ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 391-396

A Divide-and-Conquer Learning Architecture for Predicting Unknown Motion

P. Wira, J.-P. Urban, J. Gresser

Trop Research Group, University of Mulhouse
4 rue des Frères Lumière, 68093 Mulhouse, France
e-mail : p.wira@univ-mulhouse.fr

Abstract. Time varying environments or model selection problems lead to crucial dilemmas in identification and control science. In this paper, we propose a modular prediction scheme consisting in a mixture of expert architecture. Several Kalman filters are forced to adapt their dynamics and parameters to different parts of the whole dynamics of the system. The performances of this modular learning scheme are evaluated on a visual servoing problem: motion prediction of an object in a 3-D space for pursuing it with a 3 degree-of-freedom robot manipulator.

1 Introduction

If positioning is a basic task in robotic, tracking a moving target is a more sophisticated one. Indeed, target tracking implies to include the system's dynamics, and to have some information available about the target's movements.

Dynamics can be taken into account with a structure like the Kalman Filter (KF). This structure however suffer from the fact that the system's model has to be perfectly known. To overcome this limitation, a state model Adaptive Kalman Filter (AKF) is used. Thus, without having access to the system and noise models, this filter is able to approximate its dynamics around the current state.

In this paper, we propose to integrate different noise models rather than designing a controller for a specific system behavior. This is done by the divide-and-conquer principle of the *mixture of experts*. This modular architecture mediates the output of several expert networks and learns their performances as a function of the system's behavior. The expert networks, by way of state model Adaptive Kalman Filters with different noise properties, will be assigned to the whole system's workspace.

As an illustration, the proposed architecture is used to predict the movements of an object moving in a 3-D space. The predictions serve to visually control a three-degree-of-freedom redundant robot manipulator with a stereoscopic visual sensor. The target's movements (and the end-effector's movements) are thus anticipated and these predicted values are used to compute the motor responses for the robot. In practice, this prediction step also allows to compensate the incompressible time-delay introduced by the image acquisition. Without any *a priori* knowledge about the target's movements, and without the robot models and the scene configuration, the robot is able to track moving targets accurately.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 391-396

2 Neuro-Control Principle

A visual-feedback control loop allows a robotic system to interact with the scene in which it evolves. Computer vision providing closed-loop position control for a robot end-effector is referred to as visual servoing [1]. This section describes the relationships on which feedback control is based, and introduces how a simple neural network structure is able to visually servo a robot manipulator in an efficient and robust manner.

Consider $\mathcal{V} \in \mathbb{R}^3$ the image feature parameter space, and $\mathbf{v}_k \in \mathcal{V}$ and $\mathbf{d}_k \in \mathcal{V}$, respectively the image coordinate vector of the end-effector and of the visual target, with *k*, the time index. Two cameras, voluntary placed in a same plane, measure the visual error vector. This error is determined in the image space by

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{v}_k \,. \tag{1}$$

Consider $\mathbf{f}$, the forward kinematic transformation between joint and image position spaces:

$$\mathbf{f} : \mathcal{Q} \rightarrow \mathcal{V},\ \mathbf{v}_k = \mathbf{f}(\mathbf{q}_k)\,, \tag{2}$$

where $\mathcal{Q} \in \mathbb{R}^q$ is the robot joint space of dimension *q* and $\mathbf{q}_k$ the joint angles vector. Feedback control methods are generally based on a linear approximation in the neighborhood of the working point. Thus, if $\partial\mathbf{q}_k$ and $\partial\mathbf{v}_k$ are small displacements measured respectively in joint and image feature spaces at instant *k*, a local linear Jacobian approximation allows determination of the motor response $\Delta\mathbf{q}_k$ in terms of an incremental position variation:

$$\Delta\mathbf{q}_k = \mathbf{B}_k.\mathbf{e}_k \,. \tag{3}$$

We choose to approximate the inverse image Jacobian $\mathbf{B}_k = \mathbf{J}_\mathbf{f}^{-1}(\mathbf{q}_k)$ with a supervised Self-Organizing Maps (SOM) neural network as proposed and fully described in [2]. This iterative stochastic adaptation of $\mathbf{B}_k$ realizes *de facto* a linear local approximation, and is used to control the robot arm. Its performances have been emphasized in several robot positioning tasks and learning sensory-motor coordination.

3 State Model Adaptive Kalman Filter

To describe the discrete-time Kalman filtering algorithm, a stochastic state-space system can be represented by

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{D}\mathbf{w}_k \,, \tag{4}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \,, \tag{5}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the system state vector and $\mathbf{z}_k \in \mathbb{R}^p$ the measurement. Vectors $\mathbf{w}_k \in \mathbb{R}^s$ and $\mathbf{v}_k \in \mathbb{R}^p$ are respectively the disturbance noise and measurement noise, both are assumed to be zero mean and Gaussian noises with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$. If the two noise sequences are stationary and mutually independent, and mutually independent of $\mathbf{x}_0$, the Kalman algorithm recursively calculates the optimal estimation $\hat{\mathbf{x}}_{k+1|k}$.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 391-396

Kalman filtering requires an exact knowledge of the model parameters for optimal behavior. Practically, one often has only access to a simplified model, to an approximated model or to a model valid in a restricted workspace. This is true for the system model (state representation) and for the noises properties (the two first-order moments of the stochastic processes).

To overcome these limitations, the filter's parameters can be estimated online while filtering. These filters are called adaptive and are commonly limited to the estimation of $\mathbf{Q}_k$ and $\mathbf{R}_k$ [4]. They also assume all the other parameters to be known, in particular the system's dynamical model. They suppose optimality, thus if the innovation sequence is no more a white Gaussian sequence, $\mathbf{Q}_k$ and $\mathbf{R}_k$ are modified in consequence.

To cope with the unavailability of the system's dynamical model, we have proposed and evaluated an alternative approach, a state model adaptive Kalman filter [5]. This adaptive filter works without the explicit system's dynamical model. Its system matrix is continuously adjusted and estimated around the current state. If the process noise is centered ($E[\mathbf{w}_k]=0$) and if the process is only controlled by the stochastic parts, the system matrix can be expressed by

$$\hat{\mathbf{A}}_k = \mathbf{x}_k \mathbf{x}_{k-1}^{-1} = \mathbf{x}_k \left(\mathbf{x}_{k-1}^T \mathbf{x}_{k-1}\right)^{-1} \mathbf{x}_{k-1}^T . \quad (6)$$

Assuming the measurement sequence $\mathbf{v}_k$ to be centered ($E[\mathbf{v}_k]=0$), the state at iteration k can be expressed from (5): $\mathbf{x}_k = \hat{\mathbf{C}}_k^+ \mathbf{z}_k$.

These two last expressions lead to the adaptation of the state model:

$$\hat{\mathbf{A}}_k = \mathbf{C}_k^T \mathbf{G}_k^{-1} \mathbf{z}_k (\mathbf{z}_{k-1}^T \mathbf{G}_{k-1}^{-T} \mathbf{G}_{k-1} \mathbf{G}_{k-1}^{-T} \mathbf{z}_{k-1})^{-1} \mathbf{z}_{k-1}^T \mathbf{C}_k^T \mathbf{G}_{k-1}^{-T} \mathbf{C}_{k-1} , \quad (7)$$

if $p < n$, and where $\mathbf{G}_k = \mathbf{C}_k \mathbf{C}_k^T \in \mathbb{R}^{p*p}$ and $\mathbf{G}_{k-1}^{-T} = (\mathbf{G}_{k-1}^T)^{-1} = (\mathbf{G}_{k-1}^{-1})^T$.

This approach is stable, robust, and works best for representative noise statistics.

To address more complex problems, with time-varying or various noise statistics, we propose to use a modular architecture composed of AKFs with different noise models.

4 Prediction with a Modular Architecture

The mixture of experts was firstly proposed by [3]. This modular architecture, depicted in Figure 1, allows to switch between different neural networks, called expert networks. A gating network learns their performances and mediates their outputs.

At iteration k, the output of the mixture is expressed by the weighted sum of the outputs of the K expert networks, $\mathbf{y}_k = \sum_{i=1}^{K} g_{i,k} \mathbf{y}_{i,k}$. The output $\mathbf{y}_{i,k}$ of the ith expert network is mediated with an activation function elaborated by the gating network. This function is referred to as *softmax* and is defined by

$$g_{i,k} = \left(e^{\mathbf{u}_{i,k}/T}\right)\left(\sum_{j=1}^{K} e^{\mathbf{u}_{j,k}/T}\right)^{-1} , \quad (8)$$

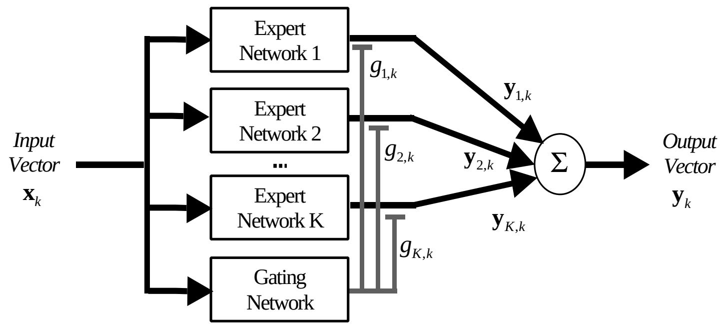

Figure 1. The mixture of experts framework

where T denotes a "temperature parameter" and $\mathbf{u}_{i,k} = \mathbf{x}_k^T \mathbf{a}_{i,k}$ is a function of the inputs $\mathbf{x}_k$. The values of the synaptic weight vector $\mathbf{a}_{i,k}$ of the *i*th output neuron in the gating network at iteration *k+1* are adapted according to

$$\mathbf{a}_{i,k+1} = \mathbf{a}_{i,k} + h_k \left(h_{i,k} - g_{i,k} \right) \mathbf{x}_k \ . \tag{9}$$

The output of the *i*th expert network is associated with a conditional *a posteriori* probability, iteratively adapted:

$$h_{i,k} = \left(\frac{g_{i,k}}{s_i} \mathrm{e}^{-\frac{1}{2s_i^2}\left\| \mathbf{d}_k - \mathbf{y}_{i,k-1} \right\|^2} \right) \left(\sum_{j=1}^{K} \frac{g_{i,k}}{s_j} \mathrm{e}^{-\frac{1}{2s_j^2}\left\| \mathbf{d}_k - \mathbf{y}_{i,k-1} \right\|^2} \right)^{-1} , \tag{10}$$

where $\mathbf{d}_k$ is the desired response and s_i denotes a scaling parameter associated with the *i*th expert network.

For the prediction task, state model AKFs with different parameters are used as expert networks. The original algorithm of the mixture of experts is then modified, firstly by the adaptation of the diagonal covariance matrices $\mathbf{Q}_{i,k}$ and $\mathbf{R}_{i,k}$ of the *i*th AKF. The strategy of adapting these estimated covariances is implemented by

$$\mathbf{Q}_{i,k+1} = \mathbf{Q}_{i,k}(1 - b h_k h_{i,k}) \text{ , with } b > 0 \text{ ,} \tag{11}$$

$$\mathbf{R}_{i,k+1} = \mathbf{R}_{i,k}(1 - b' h_k h_{i,k}) \text{ , with } b' > 0 \text{ .} \tag{12}$$

Secondly, and in order to improve the learning performance of the gating network, an adaptive learning rate scheme was implemented: $h_k = h(1 - e^{-ae})$. a is a positive constant and h is a exponentially decreasing function of time converging to a minimum positive value. This rate is not only a function of time but also of an error $e = \sum_{n=1}^{N} (\mathbf{d}_k(n) - \mathbf{y}_{i,k-1}(n))^2$. Vectors $\mathbf{d}_k$ and $\mathbf{y}_{i,k}$ now represents respectively the last measurement and the last prediction. The objective of this learning scheme is to increase the learning rate of the current expert network. Each AKF, by been specialized in only a part of the system's workspace, will thus improve its prediction performances.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 391-396

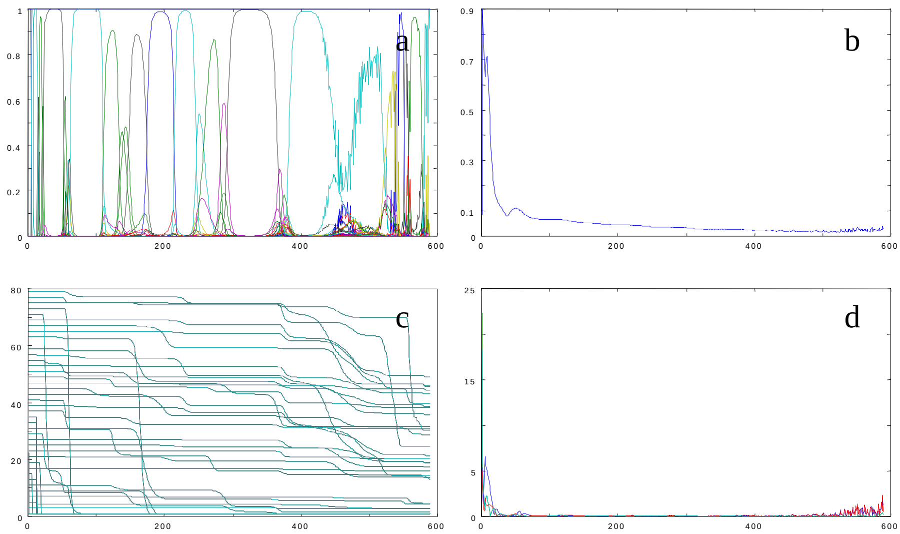

Figure 2. Evolution of the different parameters. a) the AKF weightings, b) the adaptive learning rate, c) the $\mathbf{Q}_{i,k}$ diagonal elements and, d) the absolute error prediction (pixels).

5 Predictive Robot Control

Various experiments have been conducted to compare the predictive neuro-controller architectures. Some experiments have been implemented on our robotic platform. Other experiments use a simulation of the robotic setup to test the various system parameters in a more exhaustive manner.

In a simulated application, 40 AKFs with different parameters and initializations are placed in the mixture of experts architecture. The gating network learns the performances of the experts networks on an accelerating curved trajectory which ends with little random movements in a restricted area. The speed of the target varies from 0.01 to 1.5 m/s. Figure 2 a) shows that there is often only one AKF that is the most able to do the predictions. The selected AKFs are then reinforced to predict specific kinds of trajectory. Figure 2 also shows the evolution of the adaptive learning rate h_k, the absolute error prediction of the modular architecture and the adaptation strategy of the covariances $\mathbf{Q}_{i,k}$. The results are presented in Table 1 and are compared to a single AKF working alone and to a mixture of experts whose gating network is a random process. This last modular predictor allows no learning, the experts are not associated to the target's dynamics, and the AKFs cannot thus properly adapt their parameters. The single AKF is limited, its noise properties are not valid over the whole trajectory, inducing important maximum errors.

For an experiment consisting in following a target in a 3-D space with a three-degree-of-freedom robot, the neuro-controller associated with the modular predictor allows to divide the tracking error by 3.

Absolute Prediction Error (pixels)	X coordinate			Y coordinate		
	Mean	Std Dev.	Max.	Mean	Std Dev.	Max.
Single AKF	0.5162	9.431	178.481	0.432	7.532	161.599
Mixture with softmax gating network	0.375	2.716	22.110	0.2938	2.632	13.057
Mixture with random gating network	31.270	70.296	148.361	23.35	52.269	195.607

Table 1. The absolute prediction errors for a single AKF and mixtures of experts with the softmax learning algorithm and a random weighting.

6 Conclusion

For time varying environments, neural networks alone do not always constitute an efficient solution. The Kalman filter, inherently, allows to consider a system's dynamics. In order to make this filter as flexible as neural networks, and thus to work with systems that cannot be modeled with sufficient precision, we introduce an approach using multiple adaptive state models in a modular structure where each filter specializes to handle specific noise statistics.

These state model adaptive filters adjust their dynamics to the system's dynamics, and thus estimate the system's current state. Using different adaptive filters allows to split the system's workspace into several parts characterized by different noise properties. The mixture of experts structure, based on the divide-and-conquer principle, perfectly mediates the filters' responses and learns to associate the different filters to the different system's behaviors.

This new modular approach is evaluated by predicting a moving target's movements in a 3-D space with no *a priori* knowledge. Visually controlled, a robot tracking task has been improved by the accuracy of the predictions.

References

[1] P. I. Corke, "Visual Control of Robot Manipulators - A Review," in Visual Servoing, vol. 7, Robotics and Automated Systems, K. Hashimoto, Ed., World Scientific, pp. 1-31, 1993.

[2] J.-P. Urban, J.-L. Buessler, and J. Gresser, "Neural Networks for Visual Servoing in Robotics," in Soft Computing for Intelligent Robotic Systems, L. C. Jain and T. Fukuda, Eds., Springer-Verlag, pp. 81-111, 1998.

[3] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive Mixtures of Local Experts," Neural Computation, vol. 3, pp. 79-87, 1991.

[4] A. Moghaddamjoo and R. L. Kirlin, "Robust Adaptive Kalman Filtering," in Approximate Kalman Filtering, vol. 2, Approximations and Decompositions, G. Chen, Ed., World Scientific Publishing Company, pp. 65-85, 1993.

[5] P. Wira and J.-P. Urban, "A New Adaptive Kalman Filter Applied To Visual Servoing Tasks," Proc. of 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, Brighton, UK, 2000.