Sixth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2008)
"Partnering to Success: Engineering, Education, Research and Development"
June 4 – June 6 2008, Tegucigalpa, Honduras.

Tracking Objects On Image Sequences With Unscented Kalman Filter and Dynamic Neural Networks

Fernando E. Serrano
Universidad Tecnológica Centroamericana, Tegucigalpa, F. Morazan , Honduras, feserrano@unitec.edu

ABSTRACT

Tracking objects on image sequences is a problem that has been studied by several authors using the Kalman filter as a position and orientation estimator, this approach is actually used in different applications such as tracking vehicles in traffic control system, closed loop control for robotic positioning systems and orientation tracking of surgical devices using a camera to determine the position in the task space.

In order to make the estimations of the position of an object in an image sequence, it is necessary to compute the states dynamics equations that define the object trajectory according to an "a priori" behavior of the dynamic equations and these values are used to estimate the future position in the image frame. These state transition equations are defined by the knowledge of the system dynamics; therefore, they can be represented by the respective spatial transformations. In this paper a different approximation is proposed representing the state transition equations using a dynamic neural network that is used later by the Kalman filter estimator to compute the future states that estimate the position in the image.

Keywords: Kalman Filter, Computer Vision, Estimators, Stochastic Systems, Neural Networks

1. INTRODUCTION

Recently, the Kalman filter has been an algorithm that is applied as a filter to supress noise on different kinds of stochastic systems and as an state stimator in problems where is necessary to deal with uncertainties; the estimation properties make this algorithm appropiate to be used on motion tracking in image sequences obtained from a video system that could be used in a robotic controller to follow a moving object. To solve this problem, several solution have been given using the extended Kalman filter or the unscented Kalman filter, in which both algorithms provide good approximations for the trajectory of an object in all the images of a sequence, but the unscented Kalman filter is suitable for tracking problems because it is not necessary to linearize about a nominal trajectory using the Jacobian matrix reducing the computational complexity, for this reason the unscented Kalman filter is proposed in this paper.

The unscented Kalman filter uses a deterministic sampling approach to compute the mean and covariance and there is no need to linearize the system, making this algorithm suitable to track an object when the trajectory is nonlinear, there is not enough a priori information and there are perturbations on the nominal trajectory. In most of the cases at least one of the previous condition occurs; for this reason, a dynamic neural network is proposed to represent the trajectory due to the nonlinear characteristic of the system. This paper is divided in four sections; in section two the geometric and dynamic characteristics of a moving object is presented to determine the movement characteristics and the spatial transformations that are necessary to represent the position changes. Section three the unscented kalman filter is applied considering the implementation of the dynamic neural network. Section four experimental results are shown . In section five, conclusions are presented.

2 OBJECT TRAJECTORY EQUATIONS

First of all it is necessary to define a set of equations to describe the object trajectory in order to get the dynamics equations and its respective state transition matrix that will be used on every iteration of the kalman filter. In previous works[3] the quaternions have been implemented to determine the position and rotation of the object; but in this paper we will define a set of simple linear transformations which describe the object translation in the whole image because it is not necessary to know the rotation of the object for this purpouse One of the problems to define a set of dynamics equations is the poor previous knowledge of the system in which the position and velocity variables could not be represented by an algebraic function, that is why is very important to define here the dynamic behaviour of the object because it will be transformed in the next section into a dynamic neural network keeping only the state transition matrix for the Kalman filter equation.

Consider a linear transformation function described by:

$$\Re \rightarrow \Phi_k \tag{1}$$

This function will be a linear transformation matrix that changes at every time step *k* and it is the state transition matrix used on the dynamic system equation that is converted later into a dynamic neural network. Due to the linear characteristic of this function this must be an orthogonal matrix that can be used to implement a similarity transformation for the state vector as shown in the equation 2

$$x_{k+1} = \Phi_k x_k \Phi_k^T \tag{2}$$

This state transition matrix, which results from the linear transformations, are used at every time step *k* in the estimation of the error covariance matrix and the dynamic neural network is used to compute the states on each sample image of the sequences, therefore, it is not necessary to implement a set of nonlinear difference equations avoiding the linearization of the system and reducing the computational complexity when the filter is implemented in real time.

The state dynamics of the object is represented by a model based on the position characteristics [3] that will be converted into a dynamic neural network representation. The state vector is shown in equation 3

$$x_{k+1} = \begin{bmatrix} x_k & y_k & v_x & v_y \end{bmatrix}^T \tag{3}$$

In equation 3 the state vector contains the position and velocities of the object on each image at every time step, each of these variables will be estimated with the neural network because the motion equations can not be given by a single formula due to the uncertainties on the object's trajectory

Due to the nonlinearity and the lack of a priori knowledge of the object position , it is extremely difficult to establish a set of nonlinear difference equations, but a representative model is shown in equation 4 because it will be used later to show the relation of the state equations and the neural network. The dynamic systems which represent the position and velocity [3] is

$$X_{k+1} = \Phi X_k + \Gamma W(k) \tag{4}$$

Where X_k is the state vector, Φ is the transition matrix, Γ is the disturbance matrix and *W(k)* is the unknow zero mean Gaussian process noise with assumed know covariance. In this section all the dynamics equations are shown because they are substituted by artificial neural networks because this is a good method to obtain the system's model when all the dynamic equations are unknow or there is not enough information to build it.

3 UNSCENTED KALMAN FILTER STIMATOR

This section is divided in two subsections, in section 3.1 the dynamic neural networks are explained showing how they are built based on the equations that describe the object trajectory on the image that were explained in the previous section. In the section 3.2 the unscented kalman Filter is described to show the algorithm steps that must be done to estimate the object position in the image.

3.1 DYNAMIC NEURAL UNIT DEFINITION

Dynamics neural networks are used in applications where it is necessary to represent a plant on control applications to study the stability properties and its respective control law to be applied on a dynamic system or as a estimator that could be used in a model reference in nonlinear control systems [2]. The main idea in this paper is to apply the dynamic neural network on the equations defined in the previous section because of the nonlinear characteristics, making the DNU a feasible model for this purpose.

Before the dynamic neural network is defined, it is important to know about the definition of the dynamic neural unit (DNU), because it is basic part of the network that represents the trajectory's dynamics.

The dynamic neural unit (DNU) is an element that is designed based on the conventional neural network structure (perceptron) but the DNU has a dynamic response that mades this unit appropiate to represent nonlinear systems. The DNU comprises of memory elements, and feedforward and feedback synaptic weights. The output of this dynamic structure is to a time varying nonlinear activation function *f(.)* [2]

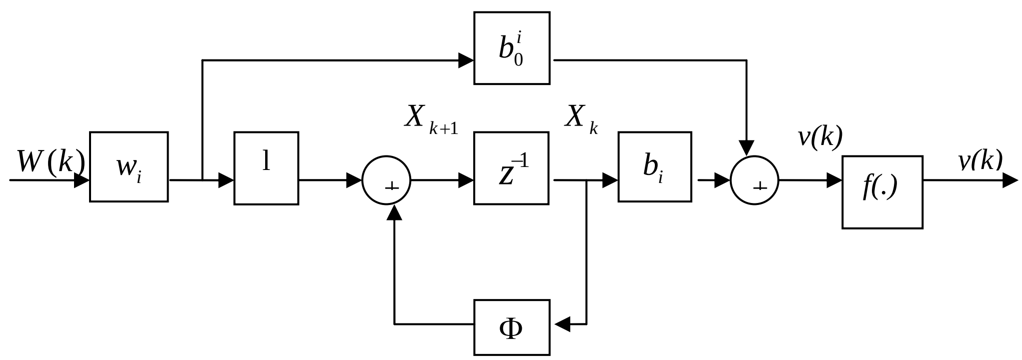

Figure 1: Dynamic Neural Unit

In figure 1 a DNU block diagram that introduces a linear dynamic system with an IIR filter structure[2] with a time delay used to record the previous state X_{k+1} to estimate the future state X_k . Based on the DNU a multilayer dynamic neural network is designed to estimate the states that represent the position and velocity of the object. A multilayer dynamic neural network unit can be formed cascading several DNU's [1] and for this implementation only three layers are used and these are the input layer, intermediate layer and output layer [1]. The dynamic neural network model is designed based on the equations 3 and 4 in which the state vector and state space model is defined. In figure 1, w_i represents the weight matrix which performs the same function of the static neural network weight matrix; this weights are applied to the zero mean Gaussian input *W(k)* which is the input to the network. It can be notice that is not necessary to define a disturbance matrix for the trajectory's equation using this IIR filter structure because it is defined by :

$$\Gamma = lw_i \tag{5}$$

Therefore the disturbance matrix can be found in the network's training process. The matrix b_0^i is the block diagonal matrix associated with the feedforward network parameter, that can be ommitted because the state vector will be the output of the system which provide the measurement variables (position and velocity) to the kalman filter estimator. The matrix component $b_i = \begin{bmatrix} b_1^i & b_2^i & b_3^i & b_4^i \end{bmatrix}$ is a feedforward vector that is implemented in the output equation, the varying nonlinear activation function *f(.)* is chosen to be a continous and differentiable nonlinear function [2] e.g. sigmoidal function in the form:

$$f(x; g_1, g_2) = \frac{1}{1 + e^{-g_2(x - g_1)}} \tag{6}$$

Where the slope is determined by $g_2 > 0$ and g_1 is the shape parameter for the sigmoidal activation function. The state space model defined by the equations 3 and 4 is implemented with a three layer DNU and it is represented by:

$$\begin{aligned} X_{k+1} &= \Phi X_k + \Gamma W(k) \\ Y_k &= f(b_i X_k) \end{aligned} \tag{7}$$

The output vector Y_k is used in the kalman filter update process and it is calculated by the three layer dynamic neural network that is shown in figure 2 [1]:

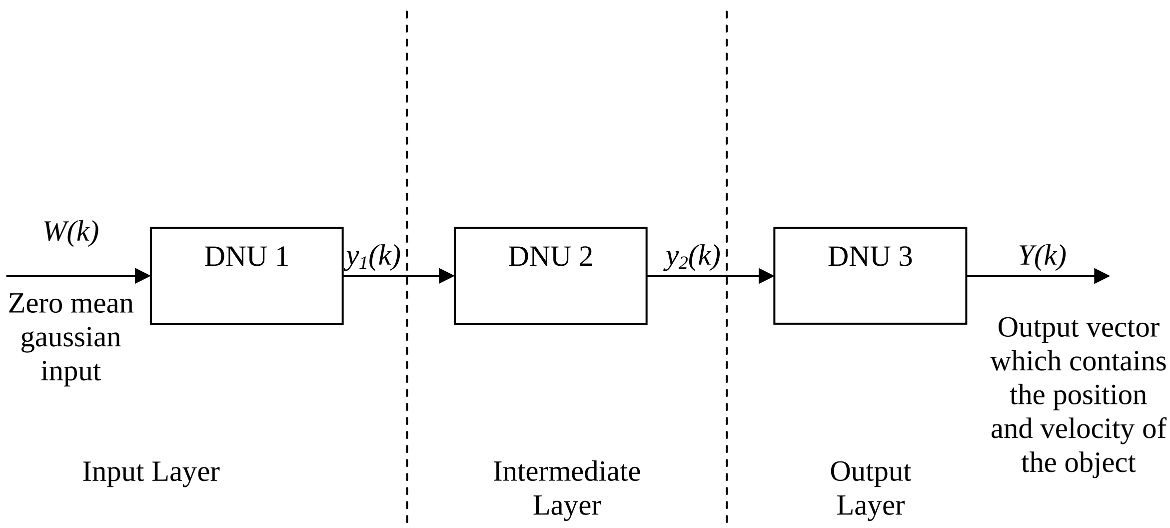

Figure 2: Multilayer Dynamic Neural Network

The training process of the dynamic network is done by the algorithm described in [1] where an optimal measurement parameter must be minimized to find the weights and the bias for the network. This cost function is:

$$J = \frac{1}{2} E[(y_d(k) - y(k))^2] \tag{8}$$

Where E is the expected value, $y_d(k)$ is the desired neuron response and $y(k)$ is the actual neuron response.

Each component of the neural network Ω is found using the steepest descent algorithm:

$$\frac{\partial y(k)}{\partial \Omega} = \frac{\partial y(k)}{\partial v}\frac{\partial v}{\partial \Omega} \tag{9}$$

Where *v* is the output of the dynamic processor in figure 1.

3.2 KALMAN FILTER ESTIMATION

The kalman filter estimation is done by the unscented kalman filter using the state dynamics and output equations defined in equation 7 using the dynamic neural equations explained in the previous section.

The unscented Kalman filter is used on this paper because under nonlinear conditions it performs better than the extended Kalman filter. The update process is explained in [3] and [5] using the state vector represented by equation 3 the update process is done by computing a set of L *x* *(2L+1)* sigma points where *L* is the dimension of the state vector that for this case is four. Then the columns of X_{k-1} are computed by:

$$(X_{k-1})_0 = \hat{X}_{k-1}$$

$$(X_{k-1})_0 = \hat{X}_{k-1} + (\sqrt{(L+\lambda)P_{k-1}})_i \text{, } i=1 \ldots L \tag{10}$$

$$(X_{k-1})_0 = \hat{X}_{k-1} - (\sqrt{(L+\lambda)P_{k-1}})_{i-1} \text{, } i=L+1 \ldots 2L$$

where $\sqrt{(L+\lambda)P_{k-1}}$ is the ith column of the square root matrix that is computed by a cholesky decomposition and λ is defined by:

$$\lambda = \alpha^2 (L+k) - L \tag{11}$$

where α is a scaling factor and *k* is a secondary scaling parameter. The prediction step is done by propagating X_{k-1} in time using the dynamic neural network system implementation defined by:

$$(X_k)_i = f((X_{k-1})_i) \text{ , } i=1 \ldots 2L \tag{12}$$

where *f* is the nonlinear system equation represented by the dynamic neural network model that is shown in figure 2. The a priori state estimate is given by:

$$\hat{X}_k^- = \sum_i^{2L} W_i^m (X_k)_i \tag{13}$$

where

$$W_i^m = \frac{\lambda}{2(L+\lambda)}, \; i = 1 \ldots 2L \tag{14}$$

The a priori error covariance is given by:

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} \left[(X_k)_i - \hat{X}_k^- \right] \left[(X_k)_i - \hat{X}_k^- \right]^T \tag{15}$$

Where $W_i^{(c)}$ is defined is [3]. The correction step is computed with the formula:

$$(Z_k)_i = h((X_k)_i), \; i = 0 \ldots 2L \tag{16}$$

$$\hat{Z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} (Z_k)_i$$

With the transformed state vector $\hat{Z}_k^-$ the a posteriori state estimate is given by:

$$\hat{X}_k = \hat{X}_k^- + K_k (Z_k - \hat{Z}_k^-) \tag{17}$$

where K_k is the kalman gain defined by:

$$K_k = P_{\hat{X}_k \hat{Z}_k} P_{\hat{Z}_k \hat{Z}_k}^{-1} \tag{18}$$

$P_{\hat{X}_k \hat{Z}_k}$ and $P_{\hat{Z}_k \hat{Z}_k}$ are defined in [3]

Finally the last step of the kalman filter is given by the a posteriori compute of the error covariance given by:

$$P_{kk} = P_k^- - K_k P_{Z_k Z_k} K_k^T \tag{19}$$

4 EXPERIMENTAL RESULTS

In this section a simulation was done using a sequence of fifteen images to test the kalman filter estimator and the dynamic neural network on the prediction of the position of the paddle that is hold by the player on the images. A two by one vector with the variables *x* and *y* represents the position of the selected item on the image's coordinate system. A morphology operation was done to select the area of the object that is tracked with the algorithm.

The parameters used for the unscented kalman filter estimator are:

$Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$ covariance of the process

R=0.01 , covariance of measurement

The parameters for the unscented transformations are:

$\alpha = 0.0001$

$k = 0$

$\beta = 2$

The dynamic neural network was trained using an incremental training with a learning rate of 0.001, this neural network is used to represent the object's dynamic of the moving object given by the state space representation *f*. For this simulation the measurement equation is:

$$z = x \qquad (20)$$

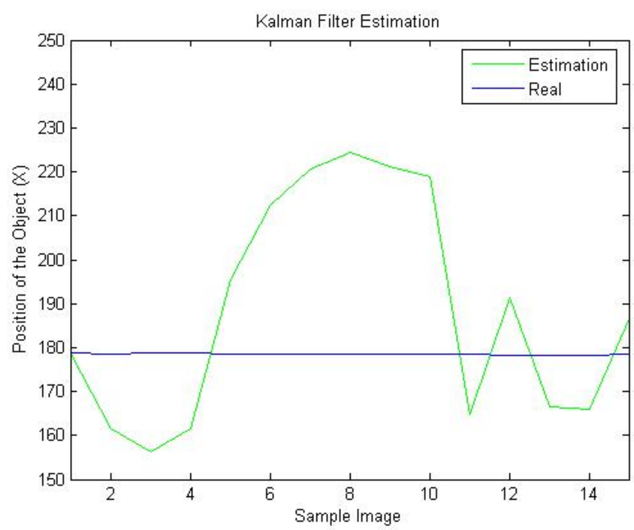

Figure 3: Kalman Filter Estimation for X

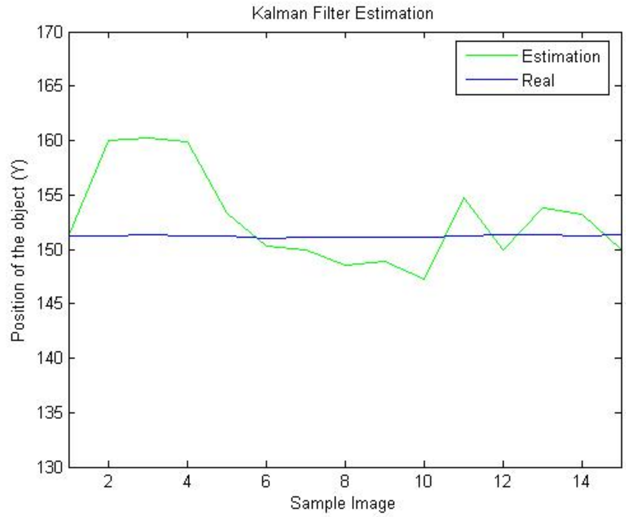

Figure 4: Kalman Filter Estimation for Y

In figures 3 and 4 the kalman filter estimation plots are shown for the X and the Y axis, it can be notice how the unscented Kalman filter algorithm gives the approximation for the centroid position of the paddle that is followed on each image. Obviously, different results could be obtained by changing the neural network learning rate and the unscented Kalman filter parameters.

On the set of figures shown on figure 5, the Kalman filter and the neural network estimate the position of the paddle on the image with a loss of precision in several cases due to the feature extraction process. The estimated position is represented by the blue circle and the original position is represented by the yellow circle on each image.

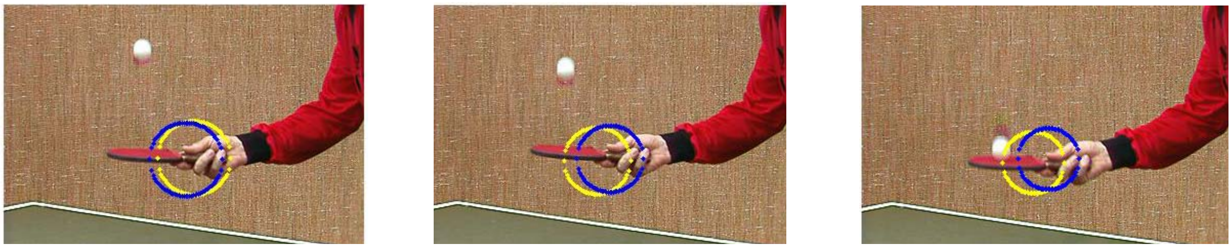

Figure 5: Real and Estimated Position

5 CONCLUSIONS

The substitution of the dynamics equation by a dynamic neural network is a good approach which allow us to simplify the analysis and implementation of the nonlinear characteristics of the object's trajectory but the learning rate an the number of layers used on the network could affect the nonlinear dynamics response of the system and the Kalman filter could yield wrong results. The feature extraction is a very important issue when a tracking algorithm is developed, because in occluded images this step could be complex due to the large number of components; some classification algorithms must be applied to extract an element from every image.

The unscented Kalman filter is an algorithm that is suitable for tracking problems due to the uncertainties and the nonlinear characteristics that can be found on this kind of problems. Changing the parameters of the unscented Kalman filter could provide different results.

REFERENCES

[1]Park Seong-Wook, Seo Bo-Hyeok (1995). Design of Controller for Nonlinear System Using Dynamic Neural Unit Model. IEEE Proceedings

[2] Krzysztof Patan (2007). Stability Analysis and the Stabilization of a Class of Discrete-Time Dynamic Neural Networks. IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 18, NO. 3, MAY 2007

[3] Joseph J. LaViola Jr. A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion. IEEE Proceedings of the American Control Conference Denver, Colorado June 4-6, 2003

[4] Murat Efe, Derek P Atherton. Maneuvering Target Tracking With an Adaptive Kalman Filter. Proceedings of the 37th IEEE Conference on Decision & Control Tampa, Florida USA December 1998.

[5] Simo Särkkä, On Unscented Kalman Filtering for State Estimation of Continuous-Time Nonlinear Systems. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 52, NO. 9, SEPTEMBER 2007

[6] J.Z. Sasiadek, J. Khe. Sensor Fusion Based on Fuzzy Kalman Filter. IEEE Second Workshop on Robot Motion and Control, October 18-20 2001