Kalman Filtering, Factor Graphs and Electrical Networks

Pascal O. Vontobel, Daniel Lippuner, and Hans-Andrea Loeliger
ISI-ITET, ETH Zurich, CH-8092 Zurich, Switzerland.

Abstract

Factor graphs are graphical models with origins in coding theory. It is well known that Kalman filtering is an instance of the generic sum(mary)-product algorithm on the corresponding factor graph. In this paper, a one-to-one correspondence between such factor graphs and a class of electrical networks is presented. The electrical network "computes" correct Bayesian estimates even for factor graphs with cycles.

Extended Summary

Background

Factor graphs [1] are a unifying framework for a wide variety of system models in coding, signal processing, and artificial intelligence. Many important algorithms in these fields are instances of the generic sum(mary)-product algorithm, which works by message passing in the factor graph. One such instance is Kalman filtering and smoothing; for the scalar case, this was shown in [1], the vector case was discussed in [2].

In the present paper, as in [2], we will use Forney-style factor graphs (FFGs) [3], a variation of factor graphs (in [3] called "normal graphs") with a number of attractive properties. Such a graph expresses the factorization of a function of several variables. E.g., let X, Y, and Z be random variables that form a Markov chain. Then their joint probability density $p_{XYZ}(x, y, z)$ can be written as

$$p_{XYZ}(x, y, z) = p_X(x)\, p_{Y|X}(y|x)\, p_{Z|Y}(z|y). \tag{1}$$

This factorization is expressed by the graph of Fig. 1. The edges in the graph correspond to the variables in (1), the nodes in the graph correspond to the factors in (1), and an edge is connected to some node if and only if the variable corresponding to the edge is an argument in the factor corresponding to the node.

One of the attractive properties of Forney-style factor graphs is that they subsume block diagrams as in Figures 2 and 3. These figures illustrate the classical linear state space model

$$X[k] = AX[k-1] + BU[k] \tag{2}$$

$$Y[k] = CX[k] + W[k] \tag{3}$$

where for $k \in \mathbb{Z}$, $U[k]$, $W[k]$, $X[k]$, and $Y[k]$ are real vectors and where A, B, and C are matrices of appropriate dimensions. For the sake of exposition we will, in this paper, focus on the case where all these variables and matrices are scalars. The nodes in Figures 2 and 3 are then defined to represent factors as shown in Table 1 (left column). In Kalman filtering, both $U[.]$ and $W[.]$ are usually assumed to be white Gaussian ("noise") processes; the corresponding nodes in these figures are then of the type shown in the top row of Table 1. The overall function (the factorization of) which is expressed by Figures 2 and 3 is the joint probability density of all involved variables.

If the variables $Y[k]$ are observed, say $Y[k] = y_k$, the corresponding branches are "terminated" by nodes corresponding to factors $\delta(y[k] - y_k)$ (the dashed boxes in Fig. 2); such factors may be viewed as degenerate cases of Gaussian distributions (top row of Table 1) with variance zero. The graph then represents (the factorization of) the joint *a posteriori* density of all involved variables.

As described in [1] and [2], Kalman filtering may be viewed as passing messages along the edges of these graphs; in general, the messages consist of mean vectors and covariance matrices (or inverses of covariance matrices) and are computed according to the rules of the generic sum-product algorithm. In its most narrow sense, Kalman filtering is only the forward sum-product recursion through the graph of Fig. 2 and yields the a posteriori probability distribution of the state $X[k]$ given the observation sequence $Y[.]$ up to time k. By computing also the backwards messages, the a posteriori probability of all quantities given the whole observation sequence $Y[.]$ may be obtained.

More generally, Kalman filtering amounts to the sum-product algorithm on any FFG (or part of an FFG) that consists of (the vector/matrix versions of) the linear building blocks listed in Table 1 (left column). As pointed out in [2], Kalman filtering in this sense is equivalent to a general least-squares algorithm.

New Results

Interestingly, the results of Kalman filtering (in its widest sense as defined above) can also be obtained by translating the state space model into an electrical network according to the rules of Table 1. These networks consist of resistors, voltage sources, and transformers. (The latter are defined by the equations $U_2 = aU_1$ and $I_2 = I_1/a$, where U_i and I_i are the voltage accross, and the current into, terminal pair i, $i = 1, 2$.) As shown in Table 1, each edge/variable of the factor graph corresponds to a pair of nodes in the electrical network.

Theorem. The voltages between such node pairs are the Bayesian estimates (i.e., the means of the *a posteriori* marginal distributions) of the corresponding random variables.

The equivalence of (traditional) Kalman filtering with an electrical network as in Table 1 was recognized already by Carter [4]. However, the above theorem holds also if the factor graph has cycles, which goes beyond [4] (and appears to be related to the convergence results of [6] and [7]).

In the electrical network, message passing according to the sum-product algorithm (and thus classical Kalman filtering) can be interpreted as follows. Each message represents a

Gaussian distribution as in the top row of Table 1, and can thus be represented by a voltage source (corresponding to the mean) in series with a resistor (corresponding to the variance). As illustrated in Table 2, the computation of messages out of the other node types in Table 1 then amounts to connecting the incoming messages/circuits to the node circuit and reducing the resulting circuit to its Thévenin equivalent circuit (i.e., a voltage source in series with a resistor).

We conclude with the following remarks:

- The correspondence given in Table 1 can be extended to the vector case.

- There is a dual version of the correspondence of Table 1 where the voltages and the currents are interchanged, as are series and parallel connections of subcircuits.

- The representation of Kalman filtering (and smoothing) as an electrical network allows an easy transition from discrete-time Kalman filtering to continuous-time Kalman filtering.

- A generalization beyond Kalman filtering / least squares was recently discovered by the first author [8]: if $p(x, y, \ldots)$ is any joint probability density function such that $-\log p(x, y, \ldots)$ is convex, then a factor graph of p can be translated into an electrical network that finds the global maximum of p. The network consists of voltage sources (or current sources) and *nonlinear* resistors. Circuit simplifications similar to those in Table 2 correspond to the message passing rules of the *max-product* version (rather than the sum-product version) of generic sum(mary)-product algorithm. Moreover, if the voltages are solutions to the primal maximization problem, then the currents are the solutions to the dual problem.

Acknowledgement

The authors thank S. Mitter for pointing out the book by Dennis [5], which inspired this work.

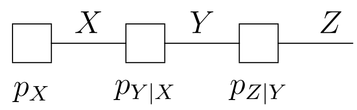


Figure 1: Forney-style factor graph (FFG) of a Markov chain.

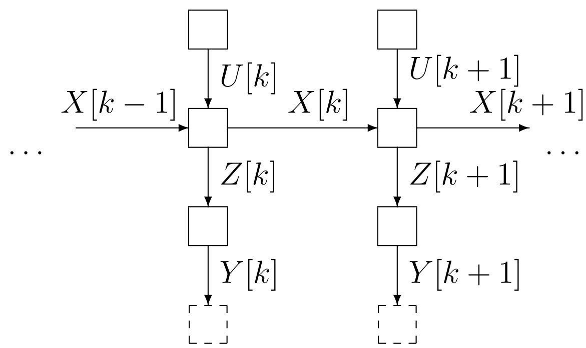


Figure 2: State space model.

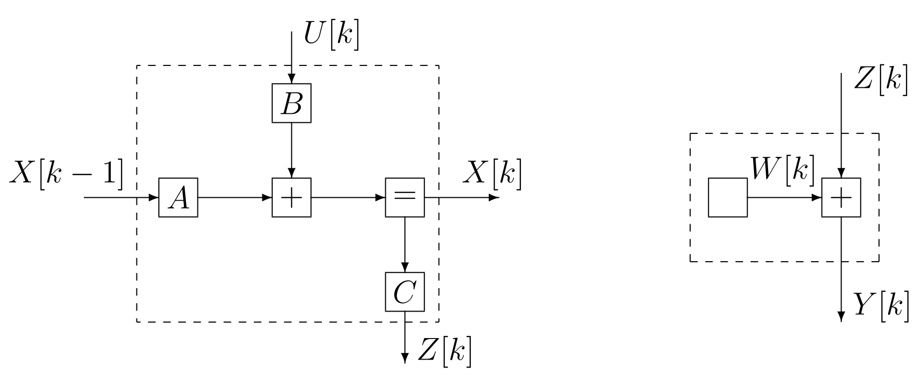


Figure 3: Details of linear state space model.

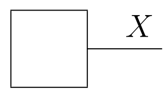
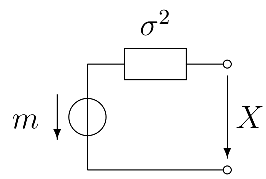
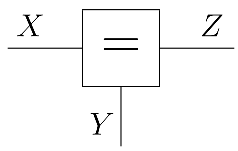
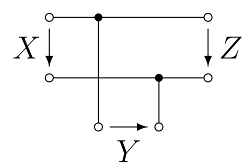
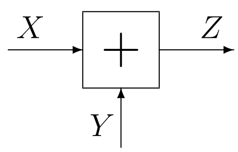
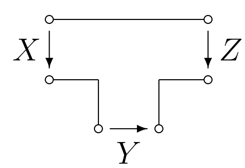
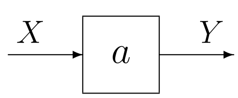
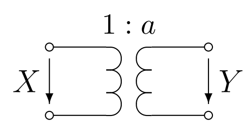
0	 $\gamma e^{-\frac{(x-m)^2}{2\sigma^2}}$	
1	 $\delta(x-y)\,\delta(x-z)$	
2	 $\delta(x+y-z)$	
3	 $\delta(y-ax)$	

Table 1: Correspondence between the nodes in the factor graph (left) and the components of the electrical network (right). All variables are scalars.

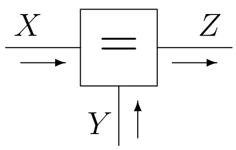
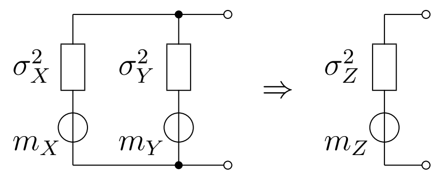
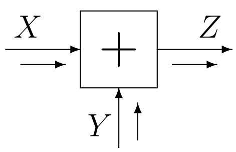
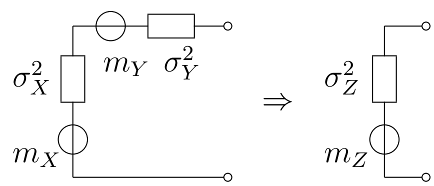
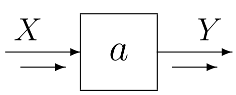
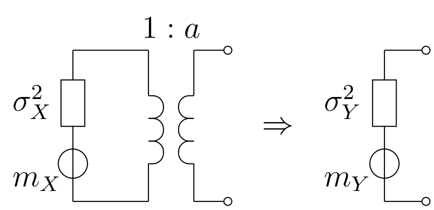
<table>
<tr><td>1</td><td></td><td>$m_Z = \frac{m_X/\sigma_X^2 + m_Y/\sigma_Y^2}{1/\sigma_X^2 + 1/\sigma_Y^2}$
$\sigma_Z^2 = \frac{1}{1/\sigma_X^2 + 1/\sigma_Y^2}$</td><td></td></tr>
<tr><td>2</td><td></td><td>$m_Z = m_X + m_Y$
$\sigma_Z^2 = \sigma_X^2 + \sigma_Y^2$</td><td></td></tr>
<tr><td>3</td><td></td><td>$m_Y = am_X$
$\sigma_Y^2 = a^2\sigma_X^2$</td><td></td></tr>
</table>

Table 2: Computation of messages (left and middle) with circuit interpretation (right).

References

[1] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. IT–47, no. 2, pp. 498–519, 2001.

[2] H.-A. Loeliger, "Least squares and Kalman filtering on Forney graphs," to appear in *Codes, Graphs, and Systems,* festschrift in honour of G. D. Forney, Kluwer, 2001.

[3] G. D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. Inform. Theory,* vol. IT-47, pp. 520–548, Feb. 2001.

[4] D. W. Carter, "A circuit theory of the Kalman filter," in *Proc. 32nd Conf. on Decision and Control* (San Antonio, TX, USA), pp. 1224–1226, Dec. 1993.

[5] J. B. Dennis, *Mathematical Programming and Electrical Networks.* MIT Press, Cambridge, Mass., 1959.

[6] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. Inform. Theory,* vol. 47, pp. 736–744, Feb. 2001.

[7] P. Rusmevichientong and B. Van Roy, "An analysis of belief propagation on the turbo decoding graph with Gaussian densities," *IEEE Trans. Inform. Theory,* vol. 47, pp. 745–765, Feb. 2001.

[8] P. O. Vontobel, *Kalman Filters, Factor Graphs, and Electrical Networks.* Internal report INT/200202, ISI-ITET, ETH Zurich, April 2002.