Neural Networks 21 (2008) 1328–1343

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# Neural network learning of optimal Kalman prediction and control

Ralph Linsker *

*IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA*

ARTICLE INFO

*Article history:*
Received 8 November 2007
Accepted 21 May 2008

*Keywords:*
Kalman filter
Kalman control
Recurrent neural network
Local cortical circuit

ABSTRACT

Although there are many neural network (NN) algorithms for prediction and for control, and although methods for optimal estimation (including filtering and prediction) and for optimal control in linear systems were provided by Kalman in 1960 (with nonlinear extensions since then), there has been, to my knowledge, no NN algorithm that learns either Kalman prediction or Kalman control (apart from the special case of stationary control). Here we show how optimal Kalman prediction and control (KPC), as well as system identification, can be learned and executed by a recurrent neural network composed of linear-response nodes, using as input only a stream of noisy measurement data.

The requirements of KPC appear to impose significant constraints on the allowed NN circuitry and signal flows. The NN architecture implied by these constraints bears certain resemblances to the local-circuit architecture of mammalian cerebral cortex. We discuss these resemblances, as well as caveats that limit our current ability to draw inferences for biological function. It has been suggested that the local cortical circuit (LCC) architecture may perform core functions (as yet unknown) that underlie sensory, motor, and other cortical processing. It is reasonable to conjecture that such functions may include prediction, the estimation or inference of missing or noisy sensory data, and the goal-driven generation of control signals. The resemblances found between the KPC NN architecture and that of the LCC are consistent with this conjecture.



## 1. Introduction

The Kalman optimal filter and controller (Kalman, 1960) are classical solutions for efficient optimal estimation (which includes prediction, filtering, and smoothing) and optimal control in linear systems. They also form the basis for extensions that yield approximately optimal solutions for certain types of nonlinear systems. Within the field of neural networks, a great many algorithms for prediction and control in a variety of settings have been developed. Yet there exists, to my knowledge, no neural algorithm for learning the optimal Kalman filter (KF), nor for learning the optimal Kalman controller (KC) except in the stationary case (discussed in Section 5.2). In addition, the classical Kalman algorithms assume that the parameters characterizing the external system (the 'plant') and the measurement process are known in advance. When they are not known, a separate process of system identification is typically performed.

In this paper we derive a neural network (NN) circuit and algorithm that learns and executes Kalman estimation and control, and that also determines the required combinations of plant and measurement process parameters, using as input *only* a stream of noisy measurement data vectors (and, for the controller, the specification of the cost function whose value is to be optimized). The differences between the Kalman filter and controller learned by the network, and those derived using the classical Kalman algorithms, can be made arbitrarily small, provided that certain expectation values over distributions are sufficiently well approximated by the corresponding finite-sample statistics[1] (as discussed in Sections 3 and 4).

The resulting artificial neural circuit and algorithm may prove useful for implementing the learning and execution of Kalman prediction and control, and its nonlinear extensions, in parallel systems consisting of simple processors.

The resulting circuit architecture also has distinctive features that invite comparison with aspects of biological neural networks, particularly in cerebral cortex, and may help in exploring the possible functions of such networks.

The paper is organized as follows. Section 2 summarizes the optimal linear estimation and control problems, and the classical Kalman filter and controller algorithms. In Section 3 we derive a neural network algorithm that both solves the system identification problem – i.e. learns the dynamical properties of

* Corresponding address: IBM T.J. Watson Research Center, Rm. 36-110, 1101 Kitchawan Road & Route 134, P.O. Box 218, Yorktown Heights, NY 10598, USA. Tel.: +1 914 945 1077; fax: +1 914 945 2141.
*E-mail address:* linsker@us.ibm.com.

[1] We use 'sample' in its statistical sense, to mean a subset of a population (ensemble) that is defined by a distribution.

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.05.002

the plant (as these properties are reflected in the measurement data) – and learns the optimal Kalman filter with arbitrary accuracy. The derivation proceeds in several stages. We first transform the classical Kalman estimation equations into a form that explicitly involves only the input measurement vectors, and not the plant state vectors themselves. (We do this because the plant state is unknowable from the data in principle, even apart from noise, when, as we assume, the transformation from plant state to measurement vector is not specified to the NN.) We then derive a learning procedure that exactly implements these transformed equations, but that is expressed in terms of certain expectation values. Next, we derive a neural network algorithm that implements this learning procedure with arbitrary accuracy (depending upon how well the expectation values are approximated by sample statistics). Some distinctive elements of this algorithm include: (a) the use of local neural network methods to perform the required learning or use of the inverse of an error covariance matrix; (b) the generation of this covariance matrix by using either an ensemble of measurement vectors at each time step (e.g., the positions of a set of tracked features in a visual scene), a single vector tracked over time to generate such an ensemble, or a combination of the two; (c) the simultaneous learning of the Kalman filter, use of that filter to predict the future plant state, and refinement of the learning of the plant dynamical parameters; and (d) a specific recurrent circuit architecture, and sequencing of computations, that are implied by the algorithm. Finally, the joint NN learning of the plant dynamics and the Kalman filter is illustrated with a numerical example.

In Section 4, we derive a neural algorithm for Kalman control. The several stages of the derivation are similar to those used for Kalman estimation. There are evident similarities that result from the mathematical duality (Kalman, 1960) between Kalman's optimal estimation and optimal control solutions, but there is an additional distinctive feature: Kalman's duality includes a time reversal operation, so that the Kalman control matrix is computed by a process that operates 'backward in time', from the future time of target (goal) acquisition to the present. We show how this requirement is implemented within the neural algorithm, which handles a decrementing time index during the learning process, and generates a sequence of controller outputs to the plant in physical (forward-moving) time. We then integrate the control method into the same neural circuit and algorithm that handles estimation and system identification.

In Section 5 we discuss several issues. First, we identify ways in which the computational task – Kalman prediction and control – places constraints on the type of NN circuitry and signal flows that are involved in performing that task. Second, we comment on applications to artificial NN designs, and discuss prior work that has used NNs in conjunction with Kalman methods.

Finally, in Section 5.3 and the speculative Section 5.4, we identify certain resemblances between the artificial NN that we are led to by the Kalman prediction and control (KPC) constraints, and the architecture (and proposed signal flows) of the putative 'local cortical circuit' (LCC, minicolumn, canonical microcircuit) of mammalian cerebral cortex. The resemblances between the KPC NN and the LCC, and important caveats that apply to the interpretation of these resemblances, are discussed.

Section 6 summarizes and concludes the paper.

## 2. Classical Kalman linear estimation and control

In classical linear estimation and control (Kalman, 1960) an external system (the 'plant') is described by a state vector $x_t$ (e.g., a point's trajectory) at each discrete time $t$, and the dynamical rule

$$x_{t+1} = Fx_t + Bu_t + m_t, \tag{1}$$

where $m_t$ is plant noise (e.g., random buffeting of an object) having zero mean and covariance $Q$, and the optional vector $u_t$ is an external driving term and/or a computed control term. Each measurement vector $y_t$ satisfies

$$y_t = Hx_t + n_t, \tag{2}$$

where $n_t$ is measurement noise having zero mean and covariance $R$. The matrices $F$, $B$, $H$, $Q$, and $R$, and the vector $u_t$, are assumed known. (Continuous-time versions of these problems and their Kalman solutions have been formulated, but we will limit ourselves to the discrete-time case for simplicity.)

### 2.1. Classical Kalman estimation (filtering and prediction)

Given measurements through time $t$, the goal of optimal filtering (or, respectively, one-step-ahead prediction) is to compute a posterior state estimate $\hat{x}_t$ (resp., a prior state estimate $\hat{x}^-_{t+1}$) that minimizes the generalized mean-square estimation error[2] $E[(\xi_t)'C\xi_t]$ (resp., $E[(\xi^-_{t+1})'C\xi^-_{t+1}]$) where $\xi_t \equiv x_t - \hat{x}_t$, $\xi^-_{t+1} \equiv x_{t+1} - \hat{x}^-_{t+1}$, and $C$ is a symmetric positive-definite matrix. Throughout this paper, a variable having a 'hat' will generally denote an estimate of the underlying variable, and a variable having a tilde will denote the result of applying a transformation to the underlying variable.

Kalman (1960) showed that, under a variety of conditions, the optimal estimation solution for both filtering and prediction is given by what we will refer to as the 'execution' equations

$$\hat{x}_t = \hat{x}^-_t + K_t(y_t - H\hat{x}^-_t); \qquad \hat{x}^-_{t+1} = F\hat{x}_t + Bu_t; \tag{3}$$

and the 'learning' equations

$$K_t = P^-_t H'(HP^-_t H' + R)^{-1}; \qquad P^-_{t+1} = F(I - K_t H)P^-_t F' + Q. \tag{4}$$

(These solutions are independent of $C$.) Eq. (4) are initialized by assuming some distribution of values for $\xi^-_0$ and setting $P^-_0 \equiv E[\xi^-_0(\xi^-_0)']$. It then follows (Kalman, 1960) that, for all $t$, $P^-_t = E[\xi^-_t(\xi^-_t)']$. Thus the KF matrix, $K_t$, is learned iteratively using Eqs. (4), starting with an arbitrary matrix and converging exponentially rapidly to its final value as each new measurement is obtained. The classical KF learning algorithm involves multiplications of one matrix by another, and matrix inversion.

The model prediction $\hat{x}^-_t$ and the current measurement $y_t$ are optimally blended (to minimize the estimation error) by using the KF (Eq. (3)). As expected intuitively, when the plant noise is much greater than the measurement noise, this blending gives greater weight to the current measurement; when the measurement noise is much greater, the model prediction receives greater weight.

### 2.2. Classical Kalman control

The classical control problem known as 'linear quadratic regulation' can be defined as follows. A controller is required to generate a set of signals $\{u_t\}$ that minimizes the expected total cost $J$ of approaching a desired target state at time $N$. Here $J$ reflects the cost of producing each control output (e.g., the energetic cost of moving a limb or firing a rocket thruster) plus a penalty that is a function of the difference between the actual state at each time step and the target state. Specifically,

$$J = E\left[\sum_{t=t_0}^{N-1}(u'_t g u_t + x'_t r x_t) + x'_N r x_N\right], \tag{5}$$

where $g$ and $r$ are specified symmetric positive-definite matrices. (We take the target state to be $x = 0$ for simplicity.)

[2] Notation: $E[\ldots]$ denotes expectation value, prime denotes transpose, and $I$ is the identity matrix.