[1] The Viterbi Algorithm.

M. S. Ryan* and G. R. Nudd.
Department of Computer Science, University of Warwick, Coventry, CV4 7AL, England.

*e-mail address - (msr@dcs.warwick.ac.uk)

12th February 1993.

Abstract.

This paper is a tutorial introduction to the Viterbi Algorithm, this is reinforced by an example use of the Viterbi Algorithm in the area of error correction in communications channels. Some extensions to the basic algorithm are also discussed briefly. Some of the many application areas where the Viterbi Algorithm has been used are considered, including it's use in communications, target tracking and pattern recognition problems. A proposal for further research into the use of the Viterbi Algorithm in Signature Verification is then presented, and is the area of present research at the moment.

1. Introduction.

The Viterbi Algorithm (VA) was first proposed as a solution to the decoding of convolutional codes by Andrew J. Viterbi in 1967, **[1]**, with the idea being further developed by the same author in **[2]**. It was quickly shown by Omura **[3]** that the VA could be interpreted as a dynamic programming algorithm. Both Omura and Forney **[3,4]** showed that the VA is a maximum likelihood decoder. The VA is often looked upon as minimizing the error probability by comparing the likelihoods of a set of possible state transitions that can occur, and deciding which of these has the highest probability of occurrence. A similar algorithm, known as the Stack Sequential Decoding Algorithm (SSDA), was described by Forney in **[5]** as an alternative to the VA, requiring less hardware to implement than the VA. The SSDA has been proposed as an alternative to the VA in such applications as target tracking **[6]**, and high rate convolutional decoding **[5]**. It can be shown though, that this algorithm is sub-optimum to the VA in that it discards some of the paths that are kept by the VA.

Since it's conception the VA has found a wide area of applications, where it has been found to be an optimum method usually out performing previous methods. The uses it has been applied to not just covers communications for which it was originally developed, but includes diverse areas such as handwritten word recognition, through to nonlinear dynamic system state estimation.

This report is in effect a review of the VA. It describes the VA and how it works, with an appropriate example of decoding corrupted convolutional codes. Extensions to the basic algorithm are also described. In section 3 some of the applications that the VA can be put to are described, including some uses in communications, recognition problems and target tracking. The area of dynamic signature verification is identified as an area requiring further research.

---

1. *Warwick Research Report RR238*

2. The Viterbi Algorithm.

In this section the Viterbi Algorithm (VA) is defined, and with the help of an example, its use is examined. Some extensions to the basic algorithm, are also looked at.

2.1 The Algorithm.

The VA can be simply described as an algorithm which finds the most likely path through a trellis, i.e. shortest path, given a set of observations. The trellis in this case represents a graph of a finite set of states from a Finite States Machine (FSM). Each node in this graph represents a state and each edge a possible transitions between two states at consecutive discrete time intervals. An example of a trellis is shown below in Figure 1a and the FSM that produced this trellis is shown in Figure 1b. The FSM referred to here is commonly used in digital electronics and is often refered to in the literature as a Markov Model (MM), e.g. **[7,8,9]**.

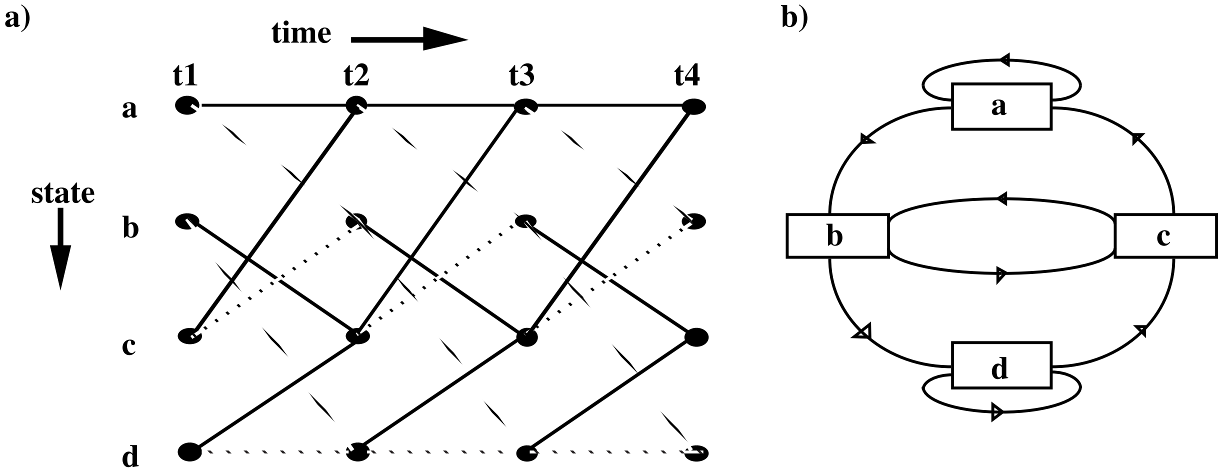


Figure 1. Showing **a)** trellis diagram spread over time and **b)** the corresponding state diagram of the FSM.

For each of the possible transitions within a given FSM there is a corresponding output symbol produced by the FSM. This data symbol does not have to be a binary digit it could instead represent a letter of the alphabet. The outputs of the FSM are viewed by the VA as a set of observation symbols with some of the original data symbols corrupted by some form of noise. This noise is usually inherent to the observation channel that the data symbols from the FSM have been transmitted along.

The trellis that the VA uses corresponds to the FSM exactly, i.e. the structure of the FSM is available, as is the case in it's use for convolutional code decoding. Another type of FSM is the Hidden Markov Model (HMM) **[8,9]**. As the name suggests the actual FSM is hidden from the VA and has to be viewed through the observations produced by the HMM. In this case the trellis's states and transitions are estimates of the underlying HMM. This type of model is useful in such applications as target tracking and character recognition, where only estimates of the true state of the system can be produced. In either type of model, MM or HMM, the VA uses a set of metrics associated with the observation symbols and the transitions within the FSM. These metrics are used to cost the various paths through the trellis, and are used by the VA to decide which path is the most likely path to have been followed, given the set of observation symbols.

Before defining the VA the following set of symbols have to be defined :-

- t - The discrete time index.
- N - Total number of states in the FSM.
- x_n - The nth state of the FSM.
- o_t - The observation symbol at time t, which can be one of M different symbols.
- sp_{nt} - The survivor path which terminates at time t, in the nth state of the FSM. It consists of an ordered list of x_n's visited by this path from time t = 0 to time t.
- T - Truncation length of the VA, i.e. the time when a decision has to be made by the VA as to which sp_{nt} is the most likely.
- π_n - Initial state metric for the nth state at t = 0. Defined as the probability that the nth state is the most likely starting start, i.e. Prob(x_n at t = 0).
- a_{nm} - The transition metric for the transition from state x_m at time t - 1 to the state x_n at time t. Defined as the probability that given that state x_m occurs at time t - 1, the state x_n will occur at time t, i.e. Prob(x_n at t | x_m at t - 1).
- b_n - The observation metric at time t, for state x_n. Defined as the probability that the observation symbol o_t would occur at time t, given that we are in the state x_n at time t, i.e. Prob(o_t | x_n at t).
- Γ_{nt} - The survivor path metric of sp_{nt}. This is defined as the Product of the metrics (π_n,a_{nm} and b_n) for each transition in the nth survivor path, from time t = 0 to time t.

The equations for the model metrics, π_n,a_{nm} and b_n, can be derived mathematically where their properties result from a known application. If the metric properties are not known, re-estimation algorithms can be used, such as the Baum-Welch re-estimation algorithm **[8,9]**, to obtain optimum probabilities for the model. It is also usual to take the natural logarithm of the metrics, so that arithmetic underflow is prevented in the VA during calculations.

The VA can now be defined :-

Initialization.

```
    t = 0;
    For all n, where 1   n   N
       Γn0 = ln πn;
       spn0 = [xn];
    End For;
```

Calculation.

```
    For all t, where 1   t   T,
       For all n, where 1   n   N
          For all m, where 1   m   N
             Γnt = Max [Γmt-1 + ln anm + ln bn];
```

```
            End For;
            sp_nt = Append[x_n,sp_mt] such that Γ_mt-1 + ln a_nm + ln b_n = Γ_nt;
        End For;
    End For;
```

Decision.

```
    If t = T,
        For all n, where 1   n   N
            Γ_T = Max [Γ_nt];
        End For;
        sp_T = sp_nt such that Γ_nt = Γ_T;
```

In English the VA looks at each state at time t, and for all the transitions that lead into that state, it decides which of them was the most likely to occur, i.e. the transition with the greatest metric. If two or more transitions are found to be maximum, i.e. their metrics are the same, then one of the transitions is chosen randomly as the most likely transition. This greatest metric is then assigned to the state's survivor path metric, Γ_{nt}. The VA then discards the other transitions into that state, and appends this state to the survivor path of the state at t - 1, from where the transition originated. This then becomes the survivor path of the state being examined at time t. The same operation is carried out on all the states at time t, at which point the VA moves onto the states at t + 1 and carries out the same operations on the states there. When we reach time t = T (the truncation length), the VA determines the survivor paths as before and it also has to make a decision on which of these survivor paths is the most likely one. This is carried out by determining the survivor with the greatest metric, again if more than one survivor is the greatest, then the most likely path followed is chosen randomly. The VA then outputs this survivor path, sp_T, along with it's survivor metric, Γ_T.

2.2 Example.

Now that the VA has been defined, the way in which it works can be looked at using an example communications application. The example chosen is that of the VA's use in convolutional code decoding, from a memoryless Binary Symetric Channel (BSC), as described in **[2]**. A picture of the communications system that this example assumes, is shown below in Figure 2. This consists of encoding the input sequence, transmitting the sequence over a transmission line (with possible noise) and optimal decoding the sequence by the use of the VA.

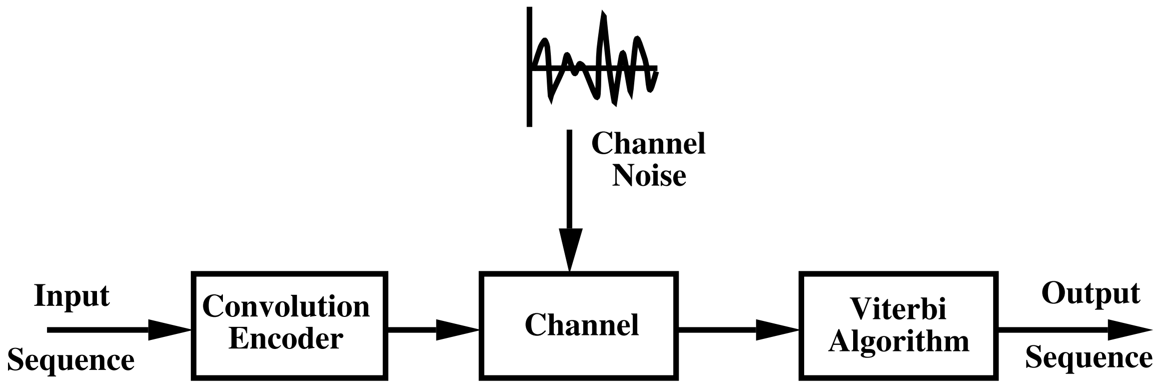


Figure 2. Communications channel set up for the example.

The input sequence, we shall call it I, is a sequence of binary digits which have to be transmitted along the communications channel. The convolutional encoder consists of a shift register, which shifts in a number of the bits from I at a time, and then produces a set of output bits based on logical operations carried out on parts of I in the register memory. This process is often referred to as convolutional encoding. The encoder introduces redundancy into the output code, producing more output bits than input bits shifted into it's memory. As a bit is shifted along the register it becomes part of other output symbols sent. Thus the present output bit that is observed by the VA has information about previous bits in I, so that if one of these symbols becomes corrupted then the VA can still decode the original bits in I by using information from the previous and subsequent observation symbols. A diagram of the convoltional encoder used in this example is shown in Figure 3. It is assumed here that the shift register only shifts in one bit at a time and outputs two bits, though other combinations of input to output bits are possible.

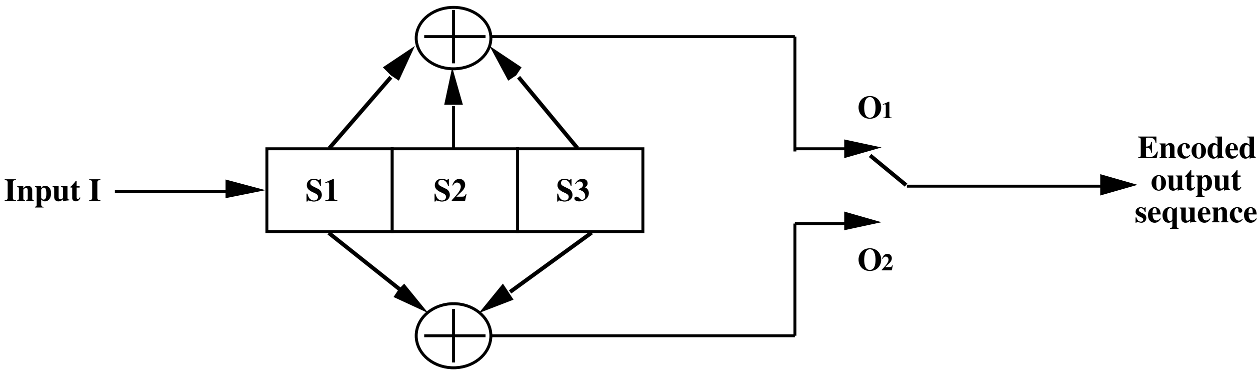


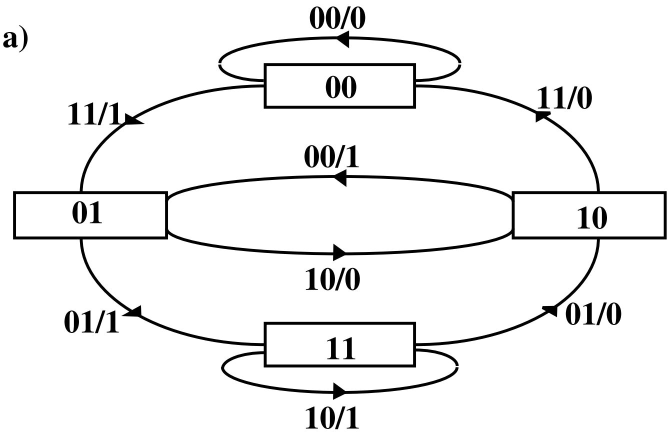
Figure 3. Example convolutional encoder.

This encoder can be represented by the FSM shown in Figure 4a. The boxes in this diagram represent the shift register and the contents are the state that the FSM is in. This state corresponds to the actual contents of the shift register locations S2 followed by S1, i.e. if we are in state 01, then the digit in S1 is 1 and the digit in S2 is 0. The lines with arrows, represent the possible transitions between the states. These transitions are labeled as x/y, where x is a two digit binary number, which represents the the output symbol sent to the communications channel for that particular transition and y represents the binary digit from I, that when shifted into the encoder causes that particular transition to occur in the state machine.

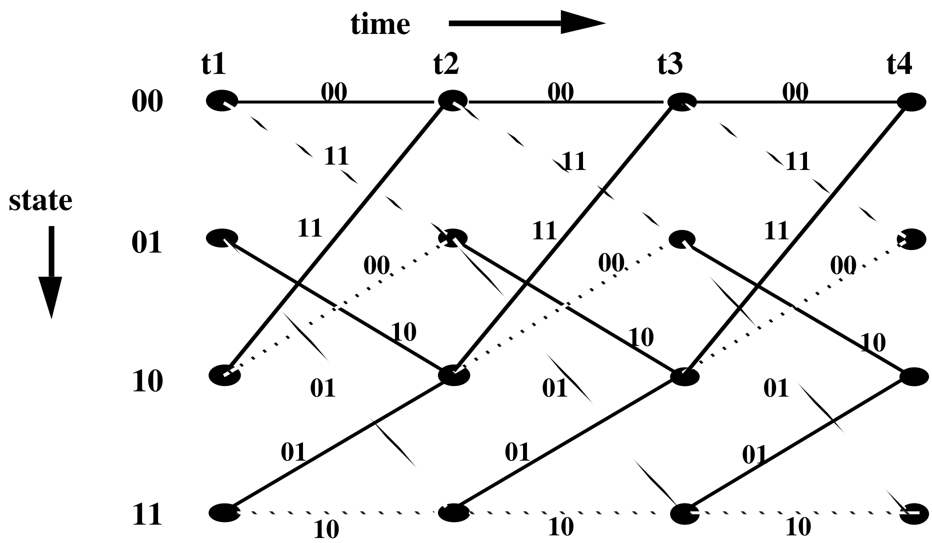
The encoded sequence produced at the output of the encoder is transmitted along the channel where noise inherent to the channel can corrupt some of the bits so that what was transmitted as a 0 could be interpreted by the receiver as a 1, and vice versa. These observed noisy symbols are then used along with a trellis diagram of the known FSM to reconstruct the original data sequence sent. In our example the trellis diagram used by the VA is shown in Figure 4b. This shows the states as the nodes which are fixed as time progresses. The possible transitions are shown as grey lines, if they were caused by a 1 entering the encoder, and the black lines, if they were caused by a 0 entering the encoder. The corresponding outputs that should of been produced by the encoder are shown, by the two bit binary digits next to the transition that caused them. As can be seen in Figure 4b the possible transitions and states remain fixed between differing time intervals. The trellis diagram of Figure 4b can be simplified to show the recursive nature of the trellis, as is shown in Figure 4c.

It was shown by Viterbi in **[2]**, that the log likelihood function used to determine survivor metrics can be reduced to a minimum distance measure, known as the Hamming Distance. The Hamming distance can be defined as the number of bits that are different between, between the symbol that the VA observes, and the symbol that the convolutional encoder should have produced if it followed a particular input sequence. This measure defines the combined measure of a_{nm} and b_n for each transition in the trellis. The π_n's are usually set before decoding begins such that the normal start state of the encoder has a $\pi_n = 0$ and the other states in the trellis have a π_n whose value is as large as possible, preferably ∞. In this example the start state of the encoder is always assumed to be state 00, so $\pi_0 = 0$, and the other π_n's are set to 100.

a)



b)



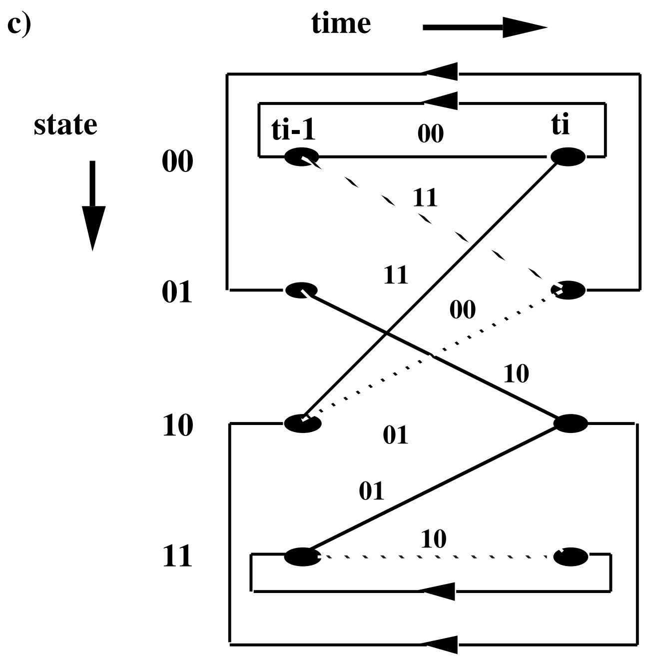


Figure 4. Showing the various aspects of the FSM in the example. a) shows the FSM diagram, b) the trellis diagram for this FSM spread over time and c) shows the recursive structure of this trellis diagram.

Input	States			Outputs	
I	S1	S2	S3	O1	O2
0	0	0	0	0	0
1	1	0	0	1	1
1	1	1	0	0	1
0	0	1	1	0	1
0	0	0	1	1	1
0	0	0	0	0	0

Table 1.

As an example if it is assumed that an input sequence I, of 0 1 1 0 0 0 is to be transmitted across the BSC, using the convolutional encoder described above, then the output obtained from the encoder will be 00 11 01 01 11 00, as shown in Table 1. The output is termed as the Encoder Output Sequence (EOS). Table 1 also shows the corresponding contents of each memory element of the shift register, where each element is assumed to be initialized to zero's at the start of encoding. As the EOS is constructed by the encoder, the part of the EOS already formed is transmitted across the channel. At the receiving end of the channel the following noisy sequence of bits may be received, 01 11 01 00 11 00. As can be seen there are two bit errors in this sequence, the 00 at the beginning has changed to 01, and similarly the fourth symbol has changed to 00 from 01. It is the job of the Viterbi Algorithm to find the most likely set of states visited by the original FSM and thus determine the original input sequence.

For simplicity the four states of the encoder are assigned a letter such that a = 00, b = 01, c = 10 and d = 11. At the start of the decoding process, at t = 1, the π_n's are first of all assigned to each of the corresponding Γ_{n0}, so $\Gamma_{a0} = 0$ and $\Gamma_{b0} = \Gamma_{c0} = \Gamma_{d0} = 100$. The hamming distance for each transition is then worked out, e.g. at t = 1 the symbol observed by the VA was 01, so the hamming distance for the transition from a at time 0 to a at time 1 is 1. Once this is done the VA then finds the total metric for each transition, e.g. for state a at t =1 there are two possible transitions into it, one from c with a total metric of 101 and one from a with a total metric of 1. This is shown in Figure 5a for all the states at time t =1, showing the π_n's for each state after the state letter. The VA then selects the best transition into each state, in the case of a this would be the transition from a at t = 0, since it's metric is the minimum of the two transitions converging into a at t = 1. Then the VA sets the sp_{n1} and Γ_{n1} for each state, so $sp_{a1} = \{a,a\}$ and $\Gamma_{a1} = 1$. The survivor paths and the corresponding survivor path lengths are shown in Figure 5b at t = 1. Then the VA repeats the above process for each time step and the decoding and survivor stages are shown in Figure 5c and 5d for t = 2, and the same stages for t = 6.

When the VA reaches time T = t, then it has to decide which of the survivor paths is the most likely one, i.e. the path with the smallest Hamming Distance. In this example T is assumed to be 6, so the path terminating at state a in Figure 5f has the minimum Hamming distance, 2, making this the most likely path taken. Next, the VA would start outputing the estimated sequence of binary digits that it thinks were part of the original input sequence. It is found that the estimated sequence is 0 1 1 0 0 0 which corresponds directly to the original input sequence. Thus the input sequence has been recovered dispite the error introduced during transmission.

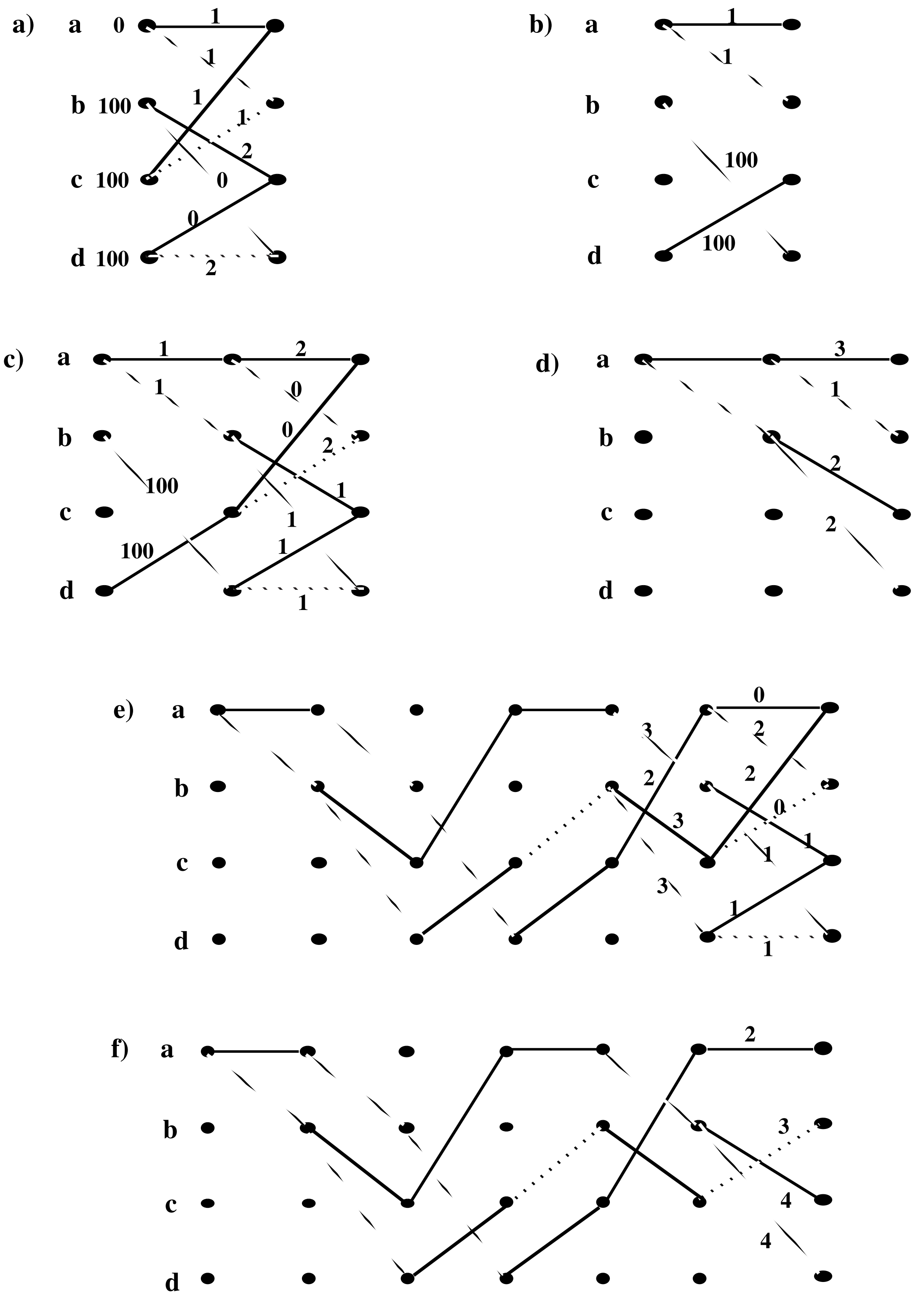


Figure 5. Some of the stages of the decoding process
a) Transitions and Metrics and b) Survivor Paths and Survivor Path Metrics at t = 1,
c) Transitions and Metrics and d) Survivor Paths and Survivor Path Metrics at t = 2,
e) Transitions and Metrics and f) Survivor Paths and Survivor Path Metrics at t = 6.

2.2 Algorithm Extensions.

Now that the VA has been examined in some detail, various aspects of the algorithm are now looked at so that a viable research area can be established. In this section, some of the possible extensions to the algorithm are looked at though these are limited in scope.

In the example above the VA relies on inputs from a demodulator which makes hard decisions on the symbols it received from the channel. That is, whatever type of modulation used, be it phase, frequency or amplitude modulation, then the demodulator has to make a firm decision whether a 0 or 1 was transmitted. One obvious extension of the VA is that of replacing this firm decision with a soft-decision **[10]**. In this method the demodulator produces soft outputs, i.e. each symbol produced by the demodulator instead of consisting of a 0 or 1 consists of the symbol that the demodulator thinks was sent along with other bits which represent the confidence that the symbol was transmitted. This increases the information presented to the VA increasing it's performance. The VA can then be used to decode these soft decisions and output a hard decision as before.

The next step up from this is a Viterbi algorithm which produces soft output decisions of it's own **[11]** - this is known as a Soft Output Viterbi Algorithm (SOVA). In this version of the VA a hard decision is given on the most likely survivor path, but information about how confident that each symbol in the path occured is also produced.

Another extension to the algorithm was suggested recently, by Bouloutas *et al* **[12]**. Bouloutas's extension generalizes the VA so that it can correct insertions and deletions in the set of observations it receives, as well as symbol changes. This method combines the known FSM, that produced the symbols in the first place, such as in the example above, with an FSM of the observation sequence. A trellis diagram is produced, known as a product trellis diagram, which compensates for insertions and deletions in the observation sequence. For each of the insertion, deletion and change operations a metric is assigned, which also depends upon the application the VA is being applied to. The VA produces the most likely path through the states of the FSM, estimates of the original data sequence, as well as the best sequence of operations performed on the data to obtain the incorrect observation sequence. An application of this VA, is in the use of correcting programming code whilst compiling programs, since many of the errors produced while writing code tend to be characters missed out, or inserted characters. Another extension to the VA is that of a parallel version. Possible solutions to this have been suggested by Fettweis and Meyr **[13]**, and Lin *et al* **[14]**. The parallel versions of the VA have risen out of the needs for fast hardware implementations of the VA. Though the area of parallel VA's will not be considered in this report.

3. Applications.

This section looks at the applications that the VA has been applied to. The application of the VA in communications is initially examined, since this is what the algorithm was initially developed for. The use of the VA in target tracking is also looked at and finally recognition problems. In all the applications, that the VA has been applied to since it's conception, the VA has been used to determine the most likely path through a trellis, as discussed in section 2. What makes the VA an interesting research area is that the metrics, (π_n, a_{nm} and b_n), used to determine the most likely sequence of states, are application specific. It should be noted that the VA is not limited to the areas of application mentioned in this section though only one other use is known to this author. This is the application of the VA in digital magnetic recording systems **[15,16]**. Though this section sums up the uses of the VA there are probably a number of other application areas that could be identified.

3.1 Communications.

A number of the uses of the VA in communications have already been covered in section 2. These include the first proposed use of the VA in decoding convolutional codes, see example in section 2 **[1,2]**. It was shown that the VA could be used to combat Intersymbol Interference (ISI), by Forney in **[17]**. ISI usually occurs in modulation systems where consecutive signals disperse and run into each other causing the filter at the demodulator to either miss a signal, or wrongly detect a signal. ISI can also be introduced by imperfections in the filter used. Forney suggested that the VA could be used to estimate the most likely sequence of 0's and 1's, that entered the modulator at the transmitter end of the channel, given the sequence of ISI affected observation symbols. So if a convolutional code was used for error control then the output from this VA would be passed through another VA to obtain the original input sequence into the transmitter. The VA used in this situation is more commonly referred to as a Viterbi Equalizer and has been used in conjunction with Trellis-Coded Modulation (TCM) **[11,18]** and also on different channel types **[19]**. The advantage of using a VA in the equalizer is that the SOVA, described in section 2, can be used to pass on more information into the decoder.

Another application of the VA in communications is that of decoding TCM codes. This method of encoding was presented by Ungerboeck, **[20]** for applications where the redundancy introduced by convolutional encoding could not be tolerated because of the reduction in data transmission rate or limited bandwidth. This method combines the error correcting abilities of a trellis code with the redundancy which can be introduced into the modulation signal itself via multiple amplitude levels or multiple phases. Instead of transmitting the extra redundant bits produced by the convolutional encoder for error control, these bits are mapped onto different amplitudes or phases in the modulation set which ensures that the bandwidth of the signal does not have to be increased. The VA is then used to decode the TCM codes, and produce the most likely set of bits as in convolutional decoding. It should also be noted that the decisions coming from the demodulator are in fact soft decisions, and a soft-decision VA has to be used in the decoding **[20]**.

Much of the research carried out in the use of the VA in communications has been directed into finding better performance TCM codes **[21]** and in the application of the VA as an equalizer in different communication environments **[19]**. It was therefore decided to look for another application that the VA could be applied too.

3.2 Target Tracking.

The use of the VA in the field of Target Tracking is investigated in this section. Work in this application area has been carried out to date using Kalman Filters for the tracking of targets. The basic aim is to take observations from a radar, sonar or some other form of detector and to estimate the actual position of the target in relation to the detector. In an ideal world this would be a simple task since the detector should give us the true position of the target, but in reality various problems arise which affect the readings from the detector. These usually from noise, be it background, signal deteoriation or due to imperfections in the detector. There is also the matter of manoeuvrering by the target which typically results in the modelling of a non-linear system. Another type of noise that can be introduced into the signals used by the detector is random interference. This is usually generated by natural causes such as weather conditions or by false alarms from the detector and can be introduced artificially - often referred to as jamming.

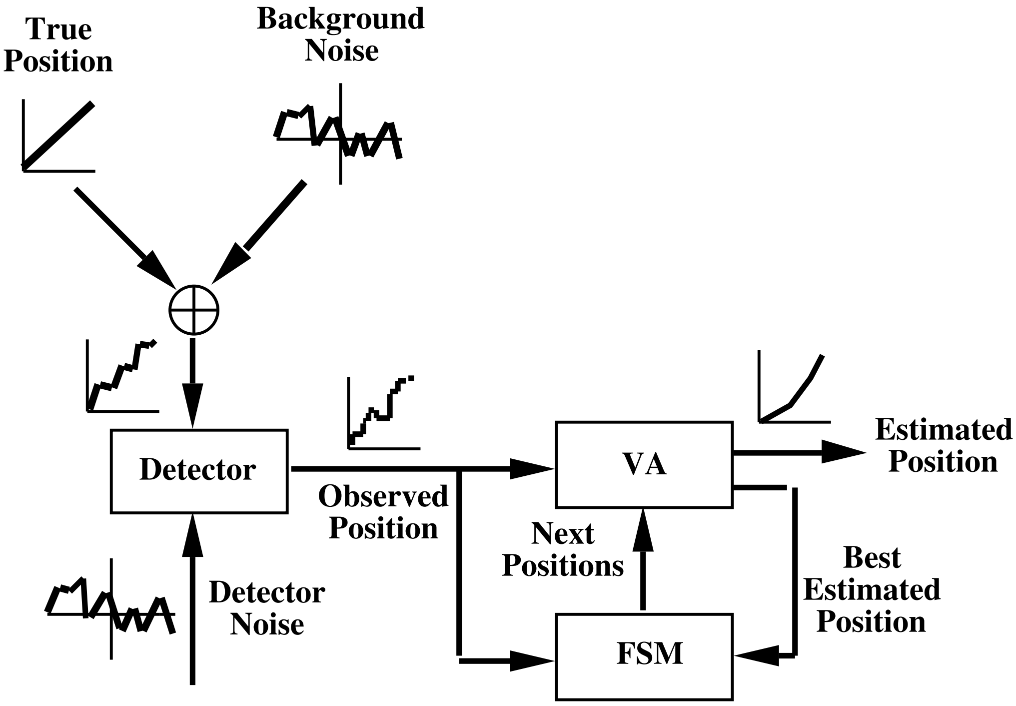


Figure 6. Model of Tracking system.

These problems were tackled by Demirbas in **[6]** where the VA was used to produce estimates of a target's state, be it in terms of speed, position or acceleration. The method uses an FSM to construct the trellis diagram, both the next states and the transitions between these states, using some a motion model. The motion model used is a approximation of non-linear motion. This recursive model is used to produce the next set of possible states that the object could go into along with the transitions into these states. A model of the tracking system proposed by Demirbas is shown in Figure 6 above.

Each state of the trellis represents a n-dimensional vector, for example a specific range, bearing and elevation position of a plane. Unlike the Kalman filter, this method does not use a linear motion model, but a non-linear one. It is also noticed that the number of states produced at the next stage in the trellis can increase or decrease, unlike most other applications of the VA where the number of states is fixed throughout the trellis. The VA can only consider a small amount of possible states that a target can move into, since in theory we can have a very large or even an infinite amount of possible states. The a_{nm} and π_n metrics can be worked out whilst the trellis diagram is being constructed and the b_n metrics depend on whether we are tracking the target in the presence of interference or with normal background noise. The VA is then used to estimate the most likely path taken through this trellis using the observation sequence produced by the detector. It was found that this method is far superior to the Kalman filter at estimating non-linear motion and it is comparable in performance to the Kalman filter when estimating linear motion. Also considered by Demirbas in **[6]** is the use of the Stack Sequential Decoding Algorithm instead of the VA, to produce fast estimates. Though this method is sub-optimum to the one using the VA.

This model can be applied to any non-linear dynamic system such as population growths, or economics, but Demirbas has applied this tracking method to tracking manoeuvrable targets using a radar system **[22,23]**. In these papers, Demirbas adapts the above system so that instead of a state in a trellis consisting of a position estimate in all dimensions, (range,bearing and elevation in this case), he splits these up into separate components and each of these components is estimated separately. So each of the dimensions has it's own trellis for estimation. The motion models for each of the trellises needed in this system are also presented in these papers and it is noted that the VA has to produce an estimate at each point so that the trellis diagrams can be extended.

The model can been taken one step further by accounting for missing observations, e.g when a plane goes out of radar range momentarily. This was dealt with by Demirbas in **[24]** where interpolating functions were used to determine the missing observations from the observations received. This set of observations was then given to the VA to estimate the true positions. Another adaption proposed in **[25]** uses the VA to estimate the position of a plane when any single observation received depends upon a number of previous observations, as in ISI.

Another tracking method has been developed by Streit and Barrett **[26]** where the VA is used to estimate the frequency of a signal, using the Fast Fourier Transform of the signal received. Unlike Demirbas, Streit uses a HMM tracker where the trellis is fixed and constructed from an observation model not a motion model. The states of the HMM represent a certain frequency. This method of tracking can be used to track changing frequency signals such as those used by some radars to detect where a target is.

All through Demirbas's work and Streit's work several improvements come to mind. One is the adaption of the VA so that it can deal with more than one target's position being estimated at the same time, i.e. multiple target tracking. Another improvement, particularly in Demirbas's work, is the use of an adaptive manoeuvrering model. Demirbas assumes in all his work that the manoeuvre is known to the tracker, but in reality this parameter would not be known at all, so it would have to be estimated as is done for the Kalman filter. This manoeuvrering parameter could be represented by a random variable, or it can be estimated by using a similar estimation scheme for the target's position, or estimated as suggested in **[27]**. Another problem not considered by Demirbas is adapting the noise models along with the finite state model. Though the tracking model given in Streit's work could be trained.

The problem of multiple target tracking using the VA has been solved for the tracking of time-varying frequency signals by Xie and Evans **[28]**. They describe a multitrack VA which is used to track two crossing targets. This system is an extension to the frequency line tracking method presented in **[26]**. Though this is not the same type of tracking as mentioned in Demirbas's work, it should be a simple matter of altering the parameters of the model to fit this type of tracking.

3.3 Recognition.

Another area where the VA could be applied is that of character and word recognition of printed and handwritten words. This has many applications such as post code and address recognition, document analysis, car licence plate recognition and even direct input into a computer using a pen. Indeed the idea of using the VA for optical character reading (OCR) was suggest by Forney in **[7]**. Also, Kunda *et al* **[29]** used the VA to select the most likely sequence of letters that form a handwritten English word. Another advantage with this model is that if a word produced by the VA is not in the system dictionary then the VA can produce the other less likely sequence of letters, along with their metrics, so that a higher syntactic/semantic model could determine the word produced. It can be easierly seen that a similar method would apply to the determination of a sequence of letters of a printed character as in OCR. In fact the VA can be used to recognize the individual characters or letters that make up a word. This is dealt with in **[30]** for chinese character recognition, though a similar method could be used to recognise English letters. The VA could be used at an even lower level of processing in the recognition phase than this, where it could be applied to character segmentation, i.e. determining which area of the paper is background, and which area contains a part of a letter. It was noted by the authors of **[30]** that the use of the VA and HMM in character recognition has not been widely investigated thus making this area an interesting one for further research.

The VA has also been used in other areas of pattern recognition, where it has been used to detect edges and carry out region segmentation of an image **[31]**. Another recognition problem is that of speech recognition, which unlike character and word recognition, the VA along with HMM's have been used widely **[8,9,32]**. The VA has also been used with neural networks to recognize continuous speech **[33]**.

4. Thesis Proposal.

Having looked at the various applications for which the VA has been used, it was decided that an untapped area of research is that of applying the VA and HMM's to the problem of signature verification. There are two distinct areas of signature recognition **[34]**, that of static and dynamic recognition. In static recognition a digitized image of the signature is used and features of the handwritting are extracted for their consistency, **[35]**. Measures are associated with these features and then the signature is classified as either a genuine signature and accepted, or a forgery and thus rejected. Some example measures are the overall length of the signature, slope measurements, **[35]**, and even trying to extract dynamic information from a static picture. It should be said that the difference between the signature given and a reference signature could not be used, since peoples signatures vary too much. In this respect, if the signature to be verified is the exactly the same as the reference signature then the signature given must be a forgery. Plamondon **[34]** mentions that the off-line verification of signatures is a problem which is hard to solve due to the inconsistencies in various static features, and trying to extract dynamic information from a static picture of a signature can prove difficult.

The other method of signature recognition, dynamic, is carried out whilst the person is actually writing the signature, i.e. on-line. The features extracted in this form of recognition can be either functions or parameters **[34]**. Typical functions that can be considered are the position of the pen at successive time intervals, the pressure applied at different times and the forces applied in different directions. Typical parameters used are the total time taken to sign, the order in which different parts of the signature were written, e.g. was the i dotted before the t was crossed, and even the area taken up by the signature.

It is proposed that further research should be carried out, and is currently under way, into the use of the VA and HMM's in the dynamic recognition of a signature. More specifically, using methods developed in the use of the VA in target tracking and dynamic non-linear system state estimation to that of estimating the sequence of positions that the pen follows. This is because the signature can be viewed as a highly manoeuvrable target whose true position is affected by noise, i.e. natural variations in the signature itself. It can also be viewed as being made up of not just one track, but of a combination of a number of separate tracks that correspond to the pen being lifted and put down again to finish another part of the signature. Each track that makes up a person' signature, would consist of a sequence of positions in a certain order, where the possible positions can be viewed as the states of a FSM. The VA would then be used to determine the most likely underlying sequence of positions, given the signature as the observation sequence. The measure produced by the most likely path could be used as a thresholding measure to determine whether the signature is a true signature, or a forgery.

Some of the problems that will have to be tackled are the type of noise model which should be used to represent natural variations in signatures, could it be assumed that it is guassian? Is there a better model that can be utilized? Another problem is that of determining the various metrics to be used, a_{nm},b_n, π_n. Can they be determined mathematically, or do the methods in **[8]** have to be used to determine these parameters. Also, how many initial signatures would be needed to give a reasonable training set for determining a particular signature's characteristics. Once a reasonable system is working then research into implementing the system in VLSI technology could be looked at since software simulations will be used to determine the performance of the new method. The VLSI implementation of the system could well be within the scope of this PhD.

Another feature that can be considered in this method is the order in which these tracks are produced, since it will be assumed that the order in which a person signs is consistent. It is also proposed to include training methods such as the Baum-Welch Re-estimation method **[8]**, so that the changes in a person's signature can be taken into account since everybodies signatures slowly but gradually change with time especially during the late teens and early twenties.

As an extension to the above work, the above methods could be applied to the dynamic recognition of handwritting which would be useful in pen based computer entry methods. In this application each letter of the alphabet can be viewed as having a unique signature, with the noise model taking into account not just the natural variations in an individuals writing, but those variations between different individuals writing.

5. Summary.

The Viterbi Algorithm has been described and various extensions to this have been covered. It was noted that the Viterbi Algorithm could be used on unknown finite state machines, refered to as Hidden Markov Models. Also, it was found that the Viterbi Algorithm has a wide field of applications, and not just in communications for which it was first developed. Particular attention was paid to it's application to target tracking and dynamic non-linear equations, and it has been suggested that this could be applied to the on-line verification of handwritten signatures. A thesis proposal was also given, concentrating on the application of the VA and HMM's to dynamic signature verification. Suggestions were also given, where appropriate, for further research in other areas which can be viewed as either extensions to the thesis proposal or beyond the scope of the proposed research.

6. Acknowledgements.

I wish to thank Chris Clarke, Darren Kerbyson, Tim Atherton and John McWhirter for their help, ideas and the useful discussions we had on the subject of the Viterbi Algorithm.

7. References.

1. Viterbi, A.J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Theory, April 1967; IT - 13 (2) : pp. 260 - 269.

2. Viterbi, A.J., Convolutional Codes and Their Performance in Communication Systems, IEEE Transactions on Communications Technology, October 1971; COM - 19 (5) : pp. 751 - 772.

3. Omura, J.K., On the Viterbi Decoding Algorithm, IEEE Transactions on Information Theory, January 1969; IT - 15 (1) : pp. 177 -179.

4. G. David Forney, Jr., Convolutional Codes II. Maximum-Likehood Decoding, Information and Control, July 1974; 25 (3) : pp. 222 - 266.

5. G. David Forney, Jr., Convolutional Codes III. Sequential Decoding, Information and Control, July 1974; 25 (3) : pp. 267 - 297.

6. Demirbas, K. *Target Tracking in the Presence of Interference*, Ph.D. dissertation, University of California, University of California, Los Angeles, 1981.

7. G. David Forney, Jr., The Viterbi Algorithm, Proceedings of the IEEE, March 1973; 61 (3) : pp. 268 - 278.

8. Rabiner, L.R. and Juang, B.H., An Introduction to Hidden markov Models, IEEE ASSP Magazine, January 1986; 2 (1) : pp. 4 - 16.

9. Rabiner, L.R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, February 1989; 77 (2) : pp. 257 - 286.

10. Sklar, B., Digital Communications - Fundamentals and Application, Prentice-Hall International, 1988.

11. Hoeher, P., TCM on Frequency-Selective Fading Channels : a Comparison of Soft-Output Probabilistic Equalizers, IEEE Global Telecommunications Conference and Exhibition : Globecom'90, 1990; 1 : pp. 376 - 381.

12. Bouloutas A., Hart G.W. and Schwartz, M., Two Extensions of the Viterbi Algorithm, IEEE Transactions on Information Theory, March 1991; IT - 37 (2) : pp. 430 - 436.

13. Fettweis, G. and Meyr, H., Feedforward Architectures for Parallel Viterbi Decoding, Journal of VLSI Signal Processing, June 1991; 3 (1/2) : pp. 105 - 119.

14. Lin, H.D. and Messerschmitt, D.G., Algorithms and Architectures for Concurrent Viterbi Decoding., Conference Record - International Conference on Communications - ICC'89, 1989; 2 : pp. 836 - 840.

15. Osawa H., Yamashita S., and Tazaki, S., Performance Improvement of Viterbi decoding For FM Recording Code, Electronics and Communication Engineering Journal, July/August 1989; 1 (4) : pp. 167 - 170.

16. Kobayashi, H., Application of Probabilistic Decoding to Digital Magnetic Recording Systems, IBM Journal of Research and Development, January 1971; 15 (1) : pp. 64 - 74.

17. Forney, G.D., Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference, IEEE Transactions on Information Theory, May 1972; IT - 18 (3) : pp. 363 - 378.

18. Kerpez, K.J., Viterbi Receivers in the Presence of Severe Intersymbol Interference, IEEE Global Telecommunications Conference and Exhibition : GLOBECOM90, 1990; 3 : pp. 2009 - 2013.

19. Lopes, L.B., Performance of Viterbi Equalizers for the GSM System, IEE National conference on Telecommunications : IEE Conference Publications no. 300, April 1989 : pp. 61 - 66.

20. Ungerboeck, G., Channel Coding with Multilevel/Phase Signals, IEEE Transactions on Information Theory, January 1982; IT - 28 (1) : pp. 55 - 67.

21. Du, J. and Vucetic, B., New M-PSK Trellis Codes for Fading Channels, Electronics Letters, August 1990; 26 (16) : pp. 1267 - 1269.

22. Demirbas, K., Maneuvering Target Tracking with Hypothesis Testing, IEEE Transactions on Aerospace and Electronic Systems, November 1987; AES - 23 (6) : pp. 757 - 766.

23. Demirbas, K., Manoeuvring-Target Tracking with the Viterbi Algorithm in the Presence of Interference, IEE Proceedings. Part F, December 1989; 136 (6) : pp. 262 - 268.

24. Demirbas, K., State Smoothing and Filtering Using a Constant Memory for Dynamic Systems with Nonlinear Interference and Missing Observations, Journal of the Franklin Institute, 1990; 327 (3) : pp. 503 - 511.

25. Demirbas, K., Trellis Representation and State Estimation for Dynamic Systems With a Kth Order Memory and Nonlinear Interference, Transactions of the ASME - Journal of Dynamic Systems, Measurement and Control, September 1990; 112 (3) : pp. 517 - 519.

26. Streit, R.L. and Barrett, R.F., Frequency Line Tracking Using Hidden Markov Models, IEEE Transactions on Acoustics, Speech, and Signal Processing, April 1990; ASSP - 38 (4) : pp. 586 - 598.

27. Bar-Shalom, Y. and Fortmann, T.E., Tracking and Data Association, Vol. 179, Academic Press, Mathematics in Science and Engineering, 1988.

28. Xie, X. and Evans, R.J., Multiple Target Tracking and Multiple Frequency Line Tracking Using Hidden Markov Models, IEEE Transactions on Signal Processing, December 1991; SP - 39 (12) : pp. 2659 - 2676.

29. Amlan Kunda, Yang He and Bahl, P., Recogintion of Handwritten Word: First and Second Order Hidden Markov Model Based Approach, Pattern Recognition, 1989; 22 (3) : pp. 283-297.

30. Jeng, B-S., Chang, M.W., Sun, S.W., Shih, C.H., and Wu, T.M., Optical Chinese Character Recognition with a Hidden Markov Model Classifier - A Novel Approach, Electronics Letters, August 1990; 26 (18) : pp. 1530 - 1531.

31. Pitas, I. A Viterbi Algorithm for Region Segmentation and Edge Detection. In *Computer Analysis of Images and Patterns*. Akademie-Verlag, Klaus Voss, D.C. and Sommer., G., pp. 129 - 134, 1989.

32. Lawrence R. Rabiner, Jay G. Wilpon and Frank K. Soong, High Performance Connected Digit Recognition Using Hidden Markov Models., IEEE Transactions on Acoustics, Speech and Signal Processing, August 1989; ASSP - 37 (8) : pp. 1214 - 1225.

33. Michael Franzini, Alex Waibel and Lee, K.F., Continuous Speech Recognition with the Connectionist Viterbi Training Procedure : A Summary of Recent Work. In *Lecture Notes in Computer Science : Artifical Neural Networks*. Springer-Verlag, Ch. 59, pp. 355 - 360, 1991.

34. Plamondon, R. and Lorette, G., Automatic Signature Verification and Writer Identification - The State of the Art, Pattern Recognition, 1989; 22 (2) : pp. 107-131.

35. Brocklehurst, E.R. Computer Methods of Signature Verification, Tech. Rept NPL-DITC-41/84 National Physical Laboratory, Teddington, England, May 1984.