LETTER — Communicated by Richard Golden

On the Design of BSB Neural Associative Memories Using Semidefinite Programming

Jooyoung Park
Department of Control and Instrumentation Engineering, Korea University, Chochiwon, Chungnam, 339-800, Korea

Hyuk Cho
Daihee Park
Department of Computer Science, Korea University, Chochiwon, Chungnam, 339-800, Korea

This article is concerned with the reliable search for optimally performing BSB (brain state in a box) neural associative memories given a set of prototype patterns to be stored as stable equilibrium points. By converting and/or modifying the nonlinear constraints of a known formulation for the synthesis of BSB-based associative memories into linear matrix inequalities, we recast the synthesis into semidefinite programming problems and solve them by recently developed interior point methods. The validity of this approach is illustrated by a design example.

1 Introduction

Since Hopfield (1982) showed that fully interconnected feedback neural networks trained by Hebbian learning rule can function as a new concept of associative memories, numerous neural network models have been proposed with synthesis methods for realizing associative memories (Michel & Farrell, 1990). Many studies on how well they perform as associative memories followed. In general, desirable characteristics emphasized in these performance evaluations include the following (Lillo, Miller, Hui, & Zak, 1994; Zak, Lillo, & Hui, 1996): asymptotic stability of each prototype pattern, large domain of attraction for each prototype pattern, small number of stable equilibrium points that do not correspond to prototype patterns (i.e., spurious states), global stability, incremental learning and forgetting capabilities, high storage capacity, and high retrieval efficiency.

Among the various kinds of promising neural models showing good performance are the so-called BSB (brain-state-in-a-box) neural networks. This model was first proposed by Anderson, Silverstein, Ritz, and Jones (1977) and has been noticed as particularly suitable for hardware implementation. Its theoretical aspects, especially stability issues, are now well documented. Cohen and Grossberg (1983) proved a theorem on the global stability of the continuous-time, continuous-state BSB dynamical systems with real symmetric weight matrices. Golden (1986) showed that all trajectories of the discrete-time continuous-state BSB dynamical systems with real symmetric weight matrices approach the set of equilibrium points under certain conditions, and a further extension of this was given in Golden (1993) for a generalized BSB model. Marcus and Westervelt (1989) also reported a related result for a large class of discrete-time, continuous-state BSB model type systems. Perfetti (1995), inspired by Michel, Si, and Yen (1991), analyzed qualitative properties of the BSB model and formulated the design of the BSB-based associative memories as a constrained optimization in the form of a linear programming with an additional nonlinear constraint. Also, he proposed an ad hoc iterative algorithm to solve the constrained optimization and illustrated the algorithm with some design examples.

In this article, we focus on the reliable search for optimally performing BSB neural associative memories. By converting and/or modifying the nonlinear constraints of Perfetti's formulation into linear matrix inequalities (LMIs), we transform the synthesis to semidefinite programming (SDP) problems, each comprising a linear objective and LMI constraints. Since efficient interior point algorithms are now available to solve SDP problems with guaranteed convergence (Boyd, El-Ghaoui, Feron, & Balakrishnan, 1994; Jansen, 1997), recasting the synthesis problem of the neural associative memories to an SDP problem is equivalent to finding a solution to the original problem. In this article, we use MATLAB LMI Control Toolbox (Gahinet, Nemirivskii, Laub, & Chilali, 1995) as an optimum searcher for each synthesis formulated as an SDP problem.

Throughout this article, the following definitions and notation are used: R^n denotes the normed linear space of real n vectors with the Euclidean norm $\|\cdot\|$. For a symmetric matrix $W \in R^{n\times n}$, $\lambda_{\min}(W)$ and $\|W\|$ denote the minimum eigenvalue and the induced matrix norm defined by $\max_{x\neq 0} \|Wx\|/\|x\|$, respectively. I_n denotes the $n \times n$ identity matrix, and H_n denotes the hypercube $[-1, +1]^n$. By binary vectors (or binary states), we mean the vectors whose elements are either -1 or $+1$, and B_n denotes the set of all these binary vectors in H_n. $HD(v, v^*)$ denotes the usual Hamming distance between two vectors $v \in B_n$ and $v^* \in B_n$.

The rest of this article is organized as follows: In section 2, we briefly introduce the BSB model, stability definitions, and Perfetti's formulation for the synthesis of BSB-based associative memories. In section 3, we obtain three SDP formulations (BSB I, BSB II, and BSB III) for the design of BSB neural associative memories via converting and/or modifying the original formulation. In section 4, we consider a design example to illustrate the validity of the SDP-based approach established in this article. BSBs are obtained by solving the corresponding SDP problems for the given prototype patterns of the example. Performance comparisons are made between the BSBs designed by the SDP approach and the one obtained by Perfetti's algorithm, which show the correctness and effectiveness of the proposed methods. Finally, in section 5, we give concluding remarks.

2 Background Results

The discrete-time dynamics of the BSB is described by the following state equation:

$$v(k+1) = g[v(k) + \alpha W v(k)], \tag{2.1}$$

where $v(k) \in R^n$ is the state vector at time k, $\alpha > 0$ is the step size, $W \in R^{n\times n}$ is the symmetric weight matrix, and $g : R^n \to R^n$ is a linear saturating function whose ith component is defined as follows:

$$g_i([v_1, \dots v_i, \dots, v_n]^T) = \begin{cases} 1 & \text{if} \quad v_i \geq 1, \\ v_i & \text{if} \quad -1 < v_i < 1, \\ -1 & \text{if} \quad v_i \leq 1. \end{cases} \tag{2.2}$$

Throughout this article, we assume $\alpha = 1$. In the discussion on the stability of the BSB (see equation 2.1), we use the following definitions (Lillo, Miller, Hui, & Zak, 1994; Haykin, 1994):

Definition 1. *A point $v_e \in R^n$ is an equilibrium point of the BSB if $v(0) = v_e$ implies $v(k) = v_e$, $\forall k > 0$.*

Definition 2. *An equilibrium point v_e of the BSB is stable if for any $\epsilon > 0$, there exists $\delta > 0$ such that $\|v(0) - v_e\| < \delta$ implies $\|v(k) - v_e\| < \epsilon$, $\forall k > 0$.*

Definition 3. *An equilibrium point v_e of the BSB is asymptotically stable if it is stable and there exists $\delta > 0$ such that $\|v(0) - v_e\| < \delta$ implies $v(k) \to v_e$ as $k \to \infty$.*

Definition 4. *The BSB is globally stable if every trajectory of the system converges to the set of equilibrium points.*

A well-chosen set of parameters W can make the BSB (see equation 2.1) work as an effective associative memory. A good associative memory must store each prototype pattern as an asymptotically stable equilibrium point of the network. Also, additional guidelines should be provided to address other performance indices such as the size of the domain of attraction for each prototype pattern. In Perfetti (1995), some guidelines were proposed for the BSB neural networks based on the conjecture that the absence of equilibrium points near stored patterns would increase their domains of attraction, and the experimental results showed that such strategy is very effective in reducing the number of spurious states as well as in increasing the attraction basins for prototype patterns. Perfetti (1995) formulated the synthesis of an optimally performing BSB neural associative memory as the following constrained optimization problem:

Find W, which maximizes $\delta > 0$ subject to the linear constraints

$$v_i^{(k)}\left(\sum_{j=1}^{n} w_{ij} v_j^{(k)}\right) > \delta, \forall i \in \{1, \ldots, n\}, \forall k \in \{1, \ldots, m\}, \tag{2.3}$$

$$-1 < w_{ij} < +1, \forall i, j \in \{1, \ldots, n\}, \tag{2.4}$$

$$w_{ij} = w_{ji}, \forall i, j \in \{1, \ldots, n\}, \tag{2.5}$$

$$w_{ii} = 0, \forall i \in \{1, \ldots, n\}, \tag{2.6}$$

and to the nonlinear constraint

$$\lambda_{\min}(W) > -2. \tag{2.7}$$

The role of each element of this optimization is as follows: The inequalities (see equation 2.3) are for the given prototype patterns $v^{(k)} \in B_n, k = 1, \ldots, m$ to be stored as asymptotically stable equilibrium points (theorem 2 of Perfetti, 1995). Note that the $v^{(k)}$ are hypercube vertices. Roughly speaking, with larger δ, the domain of attraction of each prototype pattern becomes wider; thus the formulation seeks the maximum δ. The bound for the elements of the weight matrix is set by equation 2.4. Both equations 2.5 and 2.7 are constraints for ensuring the global convergence of the synthesized BSBs (Golden, 1986). The condition 2.6, together with 2.3, guarantees that no binary stable equilibria exist at $HD = 1$ from each stored pattern (corollary 2 of Perfetti, 1995). The zero diagonal condition, 2.6, also ensures that only vertices can be stable equilibrium points (theorem 1 of Perfetti, 1995). Due to this property and noise, only binary steady-state solutions of equation 2.1 can be observed in practice (Perfetti, 1995).

3 Transformation into SDP Problems

In this section, we establish SDP-based synthesis methods for the BSB neural associative memories by converting and/or modifying the constraints 2.3–2.7 of Perfetti's formulation into LMIs.

An LMI is any constraint of the form

$$A(z) \stackrel{\triangle}{=} A_0 + z_1 A_1 + \cdots + z_N A_N > 0, \tag{3.1}$$

where $z \stackrel{\triangle}{=} [z_1 \cdots z_N]^T$ is the variable, and $A_0, \ldots, A_N$ are given symmetric matrices. In general, LMI constraints are given not in the canonical form, 3.1, but in a more condensed form with matrix variables. The linear constraints

2.3–2.6 of Perfetti's formulation are examples; they can be converted to the canonical form, 3.1, by defining z_i, $i = 1, \ldots, N$ as the independent scalar entries of δ and W satisfying $w_{ii} = 0$, $i = 1, \ldots, n$ and $W = W^T$. However, leaving LMIs in the condensed form not only saves notation, but also leads to more efficient computation. It is well known that optimization problems with a linear objective and LMI constraints, which are called the semidefinite programming problems, can be solved efficiently by interior point methods (Boyd et al., 1994; Jansen, 1997), and a toolbox of MATLAB that can solve convex problems involving LMIs is now available (Gahinet et al., 1995). Each of the solutions of SDP problems considered in this article was obtained by this toolbox.

3.1 First SDP Formulation (BSB I). The formulation by Perfetti has not only linear constraints but also a nonlinear constraint, which prevents us from applying the classical linear programming technique such as the simplex method. However, the nonlinear constraint can be easily converted into LMIs, which leads to our first SDP formulation. Consider the nonlinear condition, 2.7. Since W is real and symmetric, its eigenvalues are real, and corresponding eigenvectors can be chosen to be real orthonormal (Strang, 1988). Thus, its spectral decomposition can be written as

$$W = U \Lambda U^T,$$

where the real eigenvalues of W appear on the diagonal of Λ, and U, whose columns are the real orthonormal eigenvectors of W, satisfies $UU^T = U^T U = I_n$. Note that the nonlinear condition 3.7 is equivalent to

$$\Lambda > -2I_n.$$

Therefore, we have the following:

$$\Lambda > -2I_n \Leftrightarrow U \Lambda U^T > U(-2I_n)U^T \Leftrightarrow W > -2I_n \Leftrightarrow 2I_n + W > 0. \quad (3.2)$$

As a result, Perfetti's formulation can be transformed into the following SDP problem (BSB I):

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & v_i^{(k)} \left(\textstyle\sum_{j=1}^{n} w_{ij} v_j^{(k)}\right) > \delta (> 0), \forall i \in \{1, \ldots, n\}, \forall k \in \{1, \ldots, m\}, \\ & -1 < w_{ij} < +1, \forall i, j \in \{1, \ldots, n\}, \\ & w_{ij} = w_{ji}, \forall i, j \in \{1, \ldots, n\}, \\ & w_{ii} = 0, \forall i \in \{1, \ldots, n\}, \\ & 2I_n + W > 0. \end{aligned}$$

3.2 Second SDP Formulation (BSB II). Note that condition 2.4 is to limit the magnitude of weight matrix W. The same purpose can be achieved by imposing $\|W\| < s$, where s is an appropriate positive constant. Since the matrix norm constraint $\|W\|^2 < s^2$ is equivalent to

$$x^T W^T W x < x^T (s^2 I_n) x, \forall x \neq 0,$$

the norm-bound condition can be reduced to

$$sI_n - W^T (sI_n)^{-1} W > 0. \tag{3.3}$$

By the Schur complement (Boyd et al., 1995), this can be rewritten as the following LMI:

$$\begin{bmatrix} sI_n & W^T \\ W & sI_n \end{bmatrix} > 0. \tag{3.4}$$

Therefore, our second formulation utilizing $\|W\| < s$ can be reduced to the following SDP problem (BSB II):

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & v_i^{(k)} \left(\textstyle\sum_{j=1}^n w_{ij} v_j^{(k)}\right) > \delta (> 0), \forall i \in \{1, \ldots, n\}, \forall k \in \{1, \ldots, m\}, \\ & \begin{bmatrix} sI_n & W^T \\ W & sI_n \end{bmatrix} > 0, \\ & w_{ij} = w_{ji}, \forall i, j \in \{1, \ldots, n\}, \\ & w_{ii} = 0, \forall i \in \{1, \ldots, n\}, \\ & 2I_n + W > 0. \end{aligned}$$

3.3 Third SDP Formulation (BSB III). A measure of the degree to which equation 2.3 is satisfied can be constructed by defining the objective functions $Q_k(W) \triangleq v^{(k)^T} W v^{(k)}$, $k = 1, \ldots, m$. A required necessary (but not sufficient) condition for equation 2.3 is that

$$Q_k(W) > n\delta (> 0), \ \forall k \in \{1, \ldots, m\}. \tag{3.5}$$

With this observation and $\tilde{\delta} \triangleq n\delta$, we get the third SDP problem (BSB III):

$$\begin{aligned} \max \quad & \tilde{\delta} \\ \text{s.t.} \quad & v^{(k)^T} W v^{(k)} > \tilde{\delta} (> 0), \forall k \in \{1, \ldots, m\}, \\ & w_{ii} = 0, \forall i \in \{1, \ldots, n\}, \\ & W = W^T, \\ & \begin{bmatrix} sI_n & W^T \\ W & sI_n \end{bmatrix} > 0, \\ & 2I_n + W > 0. \end{aligned} \tag{3.6}$$

A remarkable feature of equation 3.6 is that it is substantially simpler than the original formulation, BSB I and BSB II.

4 Experiments and Results

In this section, a design example is presented to show the correctness of the proposed methods. Consider the BSB model, 2.1, with the dimension $n = 10$. Given are the following $m = 5$ prototype patterns that we should store as asymptotically stable equilibria of equation 2.1:

$$
\begin{aligned}
v^{(1)} &= [-1+1-1+1+1+1-1+1+1+1]^T \\
v^{(2)} &= [+1+1-1-1+1-1+1-1+1+1]^T \\
v^{(3)} &= [-1+1+1+1-1-1+1-1+1-1]^T \\
v^{(4)} &= [+1+1-1+1-1+1-1+1+1+1]^T \\
v^{(5)} &= [+1-1-1-1+1+1+1-1-1-1]^T
\end{aligned}
\tag{4.1}
$$

This is the same set of prototype patterns that was considered in Perfetti (1995).

Solving the corresponding SDPs (BSB I, BSB II with the norm-bound $s = 2.2$, and BSB III with the norm-bound $s = 2.5$), we obtained three different weight matrices. We call the BSB memories with these weight matrices BSB I, BSB II, and BSB III, respectively. For a comparison purpose, we also consider the BSB memory that was designed for equation 4.1 in Perfetti (1995). To evaluate the performance of these BSB memories, we performed simulations. For each BSB, every possible binary state was applied as an initial condition for the memory, and the memory was allowed to evolve from its initial condition to a final binary state. Our simulation results show that in all four BSB memories, the stable equilibria are the five prototype patterns (see equation 4.1) and their negatives. These negative patterns can be considered as spurious states of the memories. For an investigation of attraction domains, we collected data on the Hamming distances between the initial condition vectors and their final responses. Then, for each prototype pattern $v^{(k)}$ and each Hamming distance p, the recall probability $P(v^{(k)}, p)$ was computed as (the number of the initial condition vectors which are p-bits away from $v^{(k)}$ and converged to $v^{(k)}$)/ (the total number of the initial condition vectors which are p-bits away from $v^{(k)}$). Shown in Table 1 are their averages over the five prototype patterns $\{\sum_{k=1}^{5} P(v^{(k)}, p)\}/5$. The interpretation of the data in each entry of the table should be clear (e.g., in the first row of Table 1, which is for BSB I, the data in the entry corresponding to HD = 2 are 31.6/45. This indicates that there are 45 possible initial condition vectors at Hamming distance 2 away from a prototype pattern, and the simulation results show that, on average, 31.6 of them successfully converge to the prototype pattern). From the table, we can see that the four BSB memories have similar recall probabilities. For readers' convenience, we plotted the contents of Table 1 in Figure 1. Also, we investigated how many initial condition vectors converged to the nearest prototype pattern for a comparison of the recall quality. The initial condition vectors can be divided into three classes based on which is their final response among the

Table 1: Average Recall Probabilities of the BSB Neural Associative Memories.

	HD = 0	HD = 1	HD = 2	HD = 3	HD = 4	HD = 5
BSB I	1/1	9.6/10	31.6/45	38.0/120	19.4/210	2.8/252
BSB II	1/1	9.6/10	33.2/45	41.2/120	16.6/210	0.8/252
BSB III	1/1	9.6/10	30.8/45	40.8/120	15.6/210	4.2/252
Perfetti (1995)	1/1	9.6/10	28.8/45	42.4/120	19.2/210	1.4/252

Notes: For each prototype pattern $v^{(k)}$ and each Hamming distance p, the recall probability $P(v^{(k)}, p)$ was computed as (the number of the initial condition vectors which are p-bits away from $v^{(k)}$ and converged to $v^{(k)}$)/ (the total number of the initial condition vectors which are p-bits away from $v^{(k)}$). Shown in this table are their averages over the five prototype patterns $\{\sum_{k=1}^{5} P(v^{(k)}, p)\}/5$.

nearest prototype vector, a prototype pattern that is not the nearest one, and the negatives of prototype patterns. We call these classes best, good, and negative, respectively. Table 2 shows how many binary states belong to each class. The contents of the table indicate that BSB II is the most efficient in recalling the nearest prototype pattern. Finally, it should be noted

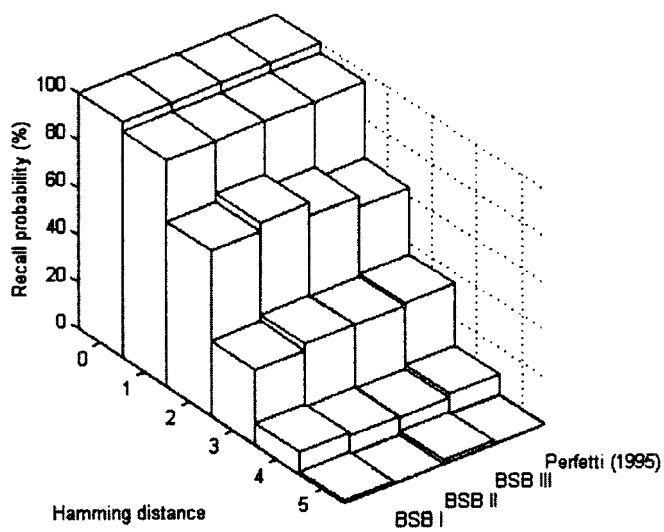

Figure 1: Comparison of average recall probabilities: For each prototype pattern $v^{(k)}$ and each Hamming distance p, the recall probability $P(v^{(k)}, p)$ was computed as the percentage of the initial condition vectors, which are p-bits away from $v^{(k)}$ and converged to $v^{(k)}$. Shown in this figure are $\{\sum_{k=1}^{5} P(v^{(k)}, p)\}/5$.

Table 2: Summary of the Final Responses of the Initial Condition Vectors.

	Class		
	Best	Good	Negative
BSB I	475	37	512
BSB II	483	29	512
BSB III	457	55	512
Perfetti (1995)	478	34	512

Notes: The initial condition vectors were divided into three classes based on their final response among the nearest prototype vector (best), a prototype pattern that is not the nearest one (good), and the negatives of prototype patterns (negative). Shown in this table are the numbers of binary vectors in these classes.

that the key contribution of this article is not that BSB I, BSB II, and BSB III have better performance than the BSB memory of Perfetti (1995) but that BSB I, BSB II, and BSB III solutions are obtained by solving a system of linear matrix inequalities, while Perfetti (1995) had to solve a system of nonlinear matrix inequalities.

5 Conclusion

In this article, we addressed the synthesis of optimally performing BSB neural associative memories by recasting a known formulation into SDP problems. This recast is particularly useful in practice, because the interior point methods that can solve SDP problems (i.e., can find the global optimum efficiently within a given tolerance or find a certificate of infeasibility) are readily available. A design example was presented to illustrate the proposed methods, and the resulting BSBs demonstrated their correctness and effectiveness, and verified positively the applicability of LMIs to the synthesis of associative memories. The main results of this article concern the problem of forcing some vertices to be attractors. To prevent other vertices from becoming attractors is often of interest. In this connection, we can easily verify with the help of corollary 4 of Perfetti (1995) that the strategy of BSB I and II, which maximizes a lower bound δ for $\sum_{j=1}^{n} w_{ij} v_i^{(k)} v_j^{(k)},\ \forall i, \forall k$, has an effect that nearby vertices of the stored prototype patterns are prevented from becoming attractors.

An important problem we have not addressed here is that of comparing the computational complexity of our LMI-based methods with other numerical methods for designing BSB model memories that exist in the literature.

References

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review, 84*, 413–451.

Boyd, S., El-Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in systems and control theory*. Philadelphia: SIAM.

Cohen, M. A., & Grossberg, S. (1983). Absolute stability of global pattern formulation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man and Cybernetics, 13*, 815–826.

Gahinet, P., Nemirivskii, A., Laub, A. J., & Chilali, M. (1995). *LMI control toolbox*. Natick, MA: Mathworks.

Golden, R. M. (1986). The brain-state-in-a-box neural model is a gradient descent algorithm. *Journal of Mathematical Psychology, 30*, 73–80.

Golden, R. M. (1993). Stability and optimization analyses of the generalized brain-state-in-a-box neural network model. *Journal of Mathematical Psychology, 37*, 282–298.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York: Macmillan.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Sciences, 79*, 2554–2558.

Jansen, B. (1997). *Interior point techniques in optimization: Complementarity, sensitivity and algorithms*. Dordrecht: Kluwer.

Lillo, W. E., Miller, D. C., Hui, S., & Zak, S. H. (1994). Synthesis of brain-state-in-a-box (BSB) based associative memories. *IEEE Transactions on Neural Networks, 5*, 730–737.

Marcus, C. M., & Westervelt, R. M. (1989). Dynamics of iterated-map neural networks. *Physical Review A, 40*, 501–504.

Michel, A. N., & Farrell, J. A. (1990). Associative memories via artificial neural networks. *IEEE Control Systems Magazine, 10*, 6–17.

Michel, A. N., Si. J., & Yen, G, (1991). Analysis and synthesis of a class of discrete-time neural networks described on hypercubes. *IEEE Transactions on Neural Networks, 2*, 32–46.

Perfetti, R. (1995). A synthesis procedure for brain-state-in-a-box neural networks. *IEEE Transactions on Neural Networks, 6*, 1071–1080.

Strang, G. (1988). *Linear algebra and its applications* (3rd ed.). Orlando, FL: HBJ Publishers.

Zak, S. H., Lillo, W. E., & Hui, S. (1996). Learning and forgetting in generalized brain-state-in-a-box (GBSB) neural associative memories. *Neural Networks, 9*, 845–854.

Received March 18, 1998; accepted December 30, 1998.