B. Appendix B. Word Error Rate for Asynchronous Radiotelegraph Signals

B.1. Introduction

An asynchronous radiotelegraph (RTTY) signal consists of a start bit interval followed by five or eight data bit intervals and a stop bit interval, as shown in Figure B1. The start bit and data bits are 1.0 bit intervals in length, while the stop bit is either 1.0 or 1.5 bit interval, depending on the particular code. Bit errors can affect not only the data bits, but the start and stop bits as well. This complicates the calculation of word error (character) rate (WER) when given the bit error rate (BER), since an error in the start or stop bits can affect not only the errored character, but following characters as well. In this appendix a derivation of WER is developed that takes this factor into account. The analysis is based on a Markov model which assumes bit errors are independent and identically distributed.

The RTTY decoder detects the beginning of a character when the received data signal transitions from mark to space, then starts a counter in order to decode the data bits. The stop bit follows the data bits and must be a mark for correct character decoding. If either the start bit is received as a mark or the stop bit is received as a space, a frame error (slip) occurs and the decoder attempts to resynchronize on the following character stream. In almost all cases such as this, a slip results in the loss of the errored character, as well as one or more following characters.

For the purposes of the following discussion, the RTTY signal is represented by a serial binary clocked data stream with from five to eight data bits and start and stop bits, all of one bit interval each. We will ignore the effects of clock frequency error or phase sampling error, so that the channel is completely specified as a binary symmetric channel with specified bit error probability. We assume that all n-bit code combinations are equally likely and that successive code combinations are independent. We use p to designate the bit error probability and $q = 1 - p$. Therefore, the error probability of an n-bit character is $1 - q^n$.

of all the n-bit error patters that can occur, half of them will have an error on the start bit. Of the remainder, half will have an error on the stop bit. An error on the start bit will cause a slip of at least one bit relative to a continuous data stream. An error on the stop bit may or may not cause a slip, depending on the recovery procedure. When a slip occurs, a number m of following characters can be lost in addition to the errored character. The expected value of m can be calculated using a Markov model of $n + 2$ states, where each state corresponds to the (mis)alignment in bits between the decoder and the transmitted signal. In the following, the general plan of development is to construct the state transition probability matrix for each of three decoder models, start the Markov chain with specified initial state probabilities, then calculate m as each model evolves to its final absorbing state.

B.2. RTTY Decoder Variants

In the design of electromechanical printers such as the once ubiquitous Teletype equipment, a mechanical escapement and distributor mechanism is used in which the escapement is triggered by

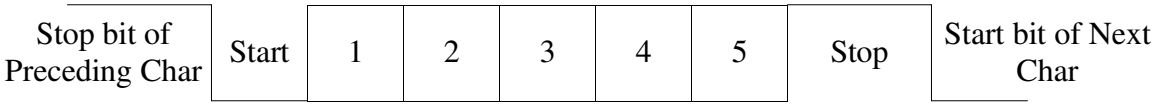

Figure B1. Asynchronous Character Framing

the start bit (space pulse). As the drive shaft revolves, the data bits are distributed to latchbars which function as a holding register. When the last data bit is found, the holding register contents are transmitted to the printing mechanism. The escapement is then reset during the stop bit (mark hold), after which it is ready for the next character. If the line signal at the stop bit is in error (space), then the drive shaft continues to rotate and the distributor overwrites previous information, usually resulting in a character garble. In this case, the machine "runs fast" or advances its phase relative to the original start bit by an amount depending on the exact rotational speed of the drive shaft.

In the design of electronic devices like the Universal Asynchronous Receiver Transmitter (UART), the receive clock is started at the mark-to-space transition which initiates the start bit and runs at some multiple (typically between eight and 16) of the baud rate. A bit clock divided down from the receive clock then samples the data bits at or near the center of each bit. The line signal at each sample is then shifted through a shift register. After all bits have been shifted into the register, the UART resumes searching for the next mark-to-space transition. Usually, but not necessarily, the UART samples the stop bit, as well as the data bits, in order to check for a frame error or break signal. Note that the next following character will not be initiated until the line signal has returned to mark following the last sample.

The UART behavior has the advantage (or disadvantage, depending on the intent of the design) that only a single character will be decoded at the beginning of a long space interval that might be generated by a frame error, break signal or space disconnect. Where it is necessary to time these intervals, the design can be modified so that, if the line signal is space during the stop bit, the sample clock continues to run. In this case, a space found at the stop bit sample is taken as the start bit of the next character and the samples continue to shift into the shift register. The behavior of this modified design is thus similar to that of the electromechanical printers described previously.

In the preceding discussion, three cases have been identified:

Case 1 (Electromechanical-1). Operation of the shift register ceases at the last data bit. The decoder either resumes searching for the next start bit, if the line signal is mark, or begins the next character if the line signal is space.

Case 2 (Electromechanical-2). Operation of the shift register continues past the last data bit to include the stop bit, in order to detect a frame error, break signal or space disconnect. The decoder either resumes searching for the next start bit, if the line signal is mark, or begins the next character if the line signal is space.

Case 3 (UART). Operation of the shift register continues past the last data bit to include the stop bit, in order to detect a frame error, break signal or space disconnect. The decoder waits until the line signal returns to mark (if not already in that condition) and resumes searching for the next start bit.

In Case 1, the decoder will begin the next character one bit early if an error occurs on the stop bit. In Case 2, the decoder will begin the next character correctly (assuming the next start bit is correct) whether or not an error occurs on the stop bit. In Case 3, if an error occurs on the stop bit, the decoder will freeze until the line signal returns to mark and then resume searching for a start bit. In the following development, each of these three cases is studied in detail in order to evaluate how well each succeeds in minimizing m with respect to p.

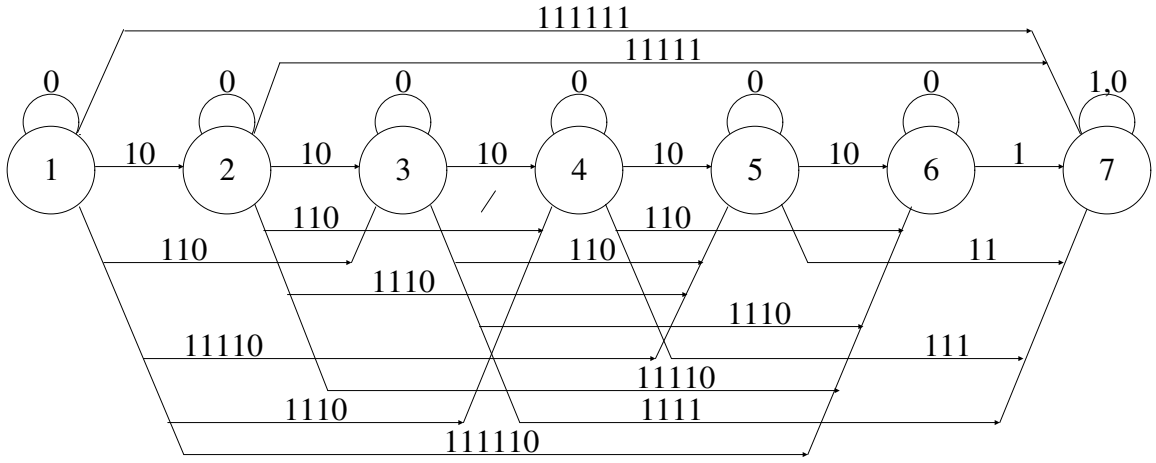

Figure B2. Markov Model for Case 2

B.3. Markov Models

In transmitting a continuous stream of RTTY characters, the stop bit of a character is followed immediately by the start bit of the following character, as shown in Figure B1. In the Markov models considered below, each state corresponds to a misalignment between the decoder and the transmitted signal. Starting in a specific initial state, succeeding state transitions are driven by the bits (mark or space) as they arrive, eventually concluding in the final state where the decoder and transmitted signal are once again correctly aligned. For each evolution of the model, the decoder slips a number of bits to the first space bit, assumed to be the start bit of the next character, and then slips five bits (Case 1) or six bits (Cases 2 and 3) to the end of that character. If not correctly aligned at that point, the model evolves an additional step. The model predicts the number of character slips m as a function of bit error probability p. The number of bit slips that may precede a character slip are ignored, since the WER contribution produced by the bit slips is small compared to the character slips.

The probabilities assigned the state transitions are computed separately for each of the three cases mentioned above. For instance, in Case 2 with $n = 5$, the transition diagram is shown in Figure B2. The transition probabilities follow directly from the assumptions of equiprobable and independent code combinations. If the decoder is in State 7 and the next bit is a space, then a start bit has been detected and the next five bits plus the stop bit will be processed as a valid character. Now consider the Markov chain with initial probability concentrated in State 3, which represents a misalignment of three bits late or, equivalently, four bits early. If the next bit is a space, the chain returns to the same state. If the next bit is a mark, the chain slips to the next higher numbered state. The Markov models considered here are absorbing, with State 7 the absorbing state. Therefore, the chain eventually slips to State 7, after which normal character decoding resumes.

The three models are approximate in that they are finite and in some cases violate the memoryless property. However, the intent of the models is to determine general behavior, not predict precise behavior in every case; moreover, the exceptional cases occur with very low probability. In models it is assumed that additional bit errors which may occur while the decoder is attempting to resynchronize to the character stream do not alter the behavior predicted by the model; in particular, they do not affect the start or stop bits. At bit error rates likely to occur in practice, this is a reasonable

$$\begin{bmatrix} 1/4 & 1/8 & 1/16 & 1/32 & 1/64 & 1/128 & 65/128 \\ 0 & 1/4 & 1/8 & 1/16 & 1/32 & 1/64 & 1/64 \\ 0 & 1/2 & 1/4 & 1/8 & 1/16 & 1/32 & 1/32 \\ 0 & 0 & 1/2 & 1/4 & 1/8 & 1/16 & 1/16 \\ 0 & 0 & 0 & 1/2 & 1/4 & 1/8 & 1/8 \\ 0 & 0 & 0 & 0 & 1/2 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1/2 & 1/4 & 1/8 & 1/16 & 1/32 & 1/64 & 1/64 \\ 0 & 1/2 & 1/4 & 1/8 & 1/16 & 1/32 & 1/32 \\ 0 & 0 & 1/2 & 1/4 & 1/8 & 1/16 & 1/16 \\ 0 & 0 & 0 & 1/2 & 1/4 & 1/8 & 1/8 \\ 0 & 0 & 0 & 0 & 1/2 & 1/4 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1/4 & 1/4 & 3/16 & 1/8 & 5/64 & 1/16 & 3/16 \\ 0 & 1/4 & 1/4 & 3/16 & 1/8 & 5/64 & 7/64 \\ 0 & 0 & 1/4 & 1/4 & 3/16 & 1/8 & 3/16 \\ 0 & 0 & 0 & 1/4 & 1/4 & 3/16 & 5/16 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/4 & 3/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Case 1 Case 2 Case 3

Figure B3. Transition Probability Matrices

assumption. When they do in fact occur, the chances are good that they will affect the data bits, but not the start or stop bits. Obviously, the Markov model could be extended to include the affects of these errors, but this is left for a subsequent exercise.

Let $v_1, v_2, ..., v_{n+2}$ represent the initial probabilities assigned to the states $s_1, s_2, ..., s_{n+2}$ respectively, and let $\pi^{(0)} = (v_1, v_2, ..., v_{n+2})$ represent the initial state vector of these probabilities. Let the matrix $P = \{p_{ij}\}$ represent the transition probabilities, where p_{ij} is the probability of a transition from state s_i to s_j. Given the state vector $\pi^{(k-1)}$ at transition $k - 1$ ($k > 0$), the state vector at transition k is

$$\pi^{(k)} = \pi^{(k-1)}P = \pi^{(0)}P^k.$$

Since this is an absorbing Markov chain, the state vector should approach

$$\pi = \lim_{k\to\infty} \pi^{(k)} = (0, 0, ..., 1) ,$$

for all $\pi^{(0)}$, where the absorbing state s_{n+2} has been ordered last.

The value of the s_{n+2} component of the state vector at each transition is the probability that the slip interval ends at or before that transition. Let $s^{(k)}$ be the value of this component at the kth transition. The mean slip interval at the kth ($k > 0$) transition is

$$m^{(k)} = \sum_{i=1}^{k} i\, s^{(i)} - (i - 1)\, s^{(i-1)} ,$$

and

$$m = \lim_{k\to\infty} m^{(k)} .$$

There is no a-priori guarantee that this limit exists for all possible sequences of character combinations; in fact, a continuous sequence of spaces following a mark would hang the Case 3 decoder forever. In practice, however, the effect of such improbable sequences can be ignored. To test this conjecture, the computations in each of the cases were performed by computer and converged to within seven decimal digits for k less than 100.

Case	Initial	Start Bit	Stop Bit I	Stop Bit II
1	6.88	4.63	7.75	
2	4.86	8.00		
3	3.16	4.78	2.33	3.75

Table B1. Word Error Probabilities

B.4. Discussion of Results

It is a straightforward, but tedious, exercise to develop Markov models for Case 1 and Case 3, which are similar to the Case 2 model of Figure B2. The state transition matrices for all three cases are shown in Figure B3. In all of these models, the signal structure of Figure B1 is used with the stop bit interval equal to one bit. As mentioned at the outset, the character error probability is $1 - q^7$, since an error on any bit, including the start and stop bits, counts as a character error. However, errors on the start and stop bits have an additional impact due to the slips that result. The purpose of this section is to predict the extent of this impact using the Markovian models developed in the last section.

A new arrival to a channel in the process of sending continuous traffic may experience a number of character slips until correctly synchronized on the transmitted start and stop bits. The mean number of character slips can be determined by setting $\pi^{(0)} = (1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7)$; that is, making all initial states equiprobable. The value of m determined by the above procedure for each of the three models is shown in the second column of Table B1.

Of all error patters, those with good start and stop bits represent errors in the data bits, which do not cause a character slip. Since one-quarter of all error patterns have good start and stop bits, the WER contribution due to data errors is

$$\frac{(1 - q^7)}{4}.$$

In the case of a bad start bit (space becomes mark), the decoder will slip to the first data bit. The number m of character slips can be determined by setting $\pi^{(0)} = (0, 1, 0, 0, 0, 0, 0)$; that is, starting the chain with an initial slip of one bit. Since half of the error patters have a bad start bit, the WER contribution due to start-bit character slips is

$$m \frac{(1 - q^7)}{2},$$

where m is determined by the above procedure for all three models, as shown in the fourth column of Table B1.

In the case of a good start bit and a bad stop bit (mark becomes space), the decoder will either slip one bit early (Case 1), slip to the first mark-space transition (Case 3) or have no slip at all (Case 2). This requires that each case be considered separately.

In Case 1, m can be determined by setting $\pi^{(0)} = (0, 0, 0, 0, 0, 1, 0)$; that is, starting the chain with an initial slip of six bits, which is equivalent to a negative slip of one bit. The WER contribution due to stop-bit character slips is

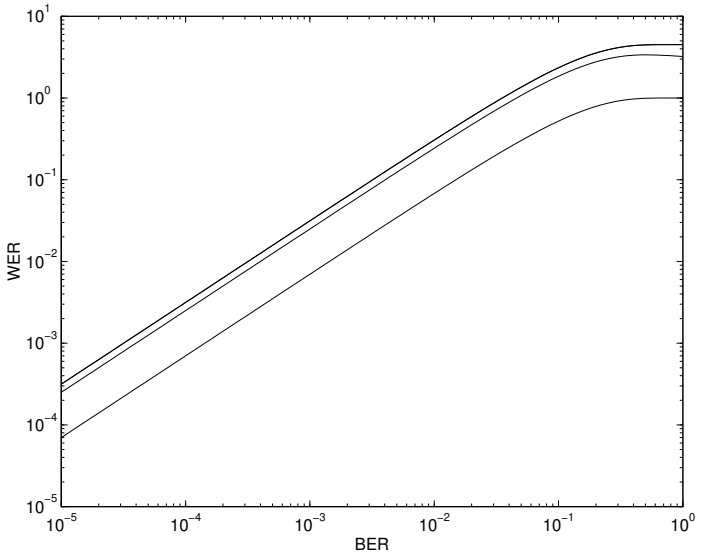

Figure B4. Word Error Rate vs. Bit Error Rate

$$m\,\frac{(1-q^7)}{4}\,.$$

where *m* is determined by the above procedure for all three models, as shown in the fifth column of Table B1.

In Case 2, a bad stop bit is treated the same as a bad data bit and does not cause a slip. This accounts for one-fourth of the data errors and is added to that WER contribution. In Case 3, there are two subcases: one in which the following character has a good start bit, and the other in which it is bad. In the first subcase, which occurs with probability *q*, the decoder slips until the first mark is found, then resets and continues to search for the next start bit. Thus, we set $\pi^{(0)} = (0, ½, ¼, ⅛, 1⁄16, 1⁄32, 1⁄32)$ to approximate the initial state occupancy in this subcase. The results are shown in the fifth column of Table B1. In the second subcase, which occurs with probability *p*, the decoder resets on the bad start bit (mark) and continues as in Case 1.

In summary, the total WER for Case 1 due to all causes is

$$\mathrm{WER}_1 = (1-q^7)(\frac{1}{4}+\frac{4.63}{2}+\frac{7.75}{4}) = 4.50(1-q^7)\,,$$

the total WER for Case 2 due to all causes is

$$\mathrm{WER}_2 = (1-q^7)(\frac{1}{4}+\frac{8.00}{2}+\frac{1}{4}) = 4.50(1-q^7)\,,$$

and the total WER for Case 3 due to all causes is

$$\mathrm{WER}_3 = (1-q^7)(\frac{1}{4}+\frac{4.78}{2}+\frac{2.33p+3.75q}{4}) \approx 3.58(1-q^7)\,,$$

assuming $q >> p$. Figure B4 shows a graph of the WER as a function of the BER for all three cases.The top trace represents both Case 1 and Case 2, which are almost identical. The middle trace represents Case 3. The bottom trace represents the WER of the five-bit character when the RTTY

decoder is operating in synchronous mode and the PLL has correctly locked on the start-bit transitions. In this case, the start and stop bits are ignored, other than to derive the Viterbi distance.