for some $\beta \geq 0$.

The components s_j in (28) are called the *scale factors* and are assumed to be i.i.d. with some probability distribution $p_S(s_j)$ that does not depend on n or d. We assume $s_j > 0$ almost surely. The diagonal scale factor matrix $\mathbf{S}$ is used to scale the powers of the components of $\mathbf{x}$. Specifically, multiplication by $\mathbf{S}^{1/2}$ scales the variance of the jth component of $\mathbf{x}$ by a factor s_j. In this way, the scale factors can be used to capture variations in the power of the components of $\mathbf{x}$ that are known *a priori* at the estimator. Variations in the power of $\mathbf{x}$ that are not known to the estimator should be captured in the distribution of $\mathbf{x}$.

The matrix $\mathbf{A}$ is deterministic, and we evaluate the performance of the relaxed BP algorithm on a deterministic sequence of input and output indices $i = i(n,d)$ and $j = j(n,d)$. The sequence of matrices $\mathbf{A}$ and indices i and j are assumed to satisfy the following conditions:

Assumption 1: Let $\mathbf{A} = \mathbf{A}(n,d)$ be a sequence of deterministic matrices in the factorization of $\Phi = \Phi(n,d)$ in (28). Let $i = i(n,d)$ and $j = j(n,d)$ be a deterministic sequence of indices. Let $t > 0$ be some fixed iteration number of the relaxed BP algorithm. Assume that $\mathbf{A}$, i and j satisfy the following:

(a) For every n and d, (i,j) is an edge in the Tanner graph G associated with Φ.

(b) The *computation subgraphs* $G_i(t)$ and $G_j(t)$ of the Tanner graph taken a depth of $2t$ hops from the output node i and input node j are trees. Precise definitions of these computation subgraphs are given in Appendix D.

(c) All the nodes in the subgraphs $G_i(t)$ and $G_j(t)$ have degrees bounded above by d.

(d) For all output nodes ℓ in the subgraph $G_i(t)$, we have the limits

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d}\sum_{r\in N_{\rm out}(\ell)}|a_{\ell r}|^2 = \beta, \quad (30a)$$

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d^{3/2}}\sum_{r\in N_{\rm out}(\ell)}|a_{\ell r}|^3 = 0. \quad (30b)$$

For all input nodes r in the subgraph $G_j(t)$, we have the limits

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d}\sum_{\ell\in N_{\rm in}(r)}|a_{\ell r}|^2 = 1, \quad (30c)$$

$$\lim_{d\to\infty}\lim_{n\to\infty}\frac{1}{d^{3/2}}\sum_{\ell\in N_{\rm in}(r)}|a_{\ell r}|^3 = 0. \quad (30d)$$

As in [12]–[14], the key assumption is that the Tanner graph G associated with the transform matrix Φ is locally tree-like around the components of interest i and j. The assumption is common in the study of BP algorithms as it makes the messages independent. This local tree-like property is only possible with the graph being sparse. This sparsity assumption is brought out explicitly by bounding the input and output degrees of the Tanner graph.

Assumption 1 uses a deterministic model for the $\mathbf{A}$ as opposed to the random matrix model with i.i.d. components studied in [12]–[14]. The deterministic model simplifies some of the proofs. In particular, the input and output degrees are deterministically bounded as opposed to be bounded on average – which simplifies some of the convergence arguments.

In the large sparse limit analysis, we first let $n \to \infty$ with m growing linearly with n and keeping d fixed. This enables the local-tree like properties. We then let $d \to \infty$, which will enable the use of a Central Limit Theorem approximation.

This order of limits is critical. Unfortunately, to analyze dense matrices, one would like an analysis where d can grow with n. Indeed, if the matrix is completely dense, we would like $d = m(n)$. Unfortunately, the large sparse limit analysis that we rely on here requires that we consider the two limits separately; it thus represents an approximation to the actual problem. Nevertheless, we will see in simulations that large sparse limit analysis appears to predict the behavior with dense matrices as well.

More sophisticated analysis techniques developed recently in [22] enable the study of dense matrices without the order of limits above. One possible avenue of future research would be to see if that analysis can be applied to the relaxed BP algorithm with general (non-AWGN) output channels as well.

B. Large Sparse Limit Convergence

Under the large sparse limit model, define the random vectors

$$\theta^x_{i\leftarrow j}(n,d,t) = (x_j, s_j, \widehat{x}_{i\leftarrow j}(t), \mu^x_{i\leftarrow j}(t)) \quad (31a)$$

$$\theta^x_{j}(n,d,t) = (x_j, s_j, \widehat{x}_{j}(t), \mu^x_{j}(t)) \quad (31b)$$

$$\theta^z_{i\to j}(n,d,t) = (z_{i\to j}, \widehat{z}_{i\to j}(t), \mu^z_{i\to j}(t)) \quad (31c)$$

$$\theta^z_{i}(n,d,t) = (z_i, \widehat{z}_i(t), \mu^z_i(t)), \quad (31d)$$

where the dependence on n and d on the right-hand side of the equations is implicit. Our goal is to describe the large sparse limit behavior of these random vectors.

A key result of [14] is that the large sparse limit behavior of BP is described by a set of simple *state evolution* (SE) equations, which can be described as follows: Given $\mathcal{E}_{\rm in}(q,\mu)$ in Section IV-A, define

$$\overline{\mathcal{E}}_{\rm in}(\mu,s) = \mathbf{E}\left[\mathcal{E}_{\rm in}(q,\mu/s)|s\right] \quad (32a)$$

$$\overline{\mathcal{E}}_{\rm in}(\mu) = \mathbf{E}\left[s\mathcal{E}_{\rm in}(q,\mu/s)\right], \quad (32b)$$

where the expectation is taken over the scalar random variables $s \sim p_S(s)$ and q given by (14) with $x \sim p_X(x)$. We will call $\overline{\mathcal{E}}_{\rm in}(\mu)$ the *input node MSE function*. In addition to the works [13], [14], this function appeared in Guo and Verdú's replica analysis of MSE estimation [16] and related works [28], [29]. Variants also appear in the analysis of the AMP algorithm [11], [22].

At the output node, let

$$\mu^z_{\rm init} = \beta\mathbf{E}(s)\mu^x_{\rm init}, \quad (33)$$

where $\mu^x_{\rm init}$ is variance of x_j according to the prior $p_X(x_j)$, and the expectation is over $s \sim p_S(s)$. Then, for $\mu \leq \mu^z_{\rm init}$, Guo and Wang [14] define the *output node MSE function* as

$$\overline{\mathcal{E}}_{\rm out}(\mu) = \frac{1}{\mathbf{E}\left[D_2(y,\widehat{z},\mu)\right]}, \quad (34)$$