IMPROVEMENT OF THE VISUAL SERVOING TASK WITH A NEW TRAJECTORY PREDICTOR

The Fuzzy Kalman Filter

C. Pérez, N. García, J. M. Sabater, J. M. Azorín and O. Reinoso
Miguel Hernández University, Avenida de la Universidad S/N, Elche, Spain

L. Gracia
Technical University of Valencia, Camino Vera S/N, Valencia, Spain

Keywords: Visual servoing, fuzzy systems, vision/image processing, Kalman filter.

Abstract: Visual Servoing is an important issue in robotic vision but one of the main problems is to cope with the delay introduced by acquisition and image processing. This delay is the reason for the limited velocity and acceleration of tracking systems. The use of predictive techniques is one of the solutions to solve this problem. In this paper, we present a Fuzzy predictor. This predictor decreases the tracking error compared with the classic Kalman filter (KF) for abrupt changes of direction and can be used for an unknown object’s dynamics. The Fuzzy predictor proposed in this work is based on several cases of the Kalman filtering, therefore, we have named it: Fuzzy Kalman Filter (FKF). The robustness and feasibility of the proposed algorithm is validated by a great number of experiments and is compared with other robust methods.

1 INTRODUCTION

During the last few years, the use of visual servoing and visual tracking has been more and more common due to the increasing power of algorithms and computers.

Visual servoing and visual tracking are techniques that can be used to control a mechanism according to visual information. This visual information is available with a time delay, therefore, the use of predictive algorithms are widely extended (notice that prediction of the object’s motion can be used for smooth movements without discontinuities).

The Kalman filter (Kalman, 1960) has become a standard method to provide predictions and solve the delay problems (considered the predominant problem of visual servoing) in visual based control systems (Corke, 1998), (Dickmanns and V., 1988) and (Wilson and Bell, 1996).

The time delay is one of the bigger problems in this type of systems. For practically all processing architectures, the vision system requires a minimum delay of two cycles, but for on-the-fly processing, only one cycle of the control loop is needed (Chroust and Vincze, 2003).

Authors of (Chroust and Vincze, 2001) demonstrate that steady-state Kalman filters ($\alpha\beta$ and $\alpha\beta\gamma$ filters) performs better than the KF in the presence of abrupt changes in the trajectory, but not as good as the KF for smooth movements. Some research works about the motion estimation are presented in (S. Soatto and Perona, 1997) and (Z. Duric and Rivlin, 1996). Further, some motion understanding and trajectory planning based on the Frenet-Serret formula are described in (J. Angeles and Lopez-Cajun, 1988), (Z. Duric and Rosenfeld, 1998) and (Z. Duric and Davis, 1993). Using the knowledge of the motion and the structure, identification of the target dynamics may be accomplished.

To solve delay problems, taking into account these considerations, we propose a new prediction algorithm, the fuzzy Kalman filter (FKF). This filter minimizes the tracking error and works better than the classic KF because it decides what of the used filters ($\alpha\beta$ slow / $\alpha\beta$ fast (Chroust and Vincze, 2003), $\alpha\beta\gamma$, $K_v$, $K_a$ and $K_j$) must be employed. The transition between them is smooth avoiding discontinuities.

These five filters should be used in a combination because: The Kalman filter is considered one of the reference algorithms for position prediction (but we must consider the right model depending on the object’s dynamics: velocity—acceleration—jerk). When
the object is outside the image plane, the best prediction is given by steady-state filters \((\alpha \beta / \alpha \beta y\) depending on the object’s dynamics: velocity—acceleration). Obviously, considering more filters and more behaviour cases, FKF can be improved but computational cost of additional considerations can be a problem in real-time execution. These five filters are considered by authors as the best consideration (solution taking into account the prediction quality and the computational cost). This is the reason to combine these five filters to obtain the FKF.

This paper is focused on the new FKF filter and is structured as follows: in section 2 we present the considered dynamics, the considered dynamics is a Jerk model with adaptable parameters obtained by KFs (Nomura and T., 2000), (Li and Jilkov, 2000) and (Mehrotra and Mahapatra, 1997). In section 3, we present the block diagram for the visual servoing task. This block diagram is widely used in several works like (Corke, 1998) or (Chroust and Vincze, 2003). Section 4 presents the basic idea applied in our case (see (Wang, 1997b) and (Wang, 1997a)), but the main work done is focused in one of the blocks described in section 3, the Fuzzy Kalman Filter (FKF) is described in section 5.

In section 6, we can see the results with simulated data. These results show that FKF can be used to improve the high speed visual servoing tasks. This section is organized in two parts: in the first one (Subsection 6.1), the analysis of the FKF behaviour is focussed and in the second one (Subsection 6.2) their results are compared those with achieved by Chroust and Vince (Chroust and Vincze, 2003) and with CPA (Tenne and Singh, 2002) algorithm (algorithm used for aeronautic/aerospace applications). Conclusions and future work are presented in section 7.

2 THE DYNAMICS OF A MOVING OBJECT

The object’s movement is not known (a priori) in a general visual servoing scheme. Therefore, it is treated as an stochastic disturbance justifying the use of a KF as a stochastic observer. The KF algorithm presented by Kalman (Kalman, 1960) starts with the system description given by 1 and 2.

\[
\begin{align*}
x_{k+1} &= F \cdot x_k + G \cdot \xi_k \\
y_k &= C \cdot x_k + N \cdot \eta_k
\end{align*}
\]

where \(x_k \in \mathbb{R}^{n_1}\) is the state vector and \(y_k \in \mathbb{R}^{n_1}\) is the output vector. The matrix \(F \in \mathbb{R}^{n_1 \times n_1}\) is the so-called system matrix which describes the propagation of the state from \(k\) to \(k+1\) and \(C \in \mathbb{R}^{m \times n_1}\) describes the way in which the measurement is generated out of the state \(x_k\). In our case of visual servoing \(m = 1\) (because only the position is measured) and \(n = 4\). The matrix \(G \in \mathbb{R}^{m \times n_1}\) distributes the system noise \(\xi_k\) to the states and \(\eta_k\) is the measurement noise. In the KF the noise sequences \(\xi_k\) and \(\eta_k\) are assumed to be gaussian, white and uncorrelated. The covariance matrices of \(\xi_k\) and \(\eta_k\) are \(Q\) and \(R\) respectively (these expressions consider 1D movement). A basic explanation for the assumed gaussian white noise sequences is given in (Maybeck, 1982).

In the general case of tracking, the usual model considered is a constant acceleration model (Chroust and Vincze, 2003), but in our case, we consider a constant jerk model described by matrices \(F\) and \(C\) are:

\[
F = \begin{bmatrix} 1 & T & T^2/2 & T^3/6 \\ 0 & 1 & T & T^2/2 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}
\]

where \(T\) is the sampling time. This model is called a constant jerk model because it assumes that the jerk \((dx^3(t)/dt^3)\) is constant between two sampling instants.

\[
\begin{align*}
&\frac{a-a_i}{T-t_i} = \frac{\Delta a}{\Delta t} = J_0 \\
x(t) = x_i + v_i(t-t_i) + \frac{1}{2}a_i(t-t_i)^2 + \frac{1}{6}J_0(t-t_i)^3 \\
v(t) = v_i + a_i(t-t_i) + \frac{1}{2}J_0(t-t_i)^2 \\
a(t) = a_i + J_0(t-t_i) \\
J(t) = J_0
\end{align*}
\]

where, \(x\) is the position, \(v\) is the velocity, \(a\) is the acceleration and \(J\) is the jerk. So the relation between them is:

\[
x(t) = f(t); \dot{x}(t) = v(t); \ddot{x}(t) = a(t); \dddot{x}(t) = J(t)
\]

3 DESCRIPTION OF THE CONTROL SYSTEM

The main objective of the visual servoing is to bring the target to a position of the image plane and to keep it there for any object’s movement. In figure 1 we can see the visual control loop presented by Corke in (Corke, 1998). The block diagram can be used for a moving camera and for a fixed camera controlling the motion of a robot. Corke use a KF to incorporate a feed-forward structure. We incorporate the FKF algorithm in the same structure (see figure 2) but reordering the blocks for an easier comprehension.
The most common fuzzy inference process used is known as Mamdani’s fuzzy inference method, but on the other hand, we can find a so-called Sugeno, or Takagi-Sugeno-Kang, method of fuzzy inference. It was introduced in 1985 (Sugeno, 1985) and is similar to the Mamdani’s method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant (for more information see (Passino and S., 1988)). For Sugeno regulators, we have a linear dynamic system as the output function so that the i^{th} rule has the form:

\[ \hat{x}_i(t) = \sum_{i=1}^{R} \left( \sum_{j=1}^{R} (U_i x_j(t) + V_i u_j(t)) \mu_i(z(t)) \right) \]

where \( x(t) = [x_1(t), x_2(t), ..., x_n(t)]^T \) is the state vector, \( u(t) = [u_1(t), u_2(t), ..., u_m(t)]^T \), \( U_i \) and \( V_i \) are the state and input matrices and \( z(t) = [z_1(t), z_2(t), ..., z_p(t)]^T \) is the input to the fuzzy system, so:

\[ \hat{x}(t) = \sum_{i=1}^{R} (\sum_{j=1}^{R} \mu_i(z(t))) \]

or

\[ \hat{x}(t) = \left( \sum_{i=1}^{R} (U_i \xi_i(z(t))) \right) x(t) + \left( \sum_{i=1}^{R} (V_i \xi_i(z(t))) \right) u(t) \]

where

\[ \xi^T = [\xi_1, ..., \xi_R] = \frac{1}{R} [\mu_1, ..., \mu_R] \]

Our work is based on this idea and these expressions (see (Passino and S., 1988) for more details). We have mixed the Mamdani’s and the Sugeno’s idea because we have implemented an algorithm similar to Sugeno but not for linear systems. We obtain a normalized weighting of several non linear recursive expressions. The system works like we can see in figure 3 (see section 5).
5 THE FUZZY KALMAN FILTER (FKF)

We have developed a new filter that mixes different types of Kalman filters depending on the conditions of the object’s movement. The main advantage of this new algorithm is the non-abrupt change of the filter’s output.

Consider the nonlinear dynamic system

\[ \dot{x} = f_1(x, u); \quad y = g_1(x, u) \]

as each one of the filters used. The application of the fuzzy regulator in our case produces the next state-space expression:

\[ \dot{y} = F \cdot \omega(x, u) \]

where

\[ \omega(x, u) = \frac{\mu_i(x, u)}{\sum_{i=1}^{N} \mu_i(x, u)} \]

The final system obtained has the same structure than filters used:

\[ \dot{x} = f_2(x, u); \quad y = g_2(x, u) \]

Figure 3 shows the FKF block diagram. In this figure, we can see that the general input is the position sequence of the target \((x_k)\). Using this information, we estimate the velocity, acceleration and jerk of the target in three separate KFs (Nomura and Naito present the advantages of this hybrid technique in (Nomura and T., 2000)). This information is used as 'Input MF' to obtain \(F_1(ins), F_2(v), F_3(a)\) and \(F_4(j)\). These MF inputs are the fuzzy membership functions defined in figure 4. The biggest KF block (rounded) shown in his figure is a combination of all used algorithms in the fuzzy filter \((\alpha\beta)\) (Chroust and Vincze, 2003), \(\alpha\beta\) (Chroust and Vincze, 2003), \(\alpha\beta\) (Chroust and Vincze, 2003), \(\alpha\beta\) (Chroust and Vincze, 2003). This block obtains the output of all specified filters. The 'Output MF' calculates the final output using the \(R_i\) rules.

Now, we present the rules \((R_i)\) considered for the fuzzy filter:

- \(R_1\): IF object IS inside AND velocity IS low AND acceleration IS low AND jerk IS low THEN \(FKF=Kv\)
- \(R_2\): IF object IS inside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN \(FKF=Kv\)
- \(R_3\): IF object IS outside AND velocity IS low AND acceleration IS low AND jerk IS low THEN \(FKF=\alpha\beta\) \(\alpha\beta\)
- \(R_4\): IF object IS outside AND velocity IS medium AND acceleration IS low AND jerk IS low THEN \(FKF=\alpha\beta\) \(\alpha\beta\)
- \(R_5\): IF object IS inside AND velocity IS high AND acceleration IS low AND jerk IS low THEN \(FKF=Kv\)
- \(R_6\): IF object IS inside AND acceleration IS medium AND jerk IS low THEN \(FKF=0.8 \cdot \alpha\beta + 0.2 \cdot Ka\)
- \(R_7\): IF object IS outside AND acceleration IS medium AND jerk IS low THEN \(FKF=0.8 \cdot \alpha\beta + 0.2 \cdot Ka\)
- \(R_8\): IF object IS inside AND acceleration IS high AND jerk IS low THEN \(FKF=Kv\)
- \(R_9\): IF object IS outside AND acceleration IS high AND jerk IS low THEN \(FKF=\alpha\beta\)
- \(R_{10}\): IF jerk IS high THEN \(FKF=Kj\)

These rules have been obtained empirically, based on the authors experience using the Kalman filter in different applications.

Notice that rule \(R_{10}\) (when jerk is high) shows that the best filter considered is \(Kj\) and it does not depend on the object’s position (inside or outside) velocity/acceleration value (low, medium or high).

We have used a product inference engine, singleton fuzzifier and centre average defuzzifier. Figure 4 presents the fuzzy sets definition where the size, \(\mu_{vel} = \mu_{acc} = 2m/s, \sigma_{vel} = \sigma_{acc} = 0.5,\)

\[
\begin{align*}
\mu_{vel} &= c_{vel} = 1, \quad c_{acc} = 1, \\
\mu_{vel} &= d_{vel} = d_{acc} = 3, \quad j_{vel} = j_{acc} = 1 \\
\mu_{vel} &= f_{vel} = f_{acc} = 1
\end{align*}
\]

(These values have been empirically obtained).
6 RESULTS

This section is composed by two different parts: first (section 6.1), we analyze the prediction algorithm presented originally in this paper (FKF block diagram shown in figure 3) and second (section 6.2), some simulations of the visual servoing scheme (see figure 2) are done including the FKF algorithm.

6.1 Fuzzy Kalman Filter (FKF) Results

In figure 5, we show the effectiveness of our algorithm's prediction compared with the classical KF methods. In this figure, we can see positions \( P_r^k \) (actual object position), \( P_{r-1}^k \) (object position in \( k-1 \)) and \( P_{r-2}^k \) (object position in \( k-2 \)). Next real position of the object will be \( P_r^{k+1} \), and points from \( \tilde{P}_1^{k+1} \) to \( \tilde{P}_6^{k+1} \), represent the prediction obtained by each single filter. The best prediction is given by the FKF filter. This experiment is done for a parabolic trajectory of an object affected by the gravity acceleration. (See figures 5 and 6).

We have done a lot of experiments for different movements of the object and we have concluded that our FKF algorithm works better than the others filters compared (filters compared are: \( \alpha \beta, \alpha \beta \gamma, K_v, K_a, K_j \) and CPA -see section 6.2- with our FKF). Figure 6 shows the real trajectory and the trajectory predicted for each filter. For this experiment, we have used the first four real positions of the object as input for all filters and they predict the trajectory using only this information. As we can see in this figure, the best prediction is again the FKF.

6.2 Visual Servoing Control Scheme Results

To prove the control scheme presented in figure 2, we have used the object motion shown in figure 7 (up). This target motion represents a ramp-like motion between \( 1 < t < 4 \) seconds and a sinusoidal motion for \( t > 6 \) seconds. This motion model is corrupted with a noise of \( \sigma = 1 \) pixel. This motion is used by Stefan Chroust and Markus Vincze in (Chroust and Vincze, 2003) to analyze the switching Kalman filter (SKF).

For this experiment, we compare the proposed filter (FKF) with a well known filter, the Circular Prediction Algorithm (CPA) (Tenne and Singh, 2002). In figure 7 (down), we can see the results of FKF and CPA algorithms. For changes of motion behaviour, the FKF produce less error than CPA. For the change in \( t=1 \), the FKF error is \([+0.008, -0]\) and \([+0.015, -0.09]\) for the CPA. For the change in \( t=4 \), FKF error =
Table 1: Numerical comparative for dispersion value of all filters implemented (bounce of a ball experiment).

<table>
<thead>
<tr>
<th>Init. pos.</th>
<th>αβ</th>
<th>αβγ</th>
<th>Kv</th>
<th>Ka</th>
<th>Kj</th>
<th>FKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0.619</td>
<td>0.559</td>
<td>0.410</td>
<td>0.721</td>
<td>0.877</td>
<td>0.353</td>
</tr>
<tr>
<td>40(bis)</td>
<td>0.547</td>
<td>0.633</td>
<td>0.426</td>
<td>0.774</td>
<td>0.822</td>
<td>0.340</td>
</tr>
<tr>
<td>50</td>
<td>0.588</td>
<td>0.663</td>
<td>0.439</td>
<td>0.809</td>
<td>0.914</td>
<td>0.381</td>
</tr>
<tr>
<td>70</td>
<td>0.619</td>
<td>0.650</td>
<td>0.428</td>
<td>0.700</td>
<td>0.821</td>
<td>0.365</td>
</tr>
<tr>
<td>90</td>
<td>0.630</td>
<td>0.661</td>
<td>0.458</td>
<td>0.818</td>
<td>0.857</td>
<td>0.343</td>
</tr>
<tr>
<td>150</td>
<td>0.646</td>
<td>0.682</td>
<td>0.477</td>
<td>0.848</td>
<td>0.879</td>
<td>0.347</td>
</tr>
</tbody>
</table>

[+0.0072] and CPA error = [+0.09,-0.015]. For the change in t=6, FKF error = [+0.022,-0] and CPA error = [+0.122,-0.76]. For the region 6 < t < 9 (sinusoidal movement between 2.5m and 0.5m) both algorithms work quite similarly: FKF error = [-0.005] and CPA error = [-0.0076]. CPA filter works well because it is designed for movements similar to a sine shape, but we can compare this results with the SKF filter proposed in (Chroust and Vincze, 2003) and SKF works better (due to the AKF (Adaptive Kalman Filter) effect). Therefore, the FKF filter proposed works better than CPA for all cases analyzed but comparing FKF with SKF, FKF is better for t=1, t=4 and t=6 but not for 6 < t < 9 (sinusoidal movement).

Figure 9 shows the zoom region 0 < t < 2 and −0.02 < Δx_p < 0.02 of the same experiment. In this figure, we can see the fast response of the FKF proposed.

6.3 Experimental Results

Experimental results are obtained for this work using the following setup: Pulnix GE series high speed camera (200 frames per second), Intel PRO/1000 PT Server Adapter card, 3.06GHz Intel processor PC computer, Windows XP Professional O.S. and OpenCV blob detection library.

For this configuration, the bounce of a ball on the ground is processed to obtain data shown in figure 10. Results of this experiment are presented in table 1. In this table, we can see the dispersion of several filters. The FKF dispersion is less than αβ, αβγ, Kv, Ka and Kj although FKF is a combination of them. This table contains data from this particular experiment (the bounce of a ball on the ground). For this experiment, the position of the ball is introduced to the filters to prove the behaviour of them. The filter proposed (FKF) is the best analyzed.

In figure 11 we can see some frames of the experiment ‘bounce of a ball on the ground’. For each frame the center of gravity of the tennis ball is obtained.
7 CONCLUSIONS AND FUTURE WORK

In section 6.1 (figures 5 and 6), we can see the quality of the new filter presented (FKF) which shows good behaviour for smooth and discontinuous motions. The object’s position is estimated even when it is inside the image plane and when it is outside the image plane. Therefore, combine classic filters (KF) when inside and steady-state filters ($\alpha\beta/\alpha\beta\gamma$) when outside.

We have compared our filter with $\alpha\beta$, $\alpha\beta\gamma$, $K_v$, $K_a$ and $K_j$ in experiments of pure prediction. We have compared too, our filter with Circular Prediction Algorithm (CPA) in this paper reproducing the same experiment as (Chroust and Vincze, 2003) for a direct comparison with the work done by Chroust and Vincze. The filter proposed works very well but not better than SKF for all conditions, therefore, the addition of a AKF action can improve the filter behaviour (future work).

The FKF is evaluated with a ramp-like and sinusoidal motions. $\Delta x_p$ is reduced in all tests done and the overshoot is decreased significantly. Results presented in this paper are obtained for $C(z) = K_p$. Other controllers like PD, PID, ... will be implemented in future work.

ACKNOWLEDGEMENTS

This work is supported by the Plan Nacional de I+D+I 2004-2007, DPI2005-08203-C02-02 of the Spanish Government (Técnicas Avanzadas de Teleoperación y Realimentación Sensorial Aplicadas a la Cirugía Asistida por Robots).

REFERENCES


