performed for each of the 10 subjects by using utterances from one trial as the test data and the other two trials as training data. For all single-speaker experiments, 25 hidden units were used. The results are summed up as shown in Table III (a) and (b). Each highlighted entry indicates a contribution of over 2% misclassification for a particular tone.

The overall recognition rates for training and test data are given by 96.6% and 89% respectively. Tone 1 and 7, being characterized by their remarkably high pitch level, give the lowest error rates of recognition. Tone 3 also has a good recognition rate because of its distinctively sharp rise of pitch. Notable confusion is found within the lower nonentering tone series and between middle and lower entering tones since all of them have very close pitch levels as shown in Fig. 3. Similar reason also gives rise to classification errors between tone 1 and 5.

D. Experiments on Multispeaker Tone Recognition

Similar to the single-speaker case, the multispeaker tone recognition systems were trained with training data from two of the three trials of all subjects and tested with the remaining data sets. The combinations of training and test data were then permuted and two similar experiments were performed. The number of hidden units was 35 for all multispeaker experiments. The results are shown in Table IV (a) and (b). The confusion patterns of multispeaker experiments are very similar to the single-speaker ones except for the increased number of recognition errors in individual entries. The overall recognition rates for training and test data are given by 89.4% and 87.6% respectively.

VI. CONCLUSION

In this paper an efficient method for the tone recognition of isolated Cantonese syllables has been proposed. This method utilizes the relative pitch levels, temporal pitch variation patterns and duration of voiced portion as the main discriminating features. The feature vector consists of five components and a three-layer feedforward neural network is used for classification purpose. It has been shown by experimental results that the proposed method performs satisfactorily for both single-speaker and multispeaker recognition of a large vocabulary, and the average recognition accuracy is found to be 89.0% and 87.6% respectively.

Cantonese dialect is very rich in tones. Therefore tone identification plays an extremely important role in automatic recognition of Cantonese. If the lexical tone carried by an unknown syllable is correctly identified, the task of phonemes recognition will become much easier since the respective vocabulary size can be considerably reduced. The advantage of such a tone-oriented approach will be even more significant for connected speech recognition of Cantonese. This is because if the tone pattern of a polysyllabic utterance is known, the number of possible Cantonese phrases represented by that utterance can be minimized. Indeed, this process is commonly adopted in human perception of tonal speech.

REFERENCES


[6] “Yueh Yin Tu Li (a Cantonese syllabary with model pronunciations),” Edited by Ng Tor-Tai Chinese Language Research Centre, The Chinese Univ. of Hong Kong.


Adaptive WRLS-VFF for Speech Analysis

D. G. Childers, J. C. Principe, and Y. T. Ting

Abstract—The purpose of this correspondence is to show that an adaptive weighted recursive least squares algorithm with a variable forgetting factor (WRLS-VFF) will adjust the size of the data segment to be analyzed according to its time-varying characteristics, as during the transitions between vowels and consonants. The algorithm can accurately estimate the vocal tract formants, anti-formants, and their bandwidths, be used for glottal inverse filtering, perform voiced (V)/unvoiced (U) identification (S) classification of speech segments, estimate the input excitation (either white noise or periodic pulse trains), and estimate the instant of glottal closure.

I. ALGORITHM DESCRIPTION

We assume that the speech signal is generated by an autoregressive, moving average (ARMA) model

\[ y_k = - \sum_{i=1}^{a} a_i(k)y_{k-i} + \sum_{i=1}^{b} b_i(k)u_{k-i} + u_k \]  

(1)

Manuscript received August 13, 1993; revised November 15, 1994. This work was supported by NIH Grant NDCD RO1 DC00577, NSF Grant IRI-9215331, the University of Florida Center of Excellence Program in Information Transfer and Processing, and the Mind Machine Interaction Research Center. The associate editor coordinating the review of this paper and approving it for publication was Dr. Amos El-Jaroudi.

D. G. Childers and J. C. Principe are with the Department of Electrical Engineering, University of Florida, Gainesville, FL 32611 USA.

Y. T. Ting is with the Chung San Institute of Science and Technology, Taiwan, Republic of China.

IEEE Log Number 940229.
where $y_k$ denotes the $k$th sample of the speech signal, $u_k$ is the input excitation, $(p, q)$ are the order of the poles and zeros, respectively, and $a_i(k)$ and $b_j(k)$ are the time-varying AR and MA parameters, respectively. Measurement noise is ignored in this model but could be included [10], [11]. We assume that the values of $p$ and $q$ are predetermined. Note that the measured speech signal, $y_k$, depends on the input, $u_k$. The excitation, $u_k$, is usually considered to be white Gaussian noise. In this paper we allow $u_k$ to be either a zero-mean, white, Gaussian noise process, $u_k^w$, with variance $\sigma_u^2$, or a train of periodic pulses, $u_k^p$. We must estimate the input excitation, either $u_k^w$ or $u_k^p$, so that the ARMA parameters can be estimated accurately from $y_k$.

Let us define a parameter vector, $\theta_k$, and a data vector, $\phi_k$, by the following equations

$$\theta_k = [a_1(k), \ldots, a_p(k), b_1(k), \ldots, b_q(k)]$$

$$\phi_k = [y_{k-1}, \ldots, y_{k-p}, u_{k-1}, \ldots, u_{k-q}]$$

where the superscript $t$ denotes transpose. The corresponding estimated quantities will be denoted by $\hat{\theta}_k$. Then

$$\hat{y}_k = \phi_k^t \hat{\theta}_k + u_k$$

(4)

$$\hat{y}_k = \phi_k^t \hat{\theta}_k + \hat{u}_k$$

(5)

Let $r_k$ be the residual error of the ARMA process, namely,

$$r_k = y_k - \hat{y}_k = y_k - \phi_k^t \phi_k^{-1} \hat{\theta}_k - \hat{u}_k$$

(6)

The predicted signal, $\hat{y}_{k|k-1}$, determined from the estimated ARMA parameters at time $(k - 1)$, is

$$\hat{y}_{k|k-1} = \phi_k^t \hat{\theta}_{k-1}$$

(7)

Consequently, the prediction error is

$$e_k = y_k - \hat{y}_{k|k-1} - \hat{u}_k$$

(8)

Note that $\hat{u}_k$ is usually assumed to be unobservable at $(k - 1)$ and is set to zero [10], [11]. We will address this issue again later and modify the algorithm accordingly. An approach to deal with time varying regression coefficients is to minimize a weighted estimation (or residual) error [10], [14]

$$E_k = w(1, k) |\phi_k^t P_k^{-1} \hat{\theta}_k| + \sum_{i=1}^{k} w(i, k) r_i^2$$

(9)

where $P_k$ is an arbitrary real symmetric positive definite matrix. The weighting coefficient $w(1, k)$ is called a forgetting factor [10], [14]

$$w(i, k) = \prod_{j=i+1}^{k} \lambda_j, \quad i = 1, 2, \ldots, k - 1$$

$$= 1, \quad i = k$$

(10)

The coefficient $\lambda_k$ decreases the weight of past estimation errors provided $0 < \lambda_k < 1$. Note for fixed $\lambda_k = \lambda$ that $w(i, k)$ becomes an exponentially weighted coefficient, e.g., $\lambda^{k-1}, \lambda^{k-2}, \ldots, \lambda, 1$. Consequently, the estimation error, $E_k$, becomes an exponentially weighted sum of squares of the estimation errors [3], [5], [14], i.e.,

$$E_k = \sum_{i=1}^{k} \lambda^{i-1} (y_i - \hat{y}_i)^2 + \lambda^{k-1} |\phi_k^t P_k^{-1} \hat{\theta}_k|$$

(11)

Minimizing the least square weighted estimation error, $E_k$, with respect to the ARMA parameter vector, $\hat{\theta}_k$, assuming that $\hat{u}_k$ is available, gives the following algorithm [10], [11], [14].

Residual error

$$r_k = y_k - \phi_k^t \hat{\theta}_k - \hat{u}_k$$

(12)

Prediction error

$$e_k = y_k - \phi_k^t \hat{\theta}_{k|k-1} - \hat{u}_k$$

(13)

Gain update

$$K_k = P_k \phi_k \lambda_k |\lambda_k^{-1} + \phi_k^t \phi_k P_k^{-1} \hat{\theta}_{k|k-1}|^{-1}$$

(14)

Parameter update

$$\hat{\theta}_k = \hat{\theta}_{k|k-1} + K_k e_k$$

(15)

Covariance matrix

$$P_k = \lambda_k^{-1} P_{k|k-1} - K_k \phi_k^t P_k^{-1} \phi_k K_k$$

(16)

The above algorithm updates the ARMA parameters at each instant $k$ and has been shown to be stable and to provide a unique solution [3], [5], [10], [11], [14]. $E_k$ can be calculated recursively, thereby allowing $\lambda$ to be calculated recursively. Since $w(k, k) = 1$ and $w(k, k-1) = \lambda_k$, then (9) leads to the following recursive expression

$$E_k = \lambda_k E_{k-1} + (y_k - \hat{y}_k)^2 + w(1, k) \times |\hat{\theta}_k^t P_k^{-1} \hat{\theta}_k - \hat{\theta}_{k|k-1}^t P_k^{-1} \hat{\theta}_{k|k-1}|$$

(17)

Then rearranging, we have

$$\lambda_k = \frac{E_k - (y_k - \hat{y}_k)^2 - w(1, k) \times |\hat{\theta}_k^t P_k^{-1} \hat{\theta}_k - \hat{\theta}_{k|k-1}^t P_k^{-1} \hat{\theta}_{k|k-1}|}{E_{k-1}}$$

(18)

Using the previous expressions leads to the following approximation to compute and update $\lambda_k$

$$\lambda_k = \frac{E_k}{E_{k-1}} - \frac{\phi_k^t \phi_k}{E_{k-1}} \left[1 - \phi_k^t \phi_k^{-1} \lambda_k\right]$$

(19)

where we have assumed that the third term in (18) becomes negligible with increasing $k$ since $w(1, k) = \lambda_k \lambda_{k-1} \cdots \lambda_2 \ll 1$ for $0 < \lambda_k < 1$.

In (19), $\lambda_k$ depends upon the ratio of the weighted estimation errors at step $k$ and $k - 1$ $(E_k$ and $E_{k-1})$, and on the prediction error at step $k$. A simplifying strategy to compute $\lambda_k$ can be defined if we require that $E_k = E_{k-1} = \cdots = E_1$ [4]. This means that the forgetting factor compensates at each step $k$ for the new error information in the latest measurement. This has the added benefit of normalizing the forgetting factor with respect to the same error information, yielding

$$\lambda_k = 1 - \frac{\phi_k^t \phi_k}{E_1} - \left[1 - \phi_k^t \phi_k^{-1} \lambda_k\right]$$

(20)

The WRLS-VFF algorithm is specified by a set of equations similar to those for the WRLS algorithm. However, the weighting factor, $\lambda_k$, is estimated by (20). We recommend that the estimation of $E_1$ be determined using an initial block of data. The minimum length of the window should be related to the size of the ARMA model, therefore, we further limit the smallest value of $\lambda$ by

$$\lambda_{min} = 1 - \frac{1}{N_e}, \quad \text{if} \quad \lambda_k < \lambda_{min}, \quad \text{then} \quad \lambda_k = \lambda_{min}$$

(21)

We have determined empirically that the value of $N_e$ should be approximately twice the model order to obtain good results.

We now estimate the residual and prediction errors as well as the gain, ARMA parameters, the covariance matrix, and the excitation, $u_k$. Furthermore, we must determine whether $\hat{u}_k$ is $\hat{u}_k^w$ or $\hat{u}_k^p$. Since the previous results assumed that we had an estimate for $u_k$, we now modify our definitions of the residual and prediction errors to account for the fact that an estimate for the excitation, $\hat{u}_k$, is not available at $k$. Thus, we define the following two errors

$$\xi_k = y_k - \hat{u}_{k|k-1} = y_k - \phi_k^t \hat{\theta}_{k|k-1}$$

(22)

$$\xi_k = y_k - \hat{u}_k = y_k - \phi_k^t \hat{\theta}_k$$

(23)
The equation for the forgetting factor (eq. (20)) remains as before with \( r_k \) replaced by \( \xi_k \). Once we make a decision regarding the input excitation, we define a new error as

\[
\begin{align*}
\epsilon_k &= \begin{cases} 
\xi_k & \text{for } \hat{u}_k = u_k^V \\
\xi_k - u_k^V & \text{for } \hat{u}_k = u_k^N
\end{cases} 
\end{align*}
\]  

(24)

We update the parameter estimates and the covariance matrix estimate. The data/excitation vector \( \phi_k \) is updated using the new speech sample \( y_k \) and \( \hat{u}_k \).

From (20), we see that an increase in the prediction error, \( \epsilon_k \), results in a decrease in \( \lambda_k \). A small value of \( \lambda_k \) indicates that the input has undergone an abrupt change, typically indicating that a glottal pulse excitation has occurred. Hence, we can determine the time of occurrence of a pulse by determining the instant at which \( \lambda_k \) falls below a minimum threshold \( \lambda_0 \). When this occurs we set \( \hat{u}_k = u_k^V \) and \( \hat{u}_k^V = 0 \). The magnitude of the pulse excitation is determined from the prediction error \( \xi_k \), at the estimated time of the input pulse by assuming that \( \xi_k = \hat{u}_k^V \) [10]. Thus,

\[
\hat{u}_k^V = y_k - \hat{u}_k^V \phi_k^{\dagger} \hat{u}_{k-1}.
\]

(25)

For white noise input, \( \lambda_k \) is close to unity upon convergence [14]. Under this condition the residual error, \( r_k \), of the adaptive process can be used as the estimate of the white noise input, \( \hat{u}_k^V \), as indicated in Morikawa's method [12], [13], i.e., from (22)

\[
r_k = y_k - \hat{y}_k = \xi_k (1 - \phi_k^{\dagger} K_k) = \hat{u}_k^V
\]

(26)

and \( \hat{u}_k^V = 0 \). This is similar to estimates used previously [5], [12], while a different approach was used in [10]. However, our approach uses only one adaptive algorithm instead of two as in [10].

In order to select the threshold value, \( \lambda_0 \), that determines whether the speech was voiced or unvoiced, we adopted the following strategy. We compute a running average of the last \( M \) values of the forgetting factor as follows, where \( M \) is typically the number of samples in

Fig. 1. Speech (a), differentiated electroglottographic (DEGG) signal (b), residual error (c), and variable forgetting factor (VFF) (d).
TABLE I
Adaptive WRLS-VFF Algorithm with Input Estimation

Prediction error:
\[ y_k = x_k - \hat{\phi}_0^T \hat{\phi}_0^{-1} \hat{\phi}_0^{-1} \]

Gain update:
\[ K_k = P_k - \hat{\phi}_0^T [\hat{\phi}_0^{-1} + \hat{\phi}_0^{-1} K_k - 1] \hat{\phi}_0^{-1} \]

Forgetting Factor:
\[ \lambda_k = 1 - \xi_k^2 / (1 - \xi_k^2 K_k^2 / E_i) \]

Input estimate:

a) Pulse input
If \( \lambda_k < \lambda_0 \) then
\[ \hat{u}_k = \hat{u}_k = \hat{u}_k = \frac{y_k - \hat{\phi}_0 \hat{\phi}_0^{-1} \hat{\phi}_0^{-1} \hat{u}_0}{\hat{\phi}_0^{-1} K_k} \]

b) White noise input
If \( \lambda_k > \lambda_0 \) then
\[ \hat{u}_k = \hat{u}_k = \hat{u}_k = \frac{y_k - \hat{\phi}_0 \hat{\phi}_0^{-1} \hat{\phi}_0^{-1} \hat{u}_0}{\hat{\phi}_0^{-1} K_k} \]

Parameter update:
\[ \hat{\phi}_0 = \hat{\phi}_0 - K_k (\hat{\phi}_0 - \hat{\phi}_0 \hat{\phi}_0^{-1} \hat{\phi}_0^{-1} \hat{u}_k) \]

Covariance matrix:
\[ P_k = \lambda_k^{-1} [P_k - K_k \hat{\phi}_0 \hat{\phi}_0^{-1} P_k^{-1}] \]

A frame
\[ L_k = \frac{1}{M} \sum_{i=0}^{M-1} \lambda_k \]

If \( L_k < 0.9 \), then \( \lambda_0 = 0.99 L_k \); if \( L_k > 0.9 \), then \( \lambda_0 = 0.9 L_k \); should \( \lambda_0 < \lambda_{\text{min}} \), then \( \lambda_0 = \lambda_{\text{min}} \). Thus, the threshold may be made adaptive, whereby, it is adjusted on a frame-by-frame basis.

We summarize the WRLS-VFF algorithm including input estimation in Table I. The algorithm in Table I differs from previous algorithms in that we: 1) update the variable forgetting factor at each step, 2) let the prediction error, \( \xi_k \), be the estimate of the noise excitation, \( \hat{\phi}_0 \), and 3) let the residual error, \( r_k \), be the estimate for the noise excitation, \( \hat{\phi}_0 \). The algorithm may be shown to be stable and to provide a unique solution following the method given in Appendix III of [10]. Several factors affect the convergence of the WRLS-VFF algorithm: 1) model order, 2) stationarity of the signal, and 3) size of the data analysis interval. We have assumed that the model order may be determined a priori and that the data analysis interval is sufficiently large for the algorithm to work. One can show from (22) and (23) that
\[ c_k^2 = r_k^2 [1 + \lambda_k^{-1} \hat{\phi}_0 \hat{\phi}_0^{-1} \hat{\phi}_0^{-1}]^2. \]

If 1) the covariance matrix \( P_k^{-1} \) is positive definite and 2) \( \hat{\phi}_0 \hat{\phi}_0^{-1} \hat{\phi}_0^{-1} \) converges to zero as \( k \) goes to infinity, then the variance of the prediction error, \( \xi_k \), converges to the variance of the residual error, \( r_k \). These two conditions may be shown to be satisfied for a stationary ARMA process with white noise, zero mean excitation [9], [10], [16] under the assumption that \( \lambda_k \) approaches unity.

The parameter estimation error is usually large when there is a large glottal open phase during speech production. Small values of \( \lambda_k \) occur at the instants of glottal closure where the prediction error is maximum. Fig. 1 shows that the minima of \( \lambda_k \) occur at nearly the same instant as the negative peaks of the differentiated electroglottographic (DEGG) signal. Since the large negative peaks in the DEGG are known to occur at glottal closure (or the minimum glottal aperture) [11], [7], then comparisons such as those in Fig. 1 serve as a partial validation of the WRLS-VFF algorithm. With such validation we have concluded that the minima of \( \lambda_k \) can be used to predict the instant of glottal closure, and, therefore, the presence of voiced excitation. When \( \lambda_k \) does not fall below the threshold value, \( \lambda_0 \), then we may decide that the excitation is unvoiced.

The WRLS-VFF algorithm requires on the order of \( (5(p + q)^2 + 6(p + q)) \) floating point multiplications and additions (flops) per data point for an ARMA model [16]. By using the idea of shift low rank the WRLS-VFF algorithm can be implemented with \( O(N(p + q)) \) flops instead of \( O(N(p + q)^2) \) [8]. Consequently, the WRLS-VFF algorithm can be made of the same complexity as other recursive algorithms.

II. CONCLUSION

We have implemented a closed phase adaptive WRLS-VFF algorithm, which is able to track formants and anti-formants for various speech sounds, e.g., vowels, diphthongs, nasals and some consonants for either isolated words or sentences. In [16] we have shown that this algorithm accomplishes these tasks with greater accuracy than other methods, such as LPC [9], Iterative Inverse Filtering (ITIF) [15], the two-stage least squares modified Yule–Walker equations (MYWE) [6], and the recursive algorithms: sequential estimation ARMA (SEARMA) [13], weighted recursive least squares (WRLS) [3], weighted least squares lattice (WLSL) [8], and modified WRLS (MRLS) [4]. The WRLS-VFF algorithm has also been shown to estimate accurately the values of the formants and their bandwidths for vowels and fricatives [2]. The algorithm is able to perform glottal inverse filtering automatically as well as or better than other procedures that require two channels of data (e.g., speech and EGG) or that require two-passes of the data. The WRLS-VFF algorithm uses the VFF and the estimation error to determine the time at which an excitation pulse occurs and excludes that interval from parameter updating. This provides an improved estimation of the vocal tract transfer function and, consequently, an improved estimation of the glottal volume-velocity waveform.

REFERENCES

On the Complexity of Explicit Duration HMM’s

Carl Mitchell, Mary Harper, and Leah Jamieson

Abstract—We introduce a new recursion that reduces the complexity of training a semi-Markov model with continuous output distributions. We show that the cost of training is proportional to $M^2 + D$, compared to $M^2 D$ with the standard recursion, where $M$ is the observation vector length and $D$ is the maximum allowed duration.

I. INTRODUCTION

In the course of our work on parallelizing HMM (hidden Markov model) algorithms [1], we have discovered a new recursion that reduces the number of vector operations in a continuous output, explicit duration HMM. Following a definition of terms in Section II, we develop the result in three stages. In Section III, we demonstrate that the complexity of the forward, backward, and Viterbi algorithms is $O(NKT(M^2 + D))$ using known recursions, where $T$ is the number of observations, $N$ is the number of states, $K$ is the average number of predecessors across all states, and $M$ is the observation vector length. In Section IV, we show that the complexity of re-estimation is $O(NKT(M^2D))$ using a recursion given by Levinson [2]. This demonstrates that the cost of explicit duration modeling is proportional to $D$, and not $D^2$ as is often quoted in the literature [3], [4]. Finally, we introduce a new recursion in Section V that lowers the complexity of re-estimation (and therefore training) from $O(NKT(M^2D))$ to $O(NKT(M^2 + D))$ for continuous HMM’s. This improved complexity is achieved by delaying vector operations until $D$ scalar weights have been accumulated.

II. TERMINOLOGY

We use a semi-Markov model [5], [6] where the number of observations drawn from the same density (i.e., duration) is modeled explicitly. We assume that the observations $\{O_1, \cdots, O_T\}$ are normally distributed, where boldface type indicates that the vector refers to a vector. The output symbols are emitted during the transitions between states (i.e., a Mealy model) as shown in Fig. 1. Let $K$ be defined as the average number of transitions to a state, where a transition exists from state $i$ to state $j$ if the transition probability

\[ a_{ij} \] is greater than zero. There are $NK$ transitions in the model

\[ \sum_{j=1}^{N} \sum_{i=1, i \neq j}^{N} 1 = NK. \]

For a fully connected model, $K = N$. Arguments similar to those in this paper can be used for the case where the observations are instead produced from the states (i.e., a Moore model) or the output distributions are not Gaussian. Definitions for terms used in this paper are shown in Table I. Let $d_{i,j}(\tau)$ be the probability mass function for the duration of the transition from state $i$ to state $j$. We assume that the maximum duration is $D$. A duration of zero (i.e., a null transition) is permitted as long as a closed loop of null transitions does not occur. The equations given in this paper can be modified slightly to allow for zero duration.

III. COMPLEXITY OF THE FORWARD, BACKWARD, AND VITERBI ALGORITHMS

Training a hidden Markov model consists of iteratively improving an estimate of $\lambda$, the set of model parameters. An improved estimate can be conveniently and efficiently expressed in terms of the forward and backward probabilities.

Define $\alpha_t(j)$, the forward probability, as follows:

\[ \alpha_t(j) = P(O_1, O_2, \cdots, O_t, \text{ state } j \text{ ends at } t \mid \lambda) \]

\[ \alpha_0(j) = \begin{cases} 1 & \text{if } j \text{ is the initial state} \\ 0 & \text{otherwise} \end{cases} \]

For $t = 1, \cdots, T$, $j = 1, \cdots, N$:

\[ \alpha_t(j) = \sum_{i=1, i \neq j}^{N} \min_{r} \alpha_{t-r}(i) a_{i,j} d_{i,j}(\tau) \prod_{p=1}^{r} b_{i,p}(O_{t-p+1}) U_{i,i}(\tau) \]

Evaluating $b_{i,j}(O_1)$ requires $O(M^2)$ operations since we assume that all $M^2$ elements of covariance matrix $a_{i,j}$ could be nonzero. The product of output probabilities can be calculated recursively [2]:

\[ U_{i,i}(\tau) = \prod_{p=1}^{r} b_{i,p}(O_{t-p+1}) = b_{i,j}(O_{t-r+1}) U_{i,i}(\tau - 1) \]

for $r = 1, \cdots, \min(D, t)$, where $U_{i,i}(0) = 1$.

\[ \text{evaluating } b_{i,j}(O_1) \text{ requires } O(M^2) \text{ operations since we assume that all } M^2 \text{ elements of covariance matrix } a_{i,j} \text{ could be nonzero.} \]

\[ \text{The product of output probabilities can be calculated recursively [2]:} \]

\[ U_{i,i}(\tau) = \prod_{p=1}^{r} b_{i,p}(O_{t-p+1}) = b_{i,j}(O_{t-r+1}) U_{i,i}(\tau - 1) \]

for $r = 1, \cdots, \min(D, t)$, where $U_{i,i}(0) = 1$. \[ \text{(1)} \]