Four Orders of Magnitude Cost Reduction in the Viterbi Algorithm for GSM MIMO Demodulation

William A. Gardner, David A. Burgess, Chris W. Reed
Statistical Signal Processing, Inc.
1909 Jefferson Street, Napa, CA 94559
wag@statsig.com

Abstract

It is shown that multiple complementary techniques for trading BER performance for cost reduction in the Viterbi algorithm for multiple-input/multiple-output joint demodulation can be merged to obtain substantial and efficient cost reduction. Quantitative results for application to the separation of GSM and IS-136 cochannel signals are presented. It is shown that up to six signals can be separated at a GSM basestation with BER of 10% before decoding gain with only a single dual-polarized antenna and $E_b/N_0 = 10$ dB in real time on a Compaq 1 GHz ES-40 server, with cost savings of up to four orders of magnitude. It is also shown that for a specified BER and number of signals, the cost of MIMO demodulation decreases as the number of sensors increases. Finally, it is shown that for a fixed computational cost, BER decreases by a factor of eight, on average, for each sensor added.

1 Introduction

It is a well-known fact that a joint demodulator for a single sensor can separate two cochannel interfering (CCI) signals and sometimes more depending on SNR, SIR, and other parameters, such as fraction of symbols that are known at the receiver. It follows that, more generally, a joint demodulator for multiple sensors can separate a number of signals that exceeds the number of sensors, and offers a distinct advantage over the linear combining of sensor outputs followed by demodulation. This advantage accrues primarily from more effective utilization of the increased channel diversity resulting from multiple sensors, but little is known about how many signals can be separated with a specified BER and number of sensors. The primary reason for the lack of progress in this area of multiple-input/multiple-output (MIMO) demodulation is computational cost. Even if the optimally efficient Viterbi algorithm is used to implement the MIMO demodulator, computational cost still grows exponentially in the number of signals. The computational cost can be so high that even simulations present a substantial challenge. Consequently, efficient means for trading off BER performance for computational cost reduction are required in order to move forward with MIMO demodulation R&D.

The purpose of this paper is to report on progress made on developing efficient techniques for trading BER performance for cost reduction. See also [1].

2 MIMO Demodulation

The MIMO demodulator implements a joint least-squares fit of all the interfering signals to all the individual sensor outputs. The sensor outputs are not linearly combined. They are simply used to form the MIMO model-fitting cost function that is minimized by a trellis search over the multiple symbol sequences. This trellis search is optimally implemented by the Viterbi algorithm.

The joint demodulators described in this paper employ three distinct and complementary techniques for decreasing BER and/or trading BER for reduced computational cost, relative to the baseline Viterbi algorithm. All three techniques can be used in conjunction, simultaneously, and together they provide a powerful means for computation reduction and/or BER reduction.

The new MIMO demodulator is a modification of the conventional Viterbi algorithm for multi-sensor JMLSD (joint maximum-likelihood sequence detection). The normal Viterbi algorithm [3] has two phases: spawning and pruning. In the spawning phase, existing joint symbol sequences are extended with permutations of possible input symbols to create a new set of candidate joint sequences. In the pruning phase, this set of candidate sequences is pruned down to a survivor set according to dynamic programming principles and an analysis of their sum-of-squared errors relative to the input. The pruning process is optimum in the sense that it is guaranteed not to remove any candidate sequence that lies along the maximally likely symbol sequence path through the trellis. The survivors of the pruning are then spawned again and pruned again with the next symbol's worth of samples of the input signal, and the algorithm repeats through its phases.

The first modification used in our MIMO demodulator applies constraints in the spawning phase. Known symbol values in the signals (due to header sequences, midambles, etc.) are taken into account to prevent the spawning of candidate sequences that will violate the signal format. Constrained spawning can improve the BER performance of the Viterbi algorithm because it exploits some of the non-random nature of communications signals. It yields the true MLSE solution for the more accurate model that recognizes the existence of known symbols. This non-statistical constrained spawning produces both a decrease in computational cost and an improvement in BER performance by eliminating candidate sequences that are known to be invalid. The degree of improvement depends on the amount of known structure in the signals being demodulated. As an example, approximately 20% of the symbols are known in GSM signals, and somewhat fewer (closer to 10%) are known in IS-136 signals.

The second and more important modification used in our MIMO demodulator adds a third phase to the conventional two-phase (spawning/pruning) Viterbi algorithm. In this third phase, many sequences that survived the pruning phase are removed with selection based on an analysis of the distribution of changes in sum-of-squared-error values. For the objective of trading off BER for computational cost in the most efficient manner possible, this statistical thinning (ST) algorithm reported on here has several advantages over all prior approaches that we have been able to find in the literature, including well-known reduced-state sequence estimation (RSSE) methods, the M-algorithm, and the T-algorithm, as explained in [1].

Typically, RSSE techniques reduce computation by grouping the trellis states into a smaller number of state classes, and allowing only one survivor sequence in each class. The partitioning algorithm used to construct these state classes is, in effect, changing the structure of the trellis. This change can greatly reduce computation, but must be planned carefully to maintain BER performance. There does not seem to be any work on class definition that is directly relevant to joint demodulation with general channels, although the approach described in [4] might be applicable here.

The third modification used in our MIMO demodulator introduces per-survivor decision feedback [5]. In the conventional Viterbi algorithm, the length of the pulse model P, is $K + 1$, where K is the number of symbols of channel memory. (The additional symbol is the current one.) By making the model parameter P independent of the algorithm parameter K, we can reduce K while still accurately modeling the channel pulse. This is equivalent to feeding tentatively decided symbol values (those decisions within survivors) back into the pulse model. For any given values of K and P, we have $P - K - 1$ feedback symbols. Because the size of the candidate set is an exponential function of $K+1$, a reduction in K produces a great reduction in computational cost; moreover, the BER performance of the demodulator can often be preserved by keeping P large enough. Likewise, increasing P can sometimes improve the BER performance of a joint demodulator without significantly increasing the algorithm's computational cost. In addition to the per-survivor decision feedback, the MIMO demodulators evaluated in this paper also use path truncation at $5P$ symbols.

3 Simulations

Experiment 1: One and Two Sensors.

The GSM signal environments used in this study were constructed using cochannel combinations of six signals, all of which were beacon channels, fully loaded (no dummy bursts), with pseudorandom bits filling the data fields of each normal burst, carrier frequency offsets ranging from 5 Hz to 20 Hz, and timing offsets ranging from 1 symbol to 738.46 symbols. These signals were combined with powers in 0 dB and 3 dB steps to form co-channel interference test environments. The GSM signal generator uses the known approximation of GMSK by complex PAM. The error of this model is about -22 dB. Test cases where this modeling error may be significant are noted. The middle symbol of the GSM pulse contains only about 50% of the pulse energy. The middle three symbols contain over 95% of the energy. Most test cases were run with $P = 4$ to ensure a full pulse support and to allow for timing offsets among CCI signals of up to ± 0.5 symbol. All of the GSM test environments were generated with a sample rate of four times the GSM symbol rate, and with additive white Gaussian noise extending over the full sampled bandwidth.

For all tests, the noise power was set at 10 dB below the strongest signal in the environment. TSNR figures indicate the ratio of total signal power to in-band noise power, where the GSM bandwidth is assumed to be the same as the symbol rate. The relationship between TSNR and primary-signal E_b/N_0 depends on the number of CCI and their relative powers. For 0 dB SIR among all CCI, individual SNRs are 4.8 dB, 6 dB, 7 dB, and 7.8 dB below TSNR for 3, 4, 5, and 6 CCI, respectively.

All GSM tests were run for 54,166 symbol periods (0.2 sec). This test length was chosen because it allowed an integer number of cycles for all of the frequency offsets chosen for these test. The T-algorithm performance curves are not included in this paper because of their similarity to those for the ST algorithm in the environments studied here where the only non-stationarity is due to differential Doppler shifts.

Bit-errors were detected by comparing the joint demodulator output against the modulator input bits. This comparison was performed for the data fields only. Fixed bit sequences defined by the GSM standard (and exploited by the constrained Viterbi algorithm) are not included in BER calculations.

Because of the similarity of the algorithms under consideration, we can justify expressing and comparing their computational costs in terms of the size of each candidate set scaled by the number of sensors, M. This policy is based on three assumptions:

1. Running time is proportional to the number of candidates, N, in software implementations. (In practice, running time is dominated by a term linear in N.)

2. For a given number of sensors, the per-candidate processing cost is about the same for all of these algorithms.

3. Computational cost rises linearly with the number of sensors, M. (In practice, running time rises sub-linearly in M.)

Some of the results are presented in Figures 1, 2 and 3 as graphs that plot average BER against average number of candidates. The "sweet spot" of such a plot is at the origin – 0 BER for 0 cost. The effectiveness of a survivor-reduced joint demodulator can be judged by how close its performance curve comes to this point. Where applicable, each graph is also marked with a horizontal dashed line that shows the approximate BER limit for good voice copy; and a vertical dashed line that shows the computational limit for real-time joint demodulation on a Compaq 1 GHz ES-40 server. The following abbreviations are used to label curves on the graphs: **VA** is the Viterbi algorithm, **C-VA** is the constrained Viterbi algorithm, **M** is the M-algorithm, and **ST** is the statistical thinning algorithm. Although not explicitly denoted in these labels, per-survivor feedback is used whenever $K \leq P - 1$.

For 3-CCI, 4-CCI, and 5-CCI GSM with $K = 2$, the cost of the Viterbi algorithm is 512, 4096, and 32,768 candidates per symbol period, respectively. Running

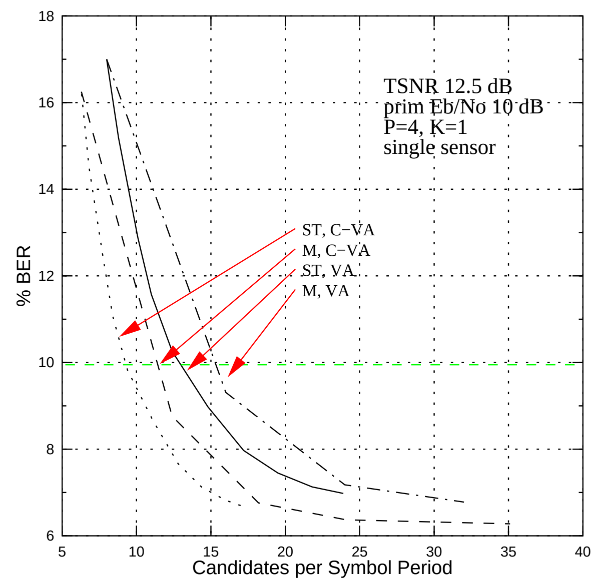

Figure 1: Results for 3-CCI @ 3 dB Spacing, $K = 2$, Single Sensor.

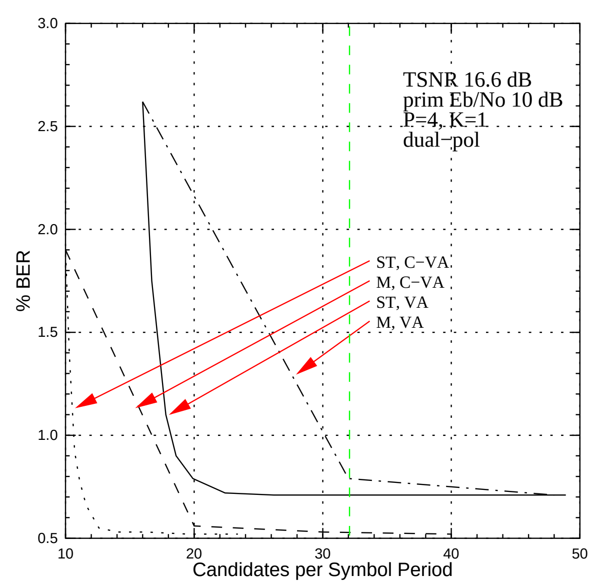

Figure 2: Results for 4-CCI @ 0 dB Spacing, K = 2, Dual-Pol

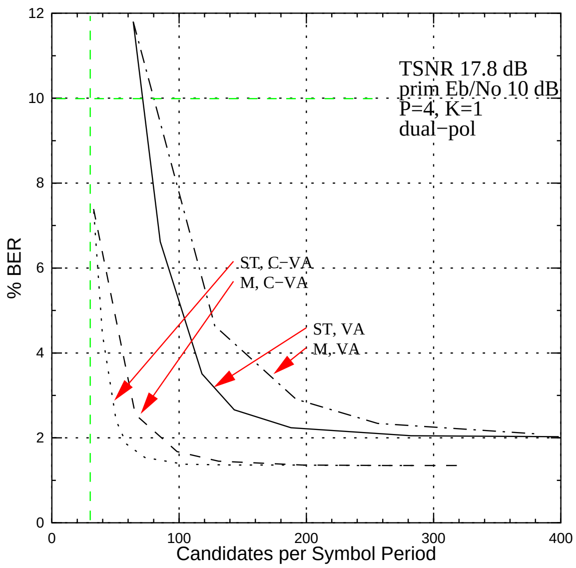

Figure 3: Results for 6-CCI @ 0 dB Spacing, $K = 2$, Dual-Pol

# of CCI	SIR Step (dB)	Prim. E_b/N_0 (dB)	TSNR (dB)	Min. BER %	Avg. BER %	Max. BER %
3	0	4.2	9	15.1	15.7	16.2
3	0	7.2	12	7.71	8.68	9.32
3	0	10.2	15	3.28	4.51	5.21
3	0	13.2	18 *	1.24	2.47	3.12
3	3	6.6	9	7.56	14.5	21.0
3	3	9.6	12	3.39	7.0	10.1
3	3	12.6	15	0.96	2.33	3.37
3	3	15.6	18	0.23	0.73	1.04
4	0	6.0	12	17.2	18.8	19.5
4	0	9.0	15	9.04	11.7	12.7
4	0	12.0	18 *	4.31	7.20	8.32
4	0	15.0	21 *	1.87	4.83	6.06
4	3	9.3	12	7.11	17.2	27.6
4	3	12.3	15	3.90	10.0	16.4
4	3	15.3	18	1.57	4.52	7.49
4	3	18.3	21 *	0.46	1.76	3.06
4	0	6.0	12	17.2	18.8	19.5
4	0	9.0	15	9.04	11.7	12.7
4	0	12.0	18 *	4.31	7.20	8.32
4	0	15.0	21 *	1.87	4.83	6.06
3	3	6.6	9	7.56	14.5	21.0
3	3	9.6	12	3.39	7.0	10.1
3	3	12.6	15	0.96	2.33	3.37
3	3	15.6	18	0.23	0.73	1.04

Table 1: BER performance baselines for 3-CCI, 4-CCI, and 5-CCI GSM environments. * indicates those test cases where PAM approximation error for GSM may be comparable to the noise level, thus causing an inflation of BER. BER expressed in percentages.

P	L	M	Cost	Savings
1	3	1	6.5	9.8
1	3	2	6.0	11
2	3	1	13	39
2	3	2	6.5	79
1	4	1	30	8.5
1	4	2	10 *	26
2	4	1	50	82
2	4	2	14	290
1	5	2	19 *	54
2	5	2	30	1,100
1	6	2	25 *	160
2	6	2	60	4,400

Table 2: Computational cost of demodulation with the ST-augmented constrained Viterbi algorithm. In each case, primary signal E_b/N_0 is 10 dB with equal-power signals. Target BER is 10%. * indicates cases were actual BER was less than 10%. Cost is in candidates per symbol period. Savings is cost relative to cost of conventional Viterbi algorithm with the same P, L, and M. Cost values are rounded to 2 significant digits.

the Viterbi algorithm for a single sensor, without constraints, decision feedback, or survivor reduction, we get the BER performance figures shown in Table 3.

From this table, we can see that 4-CCI GSM environments are only marginally copyable (BER $\leq$ 10%) with a single sensor at 10 dB E_b/N_0, and 5-CCI GSM environments are not copyable at all with a single sensor at 10 dB E_b/N_0. For the 6-CCI cases, the cumulative error in the GSM PAM approximation can be as large as -14 dB relative to the total signal power. At the same time, 6-CCI joint demodulation requires a TSNR on the order of 18 dB. It is difficult to find 6-CCI GSM environments that can produce less than 10% BER from a single sensor, regardless of E_b/N_0 and TSNR. For this reason, single sensor 6-CCI test cases and baselines are omitted from this paper. On the other hand, dual-pol processing makes 6-CCI environments copyable, although the computational requirements are high enough to make this copy only marginally feasible with present-day hardware.

For comparison, Table 2 shows computational costs for the ST-augmented constrained Viterbi algorithm, operating at a primary signal E_b/N_0 of 10 dB and a fixed target BER of 10%. The right-most columns of this table shows that cost savings of this modified Viterbi demodulator relative to the conventional Viterbi algorithm. The savings factors range from 5.3 to 4,400.

Dual-Polarization: The most striking effect of dual-pol joint demodulation is the improvement in BER. In the 4-CCI cases, environments that were marginal at best with single sensors yielded BERs $\leq 2\%$ when dual-pol processing was applied. In the 5-CCI and 6-CCI cases, dual polarization processing turned impossible environments into copyable ones with reasonably low BER.

For single and dual-pol processing of environments with 3 to 5 equal-power CCI GSM signals, we find that the BER improvements from dual-pol joint demodulation are equivalent to a 6 dB to 12 dB improvement in E_b/N_0. This is considerably better than the 3 dB improvement one might expect from linear combining of two sensors by simple addition.

Another effect of dual-pol processing is that it allows survivor reduction to operate more efficiently, yielding smaller candidate sets. This happens because the additional channel capacity makes minimum-cost sequences more distinct, reducing the difficulty of the joint demodulation problem. Because the candidate sets can shrink by more than half, the total cost of dual-pol demodulation can be lower than that of single-sensor demodulation, even though the per-candidate cost may be twice as high.

Per-Survivor Decision Feedback: Changing K from 2 to 1 degraded single-sensor BER by roughly 1% in copyable environments. For dual-pol environments, BER losses were $< 1\%$. For the full VA, going from $K = 2$ to $K = 1$ speeds the algorithm by a factor of 2^L. When used in conjunction with survivor reduction, we observed speedups, due to reduction of K by one, by roughly a factor of 2.

Survivor Reduction: In all cases, statistical thinning matched or exceeded the performance of the M-algorithm in terms of computational efficiency. The difference was especially pronounced when survivor reduction was used in conjunction with significant decision feedback ($K = 1$) or dual-pol processing.

Experiment 2: More Than Two Sensors

As illustrated here, somewhat surprisingly, the computational cost increments associated with adding more sensors are often more than offset by the resultant more-effective survivor reduction. The use of additional sensors can often decrease the total cost of demodulating a particular environment at a specified target BER. This result is demonstrated in the following simulations with $M = 2$ to $M = 5$ sensors and 2 to 6 CCI.

The simulated sensor arrays had the following characteristics: Sensors were arranged in symmetric circular arrays of $M = 2$, 3, 4, and 5 elements, each with a diameter of 2 wavelengths. Both isotropic and cardioid sensor patterns were simulated. Cardioid sensors were arranged with lobes facing away from the center of the array. Each cardioid sensor was modeled as having a gain of +3 dB in the direction its lobe.

Each array configuration was tested in environments with the following properties: GSM beacon signals with nominal 960 MHz carriers were used for all environments. Environments contained 2 to 6 cochannel emitters, each having random carrier phase and carrier offset in the range of -40 Hz to +40 Hz (normal distribution). Emitters were placed in a plane, distributed uniformly, at distances of 30.5 meters (100 wavelengths) to 6.9 kilometers (6.25 symbol periods) from the center of the sensor array. Power control was simulated so that all signals arrived at the center of the array have received powers within 3 dB of a reference level. In each environment, multipath interference was simulated by placing one image of each emitter in the above specified planar region. Each emitter's multipath image had an effective radiated power of 0 dB to -10 dB relative to its source and a distance from the array of 1 to 4 times that of its source. Image distance and power were selected randomly from uniform distributions. In each environment, noise was added at a test-specified level relative to the power control reference level.

For these joint demodulation simulations, we used the constrained Viterbi-augmented ST algorithm with per-survivor decision feedback, with the degree of thinning adapted to provide a constant average BER of 10%, and with a pulse model length of $P = 4$ symbol periods and a channel memory of $K = 1$.

Both the TSNR and the number of sensors were varied, and the computational cost of achieving the target 10% BER was observed. For brevity, only the 6-CCI cardioid cases are presented here, in Figure 4, although other test cases were similar [1] and are discussed in Section 4. For each curve, the candidate counts are scaled up by the number of sensors to obtain the computational cost in equivalent single-sensor candidate counts. Each curve is defined over only a limited range of TSNR. Below this range, a 10% BER is not possible at any cost. Above this range, the demodulator produces a BER of less than 10%, even when operating at the minimum cost of one survivor per symbol period. Many of these curves have overlapping ranges, and in these overlapping ranges it is plain to see that in each case the receiver with a larger number of sensors has a lower cost. As examples, for omnidirectional antennas, going from 2 sensors to 3 sensors enables SNR to drop by 5 dB (from 10 dB to 5 dB) with no change in cost; going from 3 sensors to 4 sensors enables cost to be cut in half while reducing SNR required by about 1 dB. (Recall that the average SNR for each signal in these six-signal environments is about 7.8 dB below the

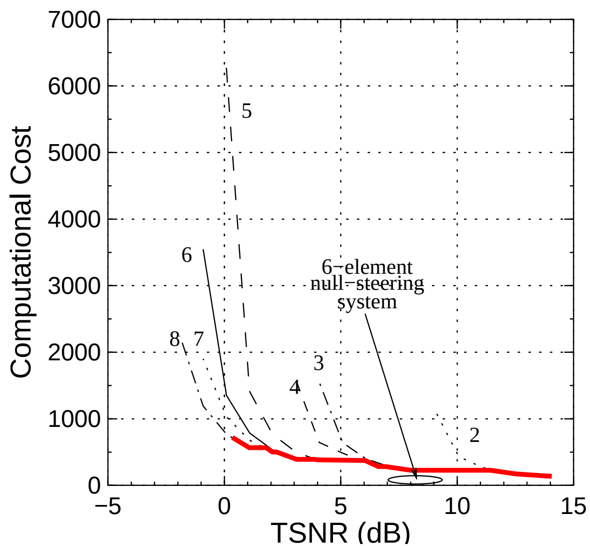

Figure 4: Computational Cost for 6 CCI Environment at 10% BER, Using Cardioid Sensors in a Circular Array. Different curves show cost-vs.-SNR for different numbers of sensors. Bold curve along bottom shows minimum-cost operation with a variable array size. Small oval below curve indicates approximate performance region for a null-steering receiver with a 6-element array.

TSNRs shown in Figure 4.)

If the demodulator is allowed to add or remove sensors as the environment changes, it can do so to minimize computational cost for a given BER target. In Figure 4, a heavy curve has been traced out along the base of the figure to show the approximate cost of a demodulator that is allowed to adjust the number of sensors it uses as TSNR changes. An oval near the bottom of the graph shows the approximate cost-TSNR operating region for a conventional null-steering receiver yielding a BER of 10% and using 6 antenna elements. This figure shows that joint demodulation may be preferable to null-steering when noise levels are high, or when the number of antenna elements is less than the number of signals. The null-steering receiver has a lower computational cost, but a much more limited range of operating conditions.

It is also useful to consider BER as a function of number of sensors, with computational cost and E_b/N_0 fixed. Data for the omnidirectional simulations are shown in Figure 5. Each curve shows BER vs. number of sensors for fixed computational cost and a fixed E_b/N_0 of 10 dB for the primary signal. Individual curves are not labeled, because this plot is intended to give a general sense of the results, not provide a

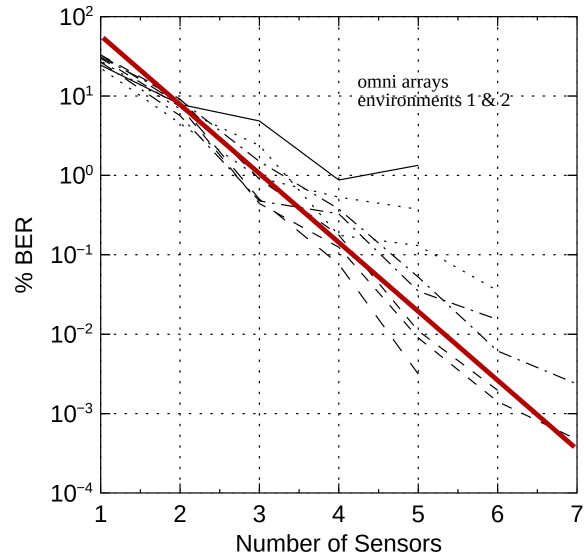

Figure 5: BER vs. Number of Sensors. With omnidirectional circular arrays, 2 to 6 CCI.

P	L	M	Cost	Savings
1	2	1	17	15
2	2	1	210	20
2	2	2	120	34
1	3	2	60	68
2	3	2	700	370

Table 3: Summary of Results from IS-136 Simulations. In each case, primary signal E_b/N_0 is 10 dB with equal-power signals. Target BER is 10%. Cost is in candidates per symbol period. Savings is cost relative to cost of conventional Viterbi algorithm with the same P, L, and M. Cost values are rounded to 2 significant digits.

comparison of specific test cases. A bold line has been added to the graph to show average slope of all of the linear-log plots. This slope corresponds to a factor-of-8 reduction in BER each time a sensor is added.

Experiment 3: Larger Alphabet.

The third series of simulations was similar to Experiment 1, but with IS-136 $\frac{\pi}{4}$-DQPSK symbols instead of binary GSM. Unlike GSM, an IS-136 signal lacks significant intersymbol interference (ISI). In the absence of CCI and severe multipath distortion, there is little advantage to using the Viterbi algorithm – a memoryless forced decision performs nearly as well. However, in a CCI environment, the symbol offsets between signals can still produce a *joint channel* with considerable ISI, so the Viterbi algorithm becomes necessary again. Table 3 summarizes the results of these simulations.

4 Conclusions

The results of the simulations performed support the following conclusions:

- Statistical thinning and the T-algorithm offer a better BER-cost tradeoff than the M-algorithm under the following conditions:
  - Smaller alphabet – Performance differences were greater for GSM than for IS-136.
  - Smaller signal spacing – Performance differences were usually greatest for equal-power environments. Both equal-power and 1 db-spaced (not shown) environments generally produced greater performance differences than 3 dB or 5 dB-spaced (not shown) environments.
  - Decision feedback – Performance differences were greater for the $K = 1$ (more feedback) cases than for the $K = 2$ (less feedback) cases. This can be seen in Table 2.
  - Dual-pol – Performance differences were greater for dual-pol reception than for single-sensor reception.

  The conclusions are born out in Figures 1, 2, and 3. In some cases, the limitation of the M-algorithm's trade-off parameter to integer values put it at a severe disadvantage to other algorithms, as is obvious in Figure 2.

- Dual-pol joint demodulation produces BER performance, for the GSM environments in this study, that is equivalent to an E_b/N_0 improvement of 8 to 12 dB relative to single-sensor.

- With survivor reduction, the total cost of multi-sensor joint demodulation can actually be lower than that of single-sensor processing (at the same BER).

- With survivor reduction, the use of per-survivor decision feedback roughly halves the computational cost of joint demodulation for GSM, regardless of the number of CCI. The BER loss is typically about 1%

- A BER of 10% for GSM should be attainable for up to 6-CCI with dual-pol reception (at 10 dB E_b/N_0) and current-generation computers.

- Non-isotropic antennas enable more efficient BER/cost tradeoff than do isotropic antennas, presumably due to increased channel diversity.

- Required SNR as well as computational cost to meet a specified BER target can both be decreased by increasing the number of antennas.

- Although exact channel estimates were used in all simulations reported here, other simulations performed show that accurate estimates of all channel parameters needed by the MIMO demodulator can be obtained in the presence of ten or more CCI by properly exploiting all the known bit sequences in GSM [2].

Acknowledgment

The research reported herein was supported in part by the National Reconnaissance office Directors Innovation Initiative, contract NRO-000-99-R-0176, and the National Science Foundation, Small Grants for Exploratory Research, NSF 0003284.

References

[1] W. A. Gardner, D. A. Burgess, C. W. Reed, "Computationally Efficient MIMO Demodulation," *Proc. 11th Virginia Tech. Symp. Wireless Personal Comms.*, June 6-8, 2001, Blacksburg, Va., pp. 153-164.

[2] "Radio Environment Analysis for GSM," SSPI Tech. Report, October 2000.

[3] G. D. Forney, "The Viterbi Algorithm," *Proc. IEEE*, vol. 61 pp. 268-278, Mar. 1973.

[4] T. Larsson, *A State-Space Partitioning Approach to Trellis Decoding*, PhD dissertation, Chalmers University, Goteborg, Sweden, 1991.

[5] R. Raheli, et. al., "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Comms.*, vol. 43, pp. 354-364, April 1995.