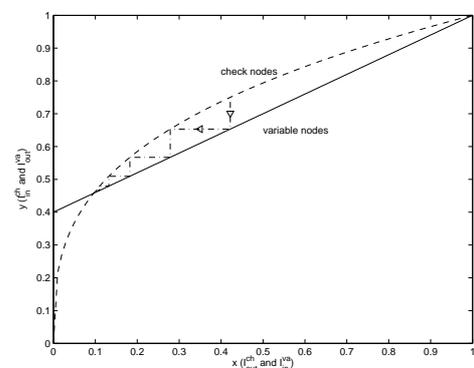
consistent with the EXIT chart, shown in Figures 6(a) and 6(b). As opposed to the “pinching off” we observed for the  $(3, 6)$ -regular EXIT chart, as the error probability increases, the variable node pivots downward with its rightmost point fixed at  $(1, 1)$ . At the threshold value of  $\epsilon = \frac{1}{3}$ , the variable node curve is tangent to the check node curve at  $(1, 1)$ . As the error probability continues to increase, the fixed point at  $(1, 1)$  bifurcates into an unstable fixed point at  $(1, 1)$  and a LAS fixed point that moves toward the origin.

## REFERENCES

- [1] D. Agrawal and A. Vardy. The turbo decoding algorithm and its phase trajectories. *IEEE Trans. Inform. Theory*, 47(2):699–722, Feb. 2001.
- [2] A. Ashikhmin, G. Kramer, and S. ten Brink. Extrinsic information transfer functions: Model and erasure channel properties. *IEEE Trans. Inform. Theory*, 50(11):2657–2673, Nov. 2004.
- [3] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. of IEEE Int'l Conference on Comm.*, volume 2, pages 1064–1070, 23–26 May 1993.
- [4] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 3rd edition, 1994.
- [5] M. Fu. Stochastic analysis of turbo decoding. *IEEE Trans. Inform. Theory*, 51(1):81–100, January 2005.
- [6] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.



(a)



(b)

Fig. 6. EXIT chart for  $(2, 4)$ -regular LDPC code. (a)  $\epsilon = 0.2$  and initial condition  $(\rho(0.75), 0.75)$ . Note trajectory moves up and right. (b)  $\epsilon = 0.6$  and initial condition  $(\rho(0.75), 0.75)$ . Note trajectory moves down and left.

- [7] C. M. Kellett and S. R. Weller. Fixed points of EXIT charts. In *Proceedings of AusCTW'06*, Perth, Australia, Feb. 1–3 2006. Available at <http://www.ee.newcastle.edu.au/staff/steve/publications.html>.
- [8] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.
- [9] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proc. of the 29th ACM Symposium on Theory of Computing*, pages 150–159, 1997.
- [10] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *IEE Elec. Lett.*, 33(6):457–458, Mar. 1997.
- [11] C. Méasson, R. L. Urbanke, A. Montanari, and T. Richardson. Life above threshold: From list decoding to area theorem and MSE. In *Proc. of IEEE Inform. Theory Workshop*, San Antonio, Texas, Oct. 24–29 2004.
- [12] T. Richardson. The geometry of turbo-decoding dynamics. *IEEE Trans. Inform. Theory*, 46(1):9–23, Jan. 2000.
- [13] T. Richardson and R. Urbanke. Modern Coding Theory. Unpublished manuscript, 30 November 2005. Available at <http://lthcwww.epfl.ch/mct>.
- [14] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, 47(2):599–618, February 2001.
- [15] I. Sutskever, S. Shamai (Shitz), and J. Ziv. Extremes of information combining. *IEEE Trans. Inform. Theory*, 51(4):1313–1325, April 2005.
- [16] S. ten Brink. Convergence of iterative decoding. *IEE Elec. Lett.*, 35(13):1117–1118, June 1999.
- [17] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Inform. Theory*, 51(7):2282–2312, July 2005.