15th Triennial World Congress, Barcelona, Spain

OPTIMAL FILTERING WITH DELAYED AND NON-DELAYED MEASUREMENTS

Héctor P. Rotstein *

* *Navigation Group - RAFAEL*
and the Dept. of Electrical Engineering, The Technion
Haifa 32000, Israel

Abstract: This paper deals with an estimation problem with two different kinds of sensors. The first sensor is characterized by a relatively fast sampling rate and a small time delay. The second sensor is characterized by a slow sampling rate and a large time delay. A typical example of the latter is a "soft sensor." The paper provides a solution to the estimation problem in a Kalman Filtering setup, and discusses implementation details using square-root UD factorization.

Keywords: Kalman Filtering, Estimation, Time-delays

1. INTRODUCTION

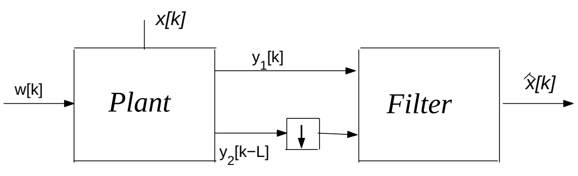

Fig. 1. Filtering setup with non-delayed/delayed measurements

This note considers the filtering problem illustrated in Figure 1. The objective is to estimate the state $x[k]$ of a discrete-time, linear but possibly time-varying system $\mathcal{P}$, driven by a white noise Gaussian signal $w[k]$. Two measurements are available from the system:

- The measurement $z_1[k]$, which is available without delay and takes values for each k, and
- The measurement $z_2[k_n - L]$, which has a delay of L samples and the sequence of indeces $\{k_n\}$ is such that $k_n - k_{n-1} \geq L$. An important assumption is that the validity time $k_n - L$ is known at least at time $\hat{k}_n - L$, although the measurement becomes available only at k_n. The filter algorithm can hence take any required provisions, like storing or processing information at the fixed time $\hat{k}_n$.

This information scheme may appear whenever the measurement signals z_1 come from actual "physical" measurements while the signals z_2 are derived using more or less elaborated computational algorithms. As a typical example, the latter case may correspond to signals z_2 obtained from a "soft sensor." Specifically, this soft sensor may be a vision system, in which images are processed by an image processing algorithm to generate a position or motion update. The time delay L may be due to the computations required for obtaining the measurement. The fact that measurements are obtained at most every other L samples reflects computational limitations or the absence of information that could be processed by the soft sensor. When z_2 is not present, the problem above reduces to a Kalman filtering problem (Anderson and Moore, 1979; Mendel, 1995; Gelb, 1974). When z_1 is not present, the problem reduces to a Kalman filtering problem with delayed measurements, which is a simple extension of the standard Kalman filter (Gelb, 1974). When both measurements are presents, then the problem becomes more complicated and a careful tracking of the estimates and their covariances is required. The purpose of this paper is to solve this problem and to provide a computational robust algorithm, based on Bierman's UD factorization (Bierman, 1977) for implementing the solution.

Assuming that $\mathcal{P}$ is a finite-dimensional, linear, possibly time-variant system, we get the model:

$$\begin{aligned} x[k+1] &= \Phi(k+1,k)x[k] + w[k] \\ z_1[k] &= H_1(k)x[k] + v_1[k] \\ z_2[k_n] &= H_2(k_n - L)x[k_n - L] + v_2[k_n - L] \end{aligned}$$

The noise signals $w[\cdot]$, $v_1[\cdot]$, $v_2[\cdot]$ verify the standard Kalman-filter assumptions (Mendel, 1995), namely they are white Gaussian sequences with covariance $Q[k]$ and $R_i[k]$, respectively. A corresponding assumption holds with respect to the initial state $x[0]$ of the system. The filter $\mathcal{F}$ in Fig. 1 processes the measurements to generate an estimate $\hat{x}[k/k]$ of the state $x[k]$, and we want $\tilde{x}[k/k] \doteq x[k] - \hat{x}[k/k]$ to be optimal in the Kalman filter sense. In principle, this can be done as follows:

(1) At time k, store $\hat{x}[k/k]$ and the error covariance $P[k/k]$.
(2) Generate the estimates $\hat{x}[k+l/k+1]$, $l = 1,\ 2,\ \cdots,\ L$ by using a standard Kalman filtering algorithm on the measurements $z_1[k+l]$.
(3) At time $k+L$ the measurement $z_2[k]$ becomes available. Update the stored estimate $\hat{x}[k/k]$ and the corresponding error covariance, using the Kalman filter update formula.
(4) Reprocess all measurements $z_1[k+l]$ using again the Kalman filtering formulas up to the present instant $k+L$.

The approach discussed above is optimal but not feasible in applications with limited computational or memory resources. Instead, one would like to obtain a *recursive* algorithm for processing the data, more amicable to real-time implementation. Alternatively, one could use *lifting* arguments as in (Mirkin *et al.*, 1999) to formulate the problem as a standard Kalman filtering one in the lifted domain. In this way, the standard formulas can be used to "solve" the problem; it remains to be seen if this route can be used to generate a recursive solution as the one considered next.

Processing measurements with delays like z_2 is relatively straightforward within the Kalman filtering algorithm. The optimal solution consists on generating an estimate $\hat{x}[k/k]$ at the time $k+L$ when the measurement becomes available, and then propagate the solution in time using the dynamics matrix $\Phi(k+L,k)$. On the other hand, given an estimate $\hat{x}[k/k]$ and "future" measurements $z_1[k+l]$, then the estimate can be improved by using an algorithm called *smoothing*. This algorithm generates estimates of the form $\hat{x}[k/k+l]$, and the estimates can be optimal in the Kalman filtering sense (see (Mendel, 1995) and the references therein). The solutions for delayed and smooth estimates suggest the following approach for the problem at hand:

(1) At time k, store $\hat{x}[k/k]$ and the error covariance $P[k/k]$.
(2) Generate the estimates $\hat{x}[k+l/k+l]$, $l = 1,\ 2,\ \cdots,\ L$ by using a standard Kalman filtering algorithm on the measurements $z_1[k+l]$.
(3) Generate the smooth estimates $\hat{x}[k/k+l]$ with corresponding covariance $P[k/k+l]$, $l = 1,\ 2,\ \cdots,\ L$. This estimates are computed recursively by using $\hat{x}[k+l/k+l-1]$.
(4) At time $k+L$ the measurement $z_2[k]$ becomes available. Update the smooth estimate $\hat{x}[k/k+l]$ and the corresponding error covariance, using the Kalman filter update formula.
(5) Propagate the estimate to current time using the dynamics matrix $\Phi(k+L,k)$ and likewise for the error covariance.

Notice that the overall algorithm implements a filtering/smoothing/propagation strategy. The intuitive idea is that all information from $k+1$ to $k+L$ obtained from the z_1's have already been extracted when computing the smooth estimate $\hat{x}[k/k+L]$, and hence propagation gives the best possible estimate. We will show by an example that this intuition is not correct. Instead, we propose a new filtering/smoothing/filtering algorithm, that produces optimal estimates under the constraints imposed by the processing. The basic idea is to formulate the optimal estimation problem and then show how the quantities required in the solution can be computed recursively.

Preliminaries and Notation

The symbol $\hat{x}[k/l]$ denotes an estimate of the state at time k based on all measures z_1 up to time l. Recall from basic Kalman Filter theory that $\hat{x}[k+1/k]$ is the *a priori* estimate computed as:

$$\hat{x}[k+1/k] = \Phi(k+1,k)\hat{x}[k/k],$$

while $\hat{x}[k+1/k+1]$ is the *a posteriori* estimate

$$\begin{aligned} \hat{x}[k+1/k+1] &= \hat{x}[k+1/k] + \\ &\quad K_1(k+1)\left(z_1[k] - H_1(k)\hat{x}[k+1/k]\right). \end{aligned}$$

Here $K_1(k+1)$ denotes the optimal Kalman gain (see, e.g., (Grewal and Andrews, 1993), ch. 4).

Smooth estimates will be of the form $\hat{x}[k/k+l]$, and can be computed recursively using:

$$\hat{x}[k/k+l] = \hat{x}[k/k+l-1] \\ +N(k/k+l)\tilde{z}_1[k+l/k], \quad l = 1,\ 2,\ \cdots$$

where

$$\tilde{z}_1[k+l/k] = z_1[k+l] \\ -H_1(k+l)\hat{x}[k+l/k+l-1]$$

The matrix $N(k/k+l)$ can be thought of as the optimal smoothing gain. The formula for the state, together with a recursive expression for $N(k/k+l)$ can be found in (Mendel, 1995). Whenever the measurement $z_2[m]$ is also involved, we will use the notation $\hat{x}[k/l,m]$. The same notation is used for the estimation error $\tilde{x}[\cdot]$. The notation for covariance matrix becomes quite involved. If $\tilde{x}[k_1/l_1]$ and $\tilde{y}[k_2/l_2]$ are two stochastic variables, then:

$$P_{xy}(k_1,k_2/l_1,l_2) \doteq \mathcal{E}\tilde{x}[k_1/l_1]\tilde{y}[k_2/l_2]^T,$$

where $\mathcal{E}$ denotes the expected value operator. If $x = y$, $k_1 = k_2$ or $l_1 = l_2$, then a single index will be used, e.g.:

$$P_{\tilde{x}}(k_1/l_1,l_2) = \mathcal{E}\tilde{x}[k_1/l_1]\tilde{x}[k_1/l_2]^T.$$

To further simplify notation, we will replace $P_{\tilde{x}\tilde{x}}$ by P. Notation is further complicated by the dependence on both z_1 and z_2.

2. AN INTUITIVE SOLUTION THAT DOES NOT WORK

As explained in the introduction an intuitive solution would proceed as follows for the case $l = 1$:

(1) Propagate $\hat{x}[k/k]$ to $\hat{x}[k+1/k]$.
(2) Update $\hat{x}[k+1/k]$ to $\hat{x}[k+1/k+1]$.
(3) Smooth $\hat{x}[k+1/k+1]$ to $\hat{x}[k/k+1]$.
(4) Propagate $\hat{x}[k/k+1]$ to ??

One is tempted to replace the question marks on the end by $\hat{x}[k+1/k+1]$ but unfortunately this is in general incorrect. It is possible to show that this estimate is optimal only if $Q[k]H_1(k+1)^T = 0$. The physical meaning of this equality is that the states generating the measurement are not affected by process noise and consequently their estimate is not degraded from sample time to sample time. Notice that for a multiple-stage smoother and under the assumption that the system is observable, the process noise should affect eventually the outputs.

3. OPTIMAL COMBINATION OF OLD MEASUREMENTS

The estimate of the state at time $k+l$ is to be estimated based on the measurement $z_2[k]$ by using:

$$\hat{x}[k+l/k+l,k] = \hat{x}[k+l/k+l] \\ +K_2(k/k+l)\left(z_2[k] - H_2(k)\hat{x}[k/k+l]\right) \\ = \hat{x}[k+l/k+l] + K_2(k/k+l)H_2(k)\tilde{x}[k/k+l] \\ +K_2(k/k+l)v_2[k].$$

The estimation error is hence:

$$\tilde{x}[k+l/k+l,k] = \tilde{x}[k+l/k+l] \\ -K_2(k/k+l)\left[H_2(k)\tilde{x}[k/k+l] - v_2[k]\right].$$

Notice that the matrix $K_2(k/k+l)$ is still to be defined. Using the notation introduced in Section 1, the covariance of the estimation error as a function of $K_2(k/k+l)$ can be evaluated to be:

$$P^{K_2}[k+l/\{k+l,k\}] = P[k+l/k+l]+ \\ K_2(k/k+l)H_2(k)P(k/k+l)H_2(k)^T K_2(k/k+l)^T \\ +K_2(k/k+l)R_2(k)K_2(k/k+l)^T \\ -P(k+l,k/k+l)H_2(k)^T K_2(k/k+l)^T \\ -K_2(k/k+l)H_2(k)P(k+l,k/k+l)$$

It is a well known fact (Mendel, 1995) that the minimum of the trace of this matrix, and consequently the optimal solution, can be found by setting:

$$K_2(k/k+l) = P[k+l,k/k+l]H_2(k)^T \\ \left[H_2(k)P(k/k+l)H_2(k)^T + R_2(k)\right]^{-1}$$

Note that, as in the standard Kalman Filter, the matrix to be inverted is the covariance of the measurement error, except that now the computation is performed after smoothing:

$$P_{\tilde{z}}[k/k+l] = H_2(k)P(k/k+l)H_2(k)^T + R_2(k)$$

Plugging the optimal gain in the expression for the covariance above, yields after some simplifications:

$$P[k+l/k+l,k] = P[k+l/k+l]- \\ P[k+l,k/k+l]H_2(k)^T P_{\tilde{z}}^{-1} H_2(k)P[k,k+l/k+l].$$

The following matrices are involved in the computation of the optimal gain and the optimal covariance:

(1) $P[k+l/k+l]$: computed by the Kalman Filter, by using the measurements $z_1[j]$, $j = k, \cdots, k+l$.

(2) $P[k/k+l]$: computed by the optimal smoother, by using the measurements $z_1[j]$, $j = k, \cdots, k+l$.
(3) $P[k+l, k/k+l] = \mathcal{E}\tilde{x}[k+l/k+l]\tilde{x}[k/k+l]^T$: a recursive formula for this term is developed next.

Using lengthy calculation, one can get:

$$P[k+l, k/k+l] = (I - K_1(k+l)H_1(k+l)) \\ \Phi(k+l, k+l-1)P[k+l-1, k/k+l-1],$$

showing that the covariance matrix can be computed via a simple recursion, initialized with $P[k, k/k] = P[k/k]$. It is informative to modify the recursion to:

$$\hat{\Phi}[k+l, k/k+l] = (I - K_1(k+l)H_1(k+l)) \\ \Phi(k+l, k+l-1)\hat{\Phi}[k+l-1, k/k+l-1],$$

initialized with $\hat{\Phi}[k, k/k] = I$. Then:

$$P[k+l, k/k+l] = \hat{\Phi}[k+l, k/k+l]P[k, k],$$

and also:

$$K_2(k/k+1) = \hat{\Phi}[k+l, k/k+l]K_2(k)$$

4. OPTIMAL COMBINATION WITHOUT SMOOTHING

The purpose of this section is to relax the assumption that the validity time of the measurement is known before the measurement actually becomes available. This problem may be of interest whenever measurements are not synchronized. In principle, the problem could be solve by "smoothing" the data over the entire lap where the measurement is expected. However, we will assume that real-time or memory constraints prevent make this approach infeasible. Since validity time is not known in advance, then the state estimate cannot be smooth and hence the optimal estimate will take the form:

$$\check{x}[k+l/k+l, k] = \check{x}[k+l/k+l] + \\ \check{K}_2(k/k+l)H_2(k)\tilde{x}[k/k+l] + \check{K}_2(k/k+l)v_2[k].$$

The estimate and filter gain are called $\check{x}[\cdot/\cdot]$ and $\check{K}(\cdot/\cdot)$ to differentiate them from the estimate and gains computed in the previous section. The estimation error, denoted by $\check{x}[\cdot/\cdot]$ is hence:

$$\check{x}[k+l/k+l, k] = \tilde{x}[k+l/k+l] \\ -\check{K}_2(k/k+l)H_2(k)\tilde{x}[k/k] - \check{K}_2(k/k+l)v_2[k].$$

Following the steps of the previous section, the covariance of this estimation error can be evaluated to be:

$$\check{P}[k+l/k+l, k] = P[k+l/k+l] + \\ \check{K}_2(k/k+l)\left[H_2(k)P(k/k)H_2(k)^T + R_2(k)\right] \\ K_2(k/k+l)^T - \\ P(k+l, k/k+l, k)H_2(k)^T\check{K}_2(k/k+l)^T - \\ \check{K}_2(k/k+l)H_2(k)P(k+l, k/k+l, k)^T$$

where

$$P(k+l, k/k+l, k) = \mathcal{E}\tilde{x}[k+l/k+l]\tilde{x}[k/k]^T.$$

The corresponding optimal gain and estimation covariance are given by:

$$\check{K}_2(k/k+l) = P(k+l, k/k+l, k) \\ \cdot H_2(k)^T\left(H_2(k)P[k/k]H_2(k)^T + R_2(k)\right)^{-1}$$

and

$$\check{P}[k+l/k+l, k] = P[k+l/k+l] \\ -P(k+l, k/k+l, k)H_2(k)^T \\ \cdot\left(H_2(k)P[k/k]H_2(k)^T + R_2(k)\right)^{-1}H_2(k) \\ \cdot P(k+l, k/k+l, k)^T.$$

Compare these two equations with the expressions in the previous section. Following analogous steps as above, the covariance $P(k+l, k/k+l, k)$ can be evaluated recursively as:

$$P(k+l, k/k+l, k) = (I - K_1(k+l)H_1(k+l)) \\ \Phi(k+l, k+l-1)P(k+l-1, k/k+l-1, k)$$

with initial condition:

$$P(k, k/k, k) = P(k/k).$$

Notice that this recursion is exactly the same as the one in the section above. Indeed, the whole filtering scheme can be obtained by setting $N[\cdot/\cdot] = 0$. As a final remark, it can be shown that the improvement achieved by the filter in the previous section is directly proportional to the covariance reduction due to smoothing. Since the effect of smoothing is relatively small when the filter has reached steady-state, one would rather prefer to eliminate the smoothing stage and favor the simpler second option considered in the present section.

5. THE ALGORITHM

In this section we will provide the sketch of an algorithm for incorporating delayed measurements into the filtering scheme, as introduced in Section 3. From the observation above, an outline of the algorithm in Section 4 can be obtained by removing the lines corresponding to the implementation of the smoothing algorithm and setting $N = 0$. Suppose k is the validity time of a new measure z_2. Then:

(1) Set $l = 0$ and:

$$\hat{x}_s = \hat{x}[k/k] \quad D_x = P[k/k]\Phi(k+1,k)^T$$
$$P_s = P[k/k] \quad P_p = P[k/k]$$

(2) Set $l = l + 1$.
(3) Do one iteration of the Kalman Filter using the measurement $z_1[k+l]$.
(4) Do one iteration of the Kalman Smoother using the measurement $z_1[k+l]$:
(5) Update the covariance for the Delayed Filter:

$$P_p = (I - K_1(k+l)H_1(k+l)) \cdot\Phi(k+l,k+l-1)P_p$$

(6) When a new measurement $z_2[k]$ becomes available, then:
(7) Compute the optimal gain for incorporating delayed measurements and update the current estimate:

$$K_2(k/k+l) = P_p H_2(k)^T \left[H_2(k)P_s H_2(k)^T + R_2(k)\right]^{-1}$$
$$\hat{x}[k+l/k+l,k] = \hat{x}[k+l/k+l] + K_2(k/k+l)\left[z_2[k] - H_2(k)\hat{x}[k/k+l]\right]$$

(8) Update the state error covariance:

$$P[k+l/k+l,k] = P[k+l/k+l] - P_p \cdot H_2(k)^T\left[H_2(k)P_s H_2(k)^T + R_2(k)\right]^{-1} \cdot H_2(k)P_p^T$$

6. CONCLUSIONS

This note contains a complete recursive solution for a filtering problem involving measurements with different time-delays. We showed by using an example that the "intuitive" solution to the problem works only in special cases and showed a solution based on careful tracking of the estimates and their covariances. The solution is attractive since, as shown in the Appendix, it can be implemented using Bierman's UD factorization.

7. REFERENCES

Anderson, Brian D. O. and James Moore (1979). *Optimal Filtering*. Prentice Hall.

Bierman, G. J. (1977). *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York.

Gelb, Arthur (1974). *Applied Optimal Estimation*. The MIT Press.

Grewal, Mohinder S. and Angus P. Andrews (1993). *Kalman Filtering - Theory and Practice*. Prentice Hall.

Mendel, Jerry M. (1995). *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice Hall PTR, Englewood Cliffs, New Jersey.

Mirkin, Leonid, Hector Rotstein and Zalman Palmor (1999). $\mathcal{H}_\infty$ and $\mathcal{H}_2$ design of sampled-data systems using lifting-part i: General framework and solutions. *SIAM Journal on Control and Optimization* **38**(1), 175–196.

Appendix A. IMPLEMENTATION OF THE NEW FILTER

This appendix discusses the implementation of the filter presented in the note using a UD Kalman Filter. Recall that the main difficulty when implementing this filter is that measurements are valid for time t_T, the current measurement time is t_M, but numerous other measurements have been processed between t_T and t_M. As mentioned above, the update has the form:

$$P_T = P - P_p H^T \left(HP_0H^T + R\right)^{-1} HP_p^T$$

where:

P	State error covariance before the new update (at t_M^-)
P_T	State error covariance after the new update (at t_M^+)
P_p	Cross-covariance between the error state at t_T and at t_M
P_0	State error covariance at t_T
H	Measurement matrix
R	Measurement noise covariance matrix, assumed diagonal

Moreover, the cross-covariance matrix can be factorized as:

$$P_p = \hat{P}_p P_0 \quad \text{(A.1)}$$

Now suppose that P is given via its UD factorization (Bierman, 1977) $P = UDU^T$, where U is a unitary upper triangular matrix and D is a diagonal matrix with positive entries. We want to show how to compute the updated factors

$$P_T = U_T D_T U_T^T$$

without passing through the covariance calculations. To do this, one needs to prove that the measurements can be processed in a sequential manner. This can be done using an argument as the one in (Grewal and Andrews, 1993), but matrix manipulations seem to produce the result more directly. Start by writing:

$$H = \begin{bmatrix} h \\ H_2 \end{bmatrix}, \quad R = \begin{bmatrix} r & 0 \\ 0 & R_2 \end{bmatrix}$$

After lengthy calculations, one gets:

$$\begin{aligned} & P_T = P - P_p h^T \left(h^T P_0 h + r\right)^{-1} h P_p^T - \\ & P_p \left(h^T r^{-1} h P_0 + I\right)^{-1} H_2^T \\ & \left[H_2 \left(I + P_0 h^T r^{-1} h\right)^{-1} P_0 H_2^T + R_2\right]^{-1} \\ & H_2 \left(I + P h^T r^{-1} h\right)^{-1} P_p^T \end{aligned}$$

Using the formula for the inverse of a rank-one perturbation, write:

$$P_p^{(1)} = \left[I - h_1^T \left(r + h P_0 h^T\right)^{-1} h P_0\right] \quad \text{(A.2)}$$

Also:

$$P_0^{(1)} = P_0 - P_0 h^T \left(h P_0 h^T + r\right)^{-1} h P_0$$

Notice that using the definition Eq.A.1 in A.2, we can write $P_p^{(1)} = \hat{P}_p P_0^{(1)}$. An algorithm for processing the measurements sequentially can be given as follows:

(1) Set $P^{(0)} = P$, $P_0^{(0)} = P_0$, $P_p^{(0)} = P_p$.
(2) For $i = 1, \cdots, m$, where m is the number of measurements, set:
   $h \leftarrow H^{(i)}$ (measurement vector corresponding to the i-th measurement)
   $r \leftarrow R(i,i)$ (noise covariance of the i-th measurement)
   and compute

$$\begin{aligned} P^{(i)} &= P^{(i-1)} - \\ & P_p^{(i-1)} h^T \left(h P_0^{(i-1)} h^T + r\right)^{-1} h P_p^{(i-1)T} \\ P_0^{(i)} &= P_0^{(i-1)} - \\ & P_0^{(i-1)} h^T \left(h P_0^{(i-1)} h^T + r\right)^{-1} h P_0^{(i-1)} \\ P_p^{(i)} &= P_p^{(i-1)} \left[I - h_1^T \left(r + h P_0^{(i-1)} h^T\right)^{-1} h P_0\right] \end{aligned}$$

   Alternatively $P_p^{(i)} = P_p P_0^{(i)}$.
(3) Set $P_T = P^{(m)}$.

When iterating over the measurements, one needs to implement the two recursions above in an efficient manner, compatible with the UD factorization. The recursion for $P_0^{(i)}$ is the case considered by the Bierman algorithm and hence, assuming that $P_0 = U_0 D_0 U_0^T$ that algorithm can be used to update the factors. On the other hand, the iteration for $P^{(i)}$ falls within the general case of degree-one perturbations to a UD factorization. The algorithm for degree-one perturbations can be described as follows. Suppose we write:

$$W E W^T = U D U^T + \alpha v v^T$$

where U is a unitary upper triangular matrix, D is a positive diagonal matrix, α is a scalar and v is a column vector of compatible dimension. The objective is to compute the unitary upper triangular matrix W and the positive diagonal matrix E. In order to see how this can be done, write:

$$W = \begin{bmatrix} W_1 & w_2 \\ 0 & 1 \end{bmatrix}, \ E = \begin{bmatrix} E_1 & 0 \\ 0 & e_2 \end{bmatrix}, \quad U = \begin{bmatrix} U_1 & u_2 \\ 0 & 1 \end{bmatrix}, \ D = \begin{bmatrix} D_1 & 0 \\ 0 & d_2 \end{bmatrix}$$

and $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$. Then:

$$\begin{bmatrix} W_1 E_1 W_1^T + e_2 w_2 w_2^T & e_2 w_2 \\ e_2 w_2^T & e_2 \end{bmatrix} = \begin{bmatrix} U_1 D_1 U_1^T + d_2 u_2 u_2^T + \alpha v_1 v_1^T & d_2 u_2 + \alpha v_2 v_1 \\ d_2 u_2^T + \alpha v_2 v_1^T & d_2 + \alpha v_2^2 \end{bmatrix}.$$

From this equation:

$$e_2 = d_2 + \alpha v_2 \quad \text{(A.3)}$$

$$w_2 = \frac{d_2 u_2 + \alpha v_2 v_1}{e_2}. \quad \text{(A.4)}$$

Replacing these values in the 1–1 block above:

$$W_1 E_1 W_1^T + \frac{\left(d_2 u_2 + \alpha v_2 v_1\right)\left(d_2 u_2 + \alpha v_2 v_1\right)^T}{d_2 + \alpha v_2^T} = U_1 D_1 U_1^T + d_2 u_2 u_2^T + \alpha v_1 v_1^T.$$

After some simple manipulations, this expression can be simplified to:

$$W_1 E_1 W_1^T = U_1 D_1 U_1^T + \frac{d_2 \alpha}{d_2 + \alpha v_2^2} \cdot \left(v_1 - v_2 u_2\right)\left(v_1 - v_2 u_2\right)^T$$

This equation has exactly the same form as the original one of the problem, except that now the size of the matrices is reduced by one. The following algorithm can then construct the updated UD factorization directly from the original ones and the degree-one perturbation.

(1) Compute e_2 and w_2 using A.3 and A.4.
(2) Re-assign:

$$\alpha \leftarrow \frac{d_2 \alpha}{d_2 + \alpha v_2^2}, \quad v \leftarrow v_1 - v_2 u$$

(3) Repeat step 1 until v reduces to a scalar.

This algorithm can now be used to update the recursion for $P^{(i)}$ by making the appropriate identifications.