Abstract—Soft-input/soft-output (SISO) algorithms have been widely used for iterative detection in various applications since this technique was introduced for decoding Turbo codes. However, the complexity of the SISO algorithms is a major concern in the detector implementation. In this paper, a novel way to simplify the SISO algorithms is proposed based on the concept of state reduction via decision feedback. The resulting complexity reduction is exponential in the number of feedback taps. The proposed low-complexity SISO algorithm can be applied directly in place of the standard SISO (e.g., RSSE in [17], DDFSE in [18] and RSSE-RTS in [19]). In this paper, the decision feedback technique is applied to SISO algorithms to reduce the complexity. Since the SISO algorithms include a forward and a backward recursion, truncated states are properly defined and “survivor” paths are generated in both recursions separately. Also, the desired soft output is yielded in a simplified way. The resulting complexity reduction is exponential in the number of feedback taps. We will refer to the proposed algorithms as reduced-state (RS) SISO algorithms. The proposed RS-SISO has the exactly same input/output interface as the standard SISO algorithm [1,2]. Therefore, the RS-SISO can be used to replace the standard SISO directly. In addition, the RS-SISO algorithm may feed its own soft output back to its input port (which we refer to as self-iteration) to refine the soft output. Therefore, in the case of sequence detection, the RS-SISO algorithm can also replace a hard-decision algorithm (e.g., VA, DDFSE) and give robust and excellent performance.

This paper is organized as follows. The FSM model and notations are given in Section II. Two versions of RS-SISO algorithm are presented in Section III along with the specific concern for iterative applications. Finally, the RS-SISO algorithm is compared numerically to some existing algorithms in two different transmission systems in Section IV. Concluding remarks are contained in Section V.

II. AN FSM MODEL AND NOTATIONS

We will only consider the time-invariant FSM in this paper. In this case, a FSM can be described by a regular trellis diagram. Such a trellis consists of a set of states \( S = \{S_1, S_2, \ldots, S_N\} \). The state of the FSM at time \( k \) is \( s_k \in S \). The trellis transition \( t \) is deterministically driven by a date source that outputs sequence \( \{a_1, a_2, \ldots, a_K\} \) with symbol \( a_k \) assumed to be independently drawn from the \( M \)-ary alphabet \( \mathcal{A} = \{A_0, A_1, \ldots, A_{M-1}\} \). Therefore, in an FSM, each transition \( t \) is associated with a starting state \( s^t(i) \), an ending state \( s^t(t) \), an input symbol \( a(t) \) and an output symbol \( x(t) \). In this paper, for the simplicity of presentation, we only consider the FSM whose state is defined as \( s_k = (a_{k-L}, a_{k-L+1}, \ldots, a_{k-1}) \). Such an FSM is said to have memory length \( L \) and the number of state \( N = M^L \). Consequently, for a transition \( t_k = (a_k, \ldots, a_k) \), we have \( s^t(t_k) = (a_{k-L}, \ldots, a_{k-1}) \) and \( s^t(t) = (a_{k-L+1}, \ldots, a_k) \) and the output symbol \( x(t_k) = x(t) \). For a general ISI channel, the function \( x(\cdot) \) can be a 1-to-1 mapping. For a trellis coded modulation (TCM) scheme, it is usually an \( n \)-to-1 mapping.

1This work was supported in part by a U.S. Army SBIR contract to ViaSat, Inc. (DAAB07-98-C-K004).
III. LOW COMPLEXITY RS-SISO ALGORITHMS

A. A-Posteriori Probability RS-SISO

In a RS-SISO, we truncate the memory of the FSM and rebuild the corresponding trellis. The truncated state at time $k$ is defined by $v_k = \{a_{k-L}, a_{k-L+1}, \ldots, a_{k-1}\}$, where $L_1 \leq L$ and define $L_2 = L - L_1$. If $L_1 = L$, then $v_k = s_k$, and the following derivation will simply result in the standard SISO algorithm. It will be shown that the standard SISO is $M^{L_2}$ times more complex than the RS-SISO. First, we define a few state sets for future use:

- $\mathcal{F}(j) = \{i : V_i \rightarrow V_j \text{ is an allowable forward transition}\}$ (1)
- $\mathcal{B}(i) = \{j : V_j \leftarrow V_i \text{ is an allowable backward transition}\}$ (2)
- $\mathcal{C}(m) = \{j : v_{k+1} = V_j \text{ is consistent with } a_k = A_m\}$ (3)

As in the standard APP-SISO algorithm, the following key quantities are defined for the APP RS-SISO algorithm recursively as:

$$\alpha_k(j) = \sum_{i \in \mathcal{F}(j)} \alpha_{k-1}(i) \gamma_k(i,j)$$
$$\beta_k(i) = \sum_{j \in \mathcal{B}(i)} \beta_{k+1}(j) \gamma_{k+1}(i,j)$$

with initial values as $\alpha_0(i) = \Pr(v_0 = V_i)$ and $\beta_{k+1}(j) = \Pr(v_{k+1} = V_j)$. For example, at time $k = 0$, if the initial state of the FSM is $V_l$, then $\alpha_0(l) = 1, \alpha_0(j) = 0, j \neq l$. If no knowledge about $v_0$ is available, then $\alpha_0(i) = M^{-L_1}$. Note that the APP algorithm does not generate “survivors” naturally. One reasonable definition is: for a state at time $k$, the forward (or backward) survivor state is the state contributing most in the summation of (4) (or (5)). Other reasonable definitions of the survivor are also applicable. Consequently, associated with each truncated state $v_k = V_l$, we can obtain a truncated survivor path in the forward recursion as $\tilde{\alpha}_k^f(i) = \{\tilde{a}_{k-L}^f(i), \tilde{a}_{k-L+1}^f(i), \ldots, \tilde{a}_{k-1}^f(i)\}$, and a backward truncated survivor path as $\tilde{\alpha}_k^b(i) = \{\tilde{a}_{k-1}^b(i), \tilde{a}_{k+1}^b(i), \ldots, \tilde{a}_{k+L_2-1}^b(i)\}$ (Fig. 1).

Based on the definition of survivor paths, we define the quantities associated with $t_k(i,j) = (v_k = V_l, v_{k+1} = V_j)$ in both forward and backward recursion as (Fig. 1):

$$\gamma_k^f(i,j) = \Pr[x_k = (\tilde{a}_k^f(i), t_k(i,j))] \Pr[\tilde{a}_k^f(i,j)]$$
$$\gamma_k^b(i,j) = \Pr[x_{k+L_2} = (\tilde{a}_k^b(i), t_k(i,j))] \Pr[\tilde{a}_k^b(i,j)]$$

where $a_k(i,j)$ is the value of $a_k$ associated with a forward transition $V_l \rightarrow V_j$. When $L_2 = 0$, $\gamma_k^b(i,j) = \gamma_k^b(i,j)$, yielding the standard APP algorithm.

Fig. 1. The construction of the complete (a) forward and (b) backward transition and completion step at time $k$ in the RS-SISO algorithm.

The complexity of the forward and backward recursion in (4) and (5) is primarily determined by the number of states, i.e., $M^{L_2}$. Compared to the standard APP-SISO algorithm, the complexity of proposed APP RS-SISO is reduced by $M^{L_2}$ times. Finally, two types of soft output for $a_k$ can be obtained by

$$P^a(a_k = A_m) = \sum_{j \in \mathcal{C}(m)} \alpha_k(j) \beta_k(j)$$ (8)
$$P^b(a_k = A_m) = c \Pr^a(a_k = A_m) / \Pr^a(a_k = A_j)$$ (9)

where $c$ is a normalization constant. Due to the truncation of state, $P^a(x_k)$ cannot be yielded directly as in the standard SISO algorithm. However, we can still obtain $P^a(x_k)$ approximately. One simple way is:

$$P^a(x_k = a(t_k)) = \prod_{l=k-L}^k \Pr^a(a_k : t_k)$$ (10)

which can work fairly well [2]. It is notable that (10) can be calculated recursively due to the temporal relationship between $t_k$ and $t_{k+1}$. $P^a(\cdot)$ can be viewed as the (approximate) a-posteriori probability normalized to the a-priori probability and is usually called the “extrinsic” information [1]. If further iteration is necessary, one should feedback $P^a(\cdot)$. Otherwise, the hard decision may be made by the rule: $a_k = A_m$ if $P^a(a_k = A_m) > P^a(a_k = A_j)$ for all $0 \leq j \leq M - 1$. The completion step in (8) is also illustrated in Fig. 1. Obviously, this RS-SISO algorithm has the exactly same input/output interface as the standard SISO algorithm. It can be observed that when the RS-SISO generates the soft output for $a_k$, the input information in time interval $[k + 1, k + L_2]$ has not been used (Fig. 1). Other completion approaches are feasible for a RS-SISO which use all the observations, but this may result in an algorithm with complexity dominated by the “full-state” completion. Regardless of the completion technique used, the RS-SISO is sub-optimal due to the decision feedback used in the forward and backward recursions. The simplified completion scheme defined in (8)–(10) results in further performance degradation and complexity reduction. In order to improve the performance of RS-SISO, we introduce the concept of self-iteration for the RS-SISO algorithm. By feeding its output back to its own input port several times, the RS-SISO can fuse the information in the time interval $[k + 1, k + L_2]$ into the final soft output. It will be shown in Section IV that the self-iteration can improve the performance significantly.

B. Minimum Sequence Metric RS-SISO

In the case of an additive white Gaussian noise (AWGN), we define the metric counterpart to (6) and (7) as

$$\lambda_k^f(i,j) = M^f[x_k = (\tilde{a}_k^f(i), t_k(i,j))] + M^a(a_k(i,j))$$ (11)
\[ \lambda^b_k(i,j) = M^b[x_{k+L-1} = x(t_k(i,j), \hat{a}^b_{k-1}(j)) + M^b[a_k(i,j)] \] respectively, where the metric \( M(w) = -\ln(P(w)) \). Then, the sequence metric associated with a symbol sequence \( \hat{a}^b_{k-1} \), or the equivalent state sequence \( v^b_{k+1} \), can be defined as

\[ \Lambda^b(v^b_{k+1}) = \sum_{i=k}^{k-1} \lambda^b(i,j) \] (13)

and the key quantities for MSM RS-SISO are defined as

\[ \delta_k(j) = \min_{i \in F(j)} \lambda^b_k(i,j) + \lambda^b_{k-1}(i,j) \] (17)

\[ \eta_k(i) = \min_{i \in F(i)} (\delta_k(j) + \lambda^b_k(i,j)) \] (18)

\[ M^\sigma(a_k = A_m) = \min_{j \in G(m)} (\delta_k(j) + \eta_k(j)) \] (19)

\[ M^\sigma(a_k = A_m) = M^\sigma(a_k = A_m) - M^\sigma(a_k = A_m) \] (20)

\[ M^\sigma(x_k = x(t_k)) = \sum_{i=k}^{k} M^\sigma(a_k : t_k) \] (21)

Consequently, the forward and backward recursion and the completion steps of MSM RS-SISO can be readily written down as:

\[ \delta_{k+1}(j) = \min_{i \in F(j)} [\delta_k(i) + \lambda^b_{k-1}(i,j)] \] (17)

\[ \eta_k(i) = \min_{i \in F(i)} (\delta_k(j) + \lambda^b_{k-1}(i,j)) \] (18)

\[ M^\sigma(a_k = A_m) = \min_{j \in G(m)} (\delta_k(j) + \eta_k(j)) \] (19)

\[ M^\sigma(a_k = A_m) = M^\sigma(a_k = A_m) - M^\sigma(a_k = A_m) \] (20)

\[ M^\sigma(x_k = x(t_k)) = \sum_{i=k}^{k} M^\sigma(a_k : t_k) \] (21)

Since the forward and backward recursion have the same computational structure as the recursion in the VA, the MSM RS-SISO can obtain the survivor paths used in (11) and (12) just as in the VA. Again, one feeds back \( M^\sigma(\cdot) \) if further iterations are needed, and uses \( M^\sigma(a_k) \) to make the hard decision by the rule: \( \hat{a}_k = A_m \) if \( M^\sigma(a_k = A_m) \leq M^\sigma(a_k = A_j) \) for all \( 0 \leq j \leq M - 1 \).

Compared to the APP version, the MSM version only involves summation and comparison operations. Computationally, it is much simpler than the APP version. By replacing \( \min(\cdot) \) by \( \min^\prime(\cdot) \) in the MSM algorithm [1,3,20], one directly obtain the corresponding log-APP algorithm. However, the way one defines the survivor path in a log-APP algorithm must be specified (i.e., as we have done in Section III-A). The numerical experiments have shown that the MSM SISO algorithm performs almost as well as its APP counterpart [9,13]. Similarly, compared to the standard MSM algorithm, the MSM RS-SISO has the same input/output interface, and a complexity \( M^{L^2} \) times smaller.

C. Iterative Detection Using RS-SISO

An iterative detection network consists of SISO modules and soft information exchange rules and schedules [1]. The soft information is circulated inside this network several times before the hard decisions are made. Usually, the maximum iteration number can be estimated numerically. For \( I \) iterations, the complexity of iterative detector employing the RS-SISO algorithm is proportional to \( IM^{L-1} \).

IV. NUMERICAL SIMULATION

Two types of transmission system are used to test the performance of RS-SISO algorithms: an ISI/ AWGN channel and a TCM/ISI/ AWGN channel. Because of the simplicity and good performance of MSM SISO algorithms, only the results for the MSM RS-SISO will be presented here. It can be expected that the APP RS-SISO should perform no worse than the MSM RS-SISO [13].

A. ISI/ AWGN channels

Two 12-tap \( L = 11 \) ISI/ AWGN channels are used in this test (Fig. 2(a)). Channel A has equal entries and Channel B is selected to be \( \{c,2c,\cdots,12c\} \). Both channels are normalized to have unit power. The transmitter uses the BPSK signaling scheme (i.e., \( a_k = \pm\sqrt{E_b} \)). The output of the ISI channel is then corrupted by an AWGN \( n_k \) with \( E[n_k^2] = N_0/2 \). For comparison, we also run two hard decision algorithms: the VA [6] and DDFSE [18] on the same channels (Fig. 2(b)). Similar to in the RS-SISO, a truncated state of length \( L_1 \) is defined in the DDFSE.

![Fig. 2](image)

Fig. 2. (a) The tested ISI/ AWGN channel and (b-c) two types of detector.

![Fig. 3](image)

Fig. 3. The convergence property of an iterative detector using the MSM RS-SISO algorithm \( L_1 = 2 \).

We first describe the convergence properties of detectors employing an MSM RS-SISO with self-iteration. As illustrated in Fig. 3, the convergence occurs after 4–5 iterations. In Fig. 4, several algorithms are compared. An index of the complexity is defined as the product of the number of transitions \( M^{L+1} \), the self-iteration number \( I \) and the recursion number \( r \). For the forward-only VA \( r = 1 \), while \( r = 2 \) for the forward/backward SISO. In Fig. 4 and the following figures, this complexity index is shown for each algorithm used. The performance of the VA with \( 2^{11} = 2,048 \) states is presented as a baseline. Note that thresholding the soft-output of the full-state MSM-SISO yields the same data estimates as the VA, namely that of maximum likelihood sequence detection (MLSD), while thresholding the RS MSM-SISO does not yield the same result as the DDFSE using the same \( L_1 \). First, we notice that the DDFSE performs roughly 3 dB worse...
than VA at a bit error rate (BER) of $10^{-4}$ when $L_1 = 5$, i.e., the DDFSE uses $2^5 = 32$ states. Without self-iteration, the RS-SISO with $L_1 = 2$ performs only 0.3 dB better than the DDFSE with a 4 times smaller complexity. However, if the self-iteration is used, after only 4 iterations, the performance of RS-SISO is improved by 1.9 dB, and is only 0.8 dB away from that of the VA. The complexity index indicates that the RS-SISO is 64 times simpler than the VA with a performance degradation of only $< 1$ dB while it outperforms a DDFSE of similar complexity by 2.2 dB.

Due to the bidirectional recursion of RS-SISO, its robustness to non-minimum phase channels is expected. However, for the DDFSE, the non-symmetric structure (only a forward recursion is used) degrades its performance greatly when the channel is of non-minimum phase. Channel B is such a channel. Channel B' is defined as the time-reversed version of Channel B, i.e., \{12c, 11c, \ldots , c\}. The simulation results in Fig. 5 clearly show that the DDFSE with $L_1 = 5$ virtually fails for Channel B but works well for Channel B'. However, the iterative detector based on RS-SISO with the same complexity ($L_1 = 2$ and $I = 4$) performs nearly optimal for both Channel B and B'. Note that for both Channel B and B', either the VA or the MSM RS-SISO performs same. For MSM RS-SISO this is due to its bidirectional architecture; while for VA this is because it is optimal in the sense of MLSD and Channel B and B' have the exactly same distance spectrum.

B. An ISI/AWGN channel with TCM signaling

An 8-state, rate $R = 2/3$ Ungerboeck 8-PSK TCM code (Figure 9 in [21]) is used in this test. The 8-PSK signals from the TCM encoder are fed into a 32 × 32 block interleaver. The interleaved 8-PSK signals pass through a 5-tap ($L = 4$) ISI channel with equal entries (normalized to unit power), and the output is corrupted by a white complex circular Gaussian noise $n_k$ with $E[n_k^2] = N_0/2$. This system is illustrated in Fig. 6(a).

It is too complex to implement a Viterbi detector by considering the concatenated TCM encoder and ISI channel as a single FSM. Alternatively, an effective approach is to build a “detector”

![Figure 4](image1.png)

Fig. 4. The performance comparison of various detection algorithms for Channel A. The number attached to each curve is the complexity index of the corresponding algorithm.

![Figure 5](image2.png)

Fig. 5. The robustness of the iterative detector using the MSM RS-SISO algorithm.

![Figure 6](image3.png)

Fig. 6. (a) The tested TCM/ISI/AWGN channel and (b-d) three types of detector (hard or soft) for each subsystem separately [1,14]. As shown in Fig. 6(b), one can use the VA as both the inner and outer processor, but a poor performance is expected because the TCM code is decoded with hard decisions. Replacing the inner VA by a SISO (Fig. 6(c)) can improve the overall performance since more reliable Euclidean distance is used at the outer Viterbi detector. Moreover, replacing the outer VA by a SISO and using iterative detection, the overall performance can approach that of the optimal performance [13,14]. However, even by treating the subsystems separately, the complexity of the inner detector is prohibitive. For example, in the above system, the inner FSM has $8^4 = 4,096$ states and $8^4 = 32,768$ transitions. Therefore, we consider the RS-SISO for the inner SISO.

Due to the sub-optimality of the RS-SISO and the concatenated detection structure [13,14], both the self-iteration of the inner RS-SISO (so called inner iteration) and the outer iteration (see Fig. 6(d)) becomes necessary for an effective detector. Therefore, besides the number of outer iterations $I_o$ as for general iterative schemes, there is another design parameter for this specific detector: the number of the inner iterations $I_i$. Before feeding the soft information on the 8-PSK symbols to the outer SISO, the inner RS-SISO may conduct several inner iterations to improve the
Fig. 7 shows the simulation results of various detection schemes. There is a 5 dB gain in $E_b/N_0$ at a BER of $10^{-4}$ by replacing the VA with a MSM-SISO at the inner stage. When replacing the inner VA by a MSM RS-SISO with $L_1 = 2$ ($I_1 = 1$), only 0.3 dB gain is obtained. This means that without any performance degradation, the detector complexity is reduced by 32 times. Two types of iterative detector using the MSM RS-SISO algorithm are tested. Both detector converges in only 4-5 iterations (results are not shown here). One detector uses an MSM RS-SISO with $L_1 = 1$ (8 states), and 3 inner iterations ($I_o = 3$). The hard decisions are made after five outer iterations ($I_o = 5$). Compared to the SISO–VA scheme, the performance degradation is only 1.1 dB while the complexity saving is 34 times. In order to obtain better performance, the other iterative detector uses $L_1 = 2$ and $I_1 = 3$. After four outer iterations ($I_o = 4$), a 0.3 dB gain is obtained over the SISO-VA approach while the complexity saving is roughly 5 times. For comparison, note that in this application any reduced complexity hard-decision processor (e.g., RSSE [17], DDFSE [18]) will perform worse than the VA–VA scheme in Fig. 7.

Fig. 7. The performance comparison of various detection algorithms for the TCM/ISI channel.

V. CONCLUDING REMARKS

In this paper, the RS-SISO algorithm for the simple FSW was presented and shown to provide both significant complexity reduction and excellent performance. It is conceptually straightforward to extend this algorithm to any FSW to which the concept of state reduction is applicable. It is notable that there exist FSWs with very special structure (i.e., many transitions, but only few outputs), e.g., a TCM encoder, such that it may be able to calculate $P^o(x_k)$ or $M^o(x_k)$ very efficiently if such soft output is required. Also, besides the feedback technique used in this paper, other state reduction (e.g., Ungerboeck-like set partitioning principle [17,21]) techniques may be employed to derive other forms of the RS-SISO. Due to the sub-optimality of the RS-SISO algorithm, the self-iteration is necessary in many cases. The flexibility of RS-SISO provides many options to trade the complexity/performance. Since the proposed RS-SISO has the same input/output interface as the standard SISO, it can be widely applied to many applications other than those presented in this paper.

REFERENCES


