International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 5 Issue 7, July-2016

Fuzzy Logic and ANFIS based Image Centroid Tracking and Filtering Schemes

Parimala, P.[1], Raol, J. R.[2]

[1]Asstt. Prof., Dept. of Telecommunication, MSRIT; Jain University Research Scholar, Bangalore.
[2]Prof. Emeritus, MSRIT, Bangalore.

ABSTRACT

Two fuzzy logic based image-centroid tracking algorithms are presented. Fuzzy logic finds numerous applications in several signal/image processing, control, and sensor data fusion applications, and it is also one of the ingredients for building practical artificial intelligence (AI) systems. These centroid tracking algorithms (CTA) are based on: a) adaptive neuro fuzzy inference system (ANFIS), and b) fuzzy logic (FL)-function process; and both use Kalman filter (KF) for tracking. These centroid tracking algorithms are evaluated using MATLAB based simulated synthetic image data, and although the performance of both the CTAs has been found to be very satisfactory, the FL based CTKF performed better than the ANFIS based CTKF.

Key words: Image centroid tracking, fuzzy logic, ANFIS, Kalman filter

1. INTRODUCTION

In certain image-based air traffic control and/or air defense system (ATC/ADS), an automatic target acquisition, identification and tracking by processing a sequence of real images of the moving object or target are very essential. Many such and similar applications require an algorithm for image detection, segmentation, feature computation, selection, classification and tracking. In tracking of a moving target using image data that involves image processing, at each time sampling time, the estimates of the target's current position and velocity are obtained. There would also be an uncertainty present regarding the origin of the received/measured data, which may or may not include measurements from the target of interest. The latter might be due to a random clutter (called false alarm), and this leads to the data association (DA) situation and requirement. The tracking algorithms need to include information on detection and false alarm probabilities. Then, the main focus is on the implementation and validation of the filtering algorithm/s for precision tracking with segmentation of the images. The main characteristics of the image, in centroid tracking, are the intensity and size of the cluster. The pixel intensity is graded/discretized into certain layers of gray level intensities. The centroid tracking algorithm (CTA) involves: i) conversion of the data from the original image into a binary image by applying upper and lower bounds/limits for the target layers, ii) the binary target image is converted into clusters using nearest neighbor (NN) criterion; if the target-image size is known, this information is used to set limits for removing those clusters that differ sufficiently from the size of the target cluster, and iii) the centroid of the clusters is then calculated and this information is used for tracking the target-image.

There has been some work in the area of centroid tacking [1-6]. In Ref. [1] by analyzing the molten pool infrared images, the keyhole was extracted by using the fixed threshold method. Using the keyhole images and calculating the keyhole centroid, the deviations between the keyhole centroid and the welding seam was analyzed. The approach of Ref. [2] is highly statistical. The Ref. [3] had proposed a Kalman filter based centroid tracking algorithm, which is in fact further extended here in the fuzzy logic setting. In [4] the weld pool image centroid algorithm for seam tracking in arc welding process was proposed. In [5], the tracking problem is addressed using deformable models, and the experimental results with traffic sequences are given. In [6], the fuzzy system is designed for the purpose of tracking and predicting the motion of light-colored objects on dark background.

In this paper the application of fuzzy logic is studied and two centroid tracking algorithms based on: a) ANFIS, and b) FL based function process are presented; and the tracking algorithms are evaluated using MATLAB based simulated data and the performance results are presented.

2. CENTROID TRACKING

First the process of segmentation, which is extracting object or particles of interest as precisely as possible from the image, is needed that decomposes the image into different regions: i) say micro images and each region is defined by a set of feature characteristics, and b) in particle segmentation, an image is partitioned into object regions and background regions. In FL-CTA the latter is used to separate the target from the background (image) when target is not fully visible. The pixel intensities are graded/discretized into 256 gray levels, the segmentation is carried out as: a) the gray level image is converted into binary image (using lower and upper bounds of the target, the thresholds being determined using the pixel intensity histograms from the target and its environment); the gray image $\mathrm{Im}(i, j)$ is converted into binary image with intensity $\beta(i, j)$ [3]:

$$\beta(i, j) = \begin{cases} 1 & I_L \le \mathrm{Im}(i, j) \le I_U \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In (1) I_L and I_U are the lower and upper limits of the target intensity. The detection probability of the pixel is defined as:

$$\begin{aligned} P\{\beta(i, j) = 1\} &= p(i, j) \\ P\{\beta(i, j) = 0\} &= 1 - p(i, j) \end{aligned} \tag{2}$$

The gray image $I(i,j)$ is assumed to have a Gaussian distribution with mean μ and variance σ^2.

b) the detected pixels are then grouped into clusters, the binary image can be grouped into clusters using the nearest neighbor data association (NNDA), a pixel belongs to the cluster only if the distance between this pixel and at least one other pixel of the cluster is less than the distance d_p, this distance given as

$$\sqrt{\frac{1}{p_t}} < d_p < \sqrt{\frac{1}{p_v}} \tag{3}$$

In (3), p_t and p_v are the detection probabilities of target and noise pixels respectively, and d_p affects the size, shape and number of clusters.

The centroid of the cluster is determined as [3] $$(x_c, y_c) = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}} \left(\sum_{i=1}^{n}\sum_{j=1}^{m} iI(i, j), \sum_{i=1}^{n}\sum_{j=1}^{m} jI(i, j) \right) \tag{4}$$

In (4), the array $[x_c, y_c]$ is the centroid of the cluster, I_{ij} is the intensity of the $(i, j)^{th}$ pixel; n, and m are the dimensions of the cluster.

FUZZY LOGIC BASED CENTROID TRACKING SCHEMES

We present and evaluate two CT algorithms that are based on fuzzy logic and Kalman filter (KF), the latter is used in the nearest neighbor data association mode.

A brief note on fuzzy logic

FL is a multi-valued logic, e.g. a trapezoidal function leading to a trapezoidal form of the membership function (MF). There are several other types of membership functions (MFs). What a MF really does is to give a value between zero and one to the variable (on x-axis). This value is the gradation, or the MF grad value/s. Thus, the x-axis variable, say pressure, is now fuzzified, and its belongingness to a given fuzzy set (low, medium, or high temperature) would be any value as per the chosen MF. The fuzzy inference system (FIS) operates over the MFs and the (fuzzy) rule-base with the help of fuzzy implication functions (FIFs), see Figure 1. Then, via the defuzzification process FL system (FLS) gives crisp outputs that are further used for the intended purpose. These If…Then… rules are obtained, primarily from the human experts who have gathered a lot of practical and intuitive experience of operation, analysis and design of such systems. FL directly can be incorporated in target-image-centroid tracking algorithm (CTA) to deal with vagueness in image representation. Any such development of a FLS would require to: i) select fuzzy sets and their appropriate MFs-a fuzzification process, ii) create rule base, with the help of human/design experts for inputs-output mapping (I/O), iii) select suitable fuzzy operators, iv) select FIF and aggregation methods, and v) select an appropriate defuzzification method, see Figure 1. A FS allows a partial membership of a member of a set, a FS A on a universe of discourse (UOD) U with elements u is expressed as

$$A = \int \{\mu_A(u)/u\} \qquad \forall\ u \in U \tag{5}$$

$$\text{or } A = \sum \{\mu_A(u)/u\} \qquad \forall\ u \in U \tag{6}$$

Here, $\mu_A(u)$ is a MF function (value) of u on the set A and provides a mapping of the UOD U the closed interval [0, 1]. The fuzzy variables take on different labels defined by linguistic values such as Very Low, Low, Medium, Normal, High, and Very high, etc., with each represented by different MFs. As is studied in the classical logic and Boolean algebra, for the most elementary crisp set the basic operations/operators are the intersection AND, the union OR, and the complement NOT (in computer logic/Boolean logic). Since, in the FL the variable u is fuzzified, and the MFGs will vary with a grade between 0 and 1, the (classical-logic) operators/operations AND, OR, and NOT now have some new meanings, and there could be more than one definition for these. For FL the corresponding (to AND, OR, NOT), the operators specified are min, max, and complement and are defined as

$$\mu_{A\cap B}(u) = min[\mu_A(u), \mu_B(u)] \qquad \text{(intersection)} \tag{7}$$

$$\mu_{A\cup B}(u) = max[\mu_A(u), \mu_B(u)] \qquad \text{(union)} \tag{8}$$

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \qquad \text{(complement)} \tag{9}$$

For FL, first the rules are defined or constructed specifically for the given application; these rules can be learned from human experts (depending on the situation) or can be devised from the data of the system that is under study. A fuzzy rule is 'If u is A, Then v is B'. The main aspect in the use of FL is FIS that via FIF defines mapping from input FSs into output FSs. It is also possible that one or more rules may fire at the same time, in such a case, outputs of all rules are then aggregated, i.e. FSs that represent the output of each rule are combined into single FS. The fuzzifier/fuzzification maps input values into corresponding memberships via MFs; the fuzzifier-MF takes input values and determines the degree to which these numbers belong to each of FSs.

The steps in FL/FIS are: a) fuzzify the inputs u, and v using MF ($\mu^i(u)$, and $\mu^i(v)$) for ith rule, → it means that appropriate MFs are specified and used; b) since antecedent (If…) part of every rule has more than one clause, FL operator is used to resolve the antecedent to a single number between 0 and 1; c) use FIF to shape the consequent part (Then…), the output FS, based on the antecedent; d) since more than one rule, i.e. more than one output FS, can be fired at a time, it is essential to combine the corresponding output FSs into single composite FS. This is known as aggregation, Σ, Figure 1; and e) In order to get crisp value of output variable w, defuzzification step is carried out.

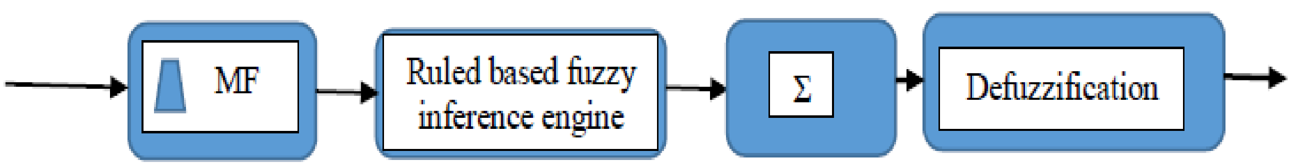

Figure 1 Fuzzy inference system (FIS)

Adaptive neuro-fuzzy inference system-ANFIS

The ANFIS system can be easily incorporated into any filtering algorithm so that FL can be used in augmentation for improving the performance of thus combined system, this artefact of ANFIS can be used for image centroid tracking and sensor data fusion also. ANFIS utilizes the rule based process to represent the system/data behavior in absence of a precise model of the system. From a collection of I/O data sets we can build a fuzzy inference model/system (FIS) that would approximate these data as well as use the rule base and output the final result, see Figure 2.

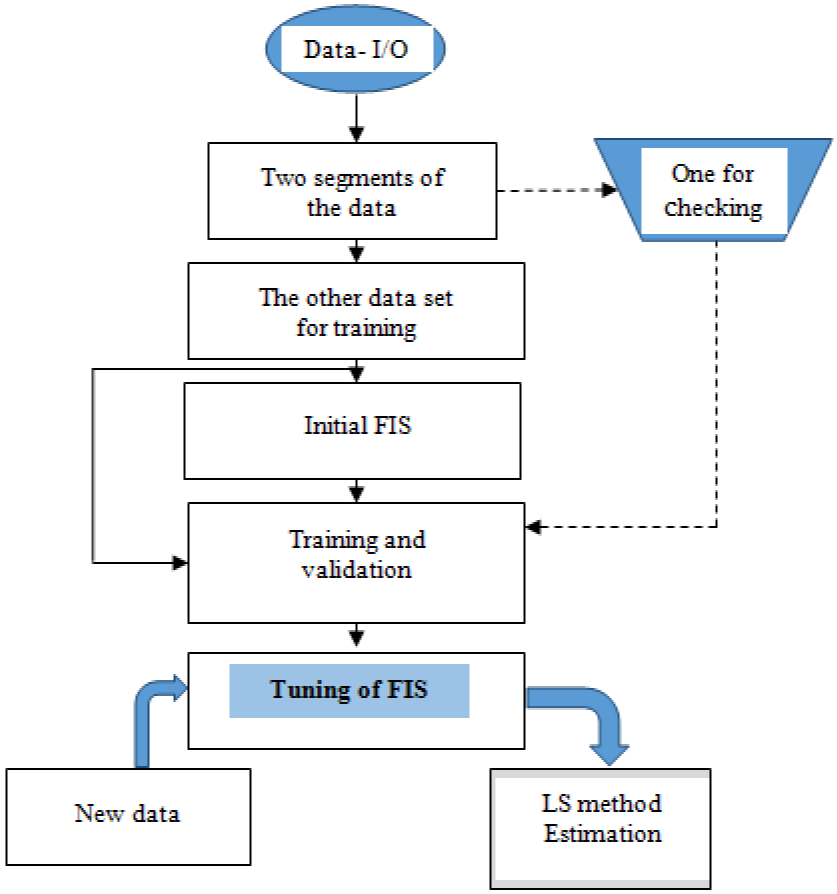

Figure 2 Basic ANFIS procedure (for image centroid tracking)

This type of system then should consist of some MFs and If…Then… rules with adjustable parameters (that define the MF), and these parameters can be chosen so as to adapt the MFs to the input data. This means that the MFs (i.e. the parameters that specify the structure/shape of the MFs) are now adaptive to the variations in the I/O data-sets. For this purpose the neural network based adaptive learning mechanisms can provide for the fuzzy-modelling procedure to learn information from these data sets.

Basically this will facilitate computation of suitable MFs and their defining parameters and constants that would allow the associated FIS to track the given I/O data. This leads to the ANFIS which is a class of the adaptive networks that are functionally equivalent to FIS. It uses a hybrid learning method to determine the parameters. It also uses the I/O data to determine the MFs' parameters, and consists of FIS-these parameters are tuned using either a BP steepest descent (back-propagation) algorithm or a combination BP and LS method. These parameters are updated via the learning and iterative cycles that are facilitated by a gradient vector (of the chosen cost function). Thus, the MFs are adaptively tuned and determined by using ANN and the I/O data of a given system in an iterative manner.

The major components of the ANFIS are: i) If u_1 is A_1 and u_2 is B_1, Then $y_1=c_{11}u_1 + c_{12}u_2 + c_{10}$; b) If u_1 is A_2 and u_2 is B_2, Then $y_2=c_{21}u_1 + c_{22}u_2 + c_{20}$; with u_1, u_2 as non-fuzzy inputs, and y as the desired output; c) some internal computations using product operators/normalization, etc.; d) computation of output, $\overline{w}_i = \dfrac{w_i}{w_1 + w_2}$,

$\overline{w}_i y_i = \overline{w}_i (c_{i1}u_1 + c_{i2}u_2 + c_{i0})$, $i = 1,2.$, and $y_p = \overline{w}_1 y_1 + \overline{w}_2 y_2$, where y_p is the predicted output. The output, linear in the consequent parameters is written as

$$\begin{aligned} y_p &= \overline{w}_1 y_1 + \overline{w}_2 y_2 = \overline{w}_1 (c_{11}u_1 + c_{12}u_2 + c_{10}) + \overline{w}_2 (c_{21}u_1 + c_{22}u_2 + c_{20}) \\ &= (\overline{w}_1 u_1)c_{11} + (\overline{w}_1 u_2)c_{12} + \overline{w}_1 c_{10} + (\overline{w}_2 u_1)c_{21} + (\overline{w}_2 u_2)c_{22} + \overline{w}_2 c_{20} \end{aligned} \tag{10}$$

Then, a hybrid training-cum-estimation algorithm adjusts the consequent parameters in a forward pass and the premise parameters are updated in the backward pass. The MATLAB-ANFIS programming steps and system are used for centroid tracking in conjunction with KF.

Fuzzy logic based KF for centroid tracking

Since, KF is used as a basic tracking algorithm its equations are given as follows.

Centroid-state and state-error covariance time propagation:

$$\tilde{X}_{(k+1)} = \Phi \hat{X}_{(k)} \tag{11}$$

$$\tilde{P}_{(k+1)} = \Phi \hat{P}_{(k)} \Phi^T + GQG^T \tag{12}$$

Centroid-state and state-error covariance update:

$$K_{(k+1)} = \tilde{P}_{(k+1)} H^T \left[H \tilde{P}_{(k+1)} H^T + R \right]^{-1} \tag{13}$$

$$\hat{X}_{(k+1)} = \tilde{X}_{(k+1)} + K_{(k+1)} \left[Z_{(k+1)} - H \tilde{X}_{(k+1)} \right] \tag{14}$$

$$\hat{P}_{(k+1)} = \left[I - K_{(k+1)} H \right] \tilde{P}_{(k+1)} \tag{15}$$

The centroid coordinates are represented as a state X in the KF.

FL can also be used for tuning a Kalman filter, and centroid tracking algorithm can be developed by a combination of FL and KF. In such systems the FL can be considered as aiding soft decision-making in the filtering process because of the use of fuzzy 'IF… Then' rules in making some judgment on the use of, say residuals, in navigating the filtering in the direction of achieving accurate results in tracking process. For the purpose of centroid tracking FL is combined with KF filter at the measurement update level. The equations for the FLKF are the same as those of KF except equation (14) [7]:

$$\hat{X}(k+1) = \tilde{X}(k+1) + K f_{FL}(k+1) \tag{16}$$

Here, f(k+1)is regarded as an output of the FL based process and is, in general, a nonlinear function of inputs as the innovations 'e' (and derivative of e) of the KF. It is assumed that position-in x-y axes-measurements of the target are available. The f-vector consists of the modified innovation sequence for x and y axes

$$f_{FL}(k+1) = \left[f_{FLx}(k+1) \quad f_{FLy}(k+1) \right] \tag{17}$$

To determine (17), the innovation vector e is first separated into its x and y components, e_x and e_y, with the target motion in each axis assumed to be independent. The f-vector for the x direction can be developed and then it is generalized to include y direction. This vector consists of two inputs, e_x and $\dot{e}_x$, and single output $f_{FLx}(k+1)$, where $\dot{e}_x$ is computed by

$$\dot{e}_x = \frac{\{e_x(k+1) - e_x(k)\}}{T} \tag{18}$$

Here, T is the sampling interval in seconds, and the expression (18) is extended to y-direction (and even z-direction if required). Appropriate FL based MFs are defined and the technique is evaluated using MATLAB based fuzzy logic tool box.

EVALUATION OF THE CTA ALGORITHMS

A model of FLIR (forward looking infrared sensor) for generation of synthetic image is considered here. To simulate the motion of the target in the frame, kinematic models of target motion are used. The state model used to describe the constant velocity target motion is given by

$$X(k+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X(k) + \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} w(k) \tag{19}$$

In (19), $X(k) = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T$ is the state vector, and $w(k)$ is zero mean white Gaussian noise with co-variance matrix $Q = \begin{bmatrix} \sigma_{w1}^2 & 0 \\ 0 & \sigma_{w2}^2 \end{bmatrix}$. The measurement model is given as

$$z(k+1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X(k+1) + v(k+1) \tag{20}$$

Here, $v(k)$ is the centroid measurement noise, zero mean and Gaussian, with diagonal covariance matrix. A two-dimensional array of 64×64 pixels is considered for the background image, which is modeled (generated) as white Gaussian random field as $N(\mu_n, \sigma_n^2)$. The results here, are presented for the standard deviation of the background image as 50. Another two-dimensional array of pixels, which is modeled as white Gaussian random field as $N(\mu_t, \sigma_t^2)$ is used to generate a target of size (9x9). The total number of scans is 50 and image frame rate (T) is 1 frame/sec. The initial state vector of the target in the image frame is: $X = \begin{bmatrix} x & \dot{x} & y & \dot{y} \end{bmatrix}^T = \begin{bmatrix} 10 & 1 & 10 & 1 \end{bmatrix}^T$. The synthetic image with these parameters is converted into binary image using the upper, $I_U = 110$ and lower, $I_L = 90$ limits of a target layer and then grouped into clusters by the nearest neighbor data association method using the optimal proximity distance d_p =2.

The centroids of the clusters are computed. Since, the background is very noisy the cluster algorithm produces more clusters and more centroids. This requires NNKF to associate the true measurement to the target. The performance metrics: percentage fit errors (for the measurements and states), and rms (root mean square) values of position and velocity errors are evaluated. Also, the time histories of the state errors and measurements errors and KF residuals are evaluated to ascertain the performance of these centroid tracking algorithms. Table 1 shows the numerical values of these performance metrics for three values of the standard deviation of the additive white Gaussian noise in the target image itself, it is seen that the performance metrics of these tracking algorithms show somewhat upward trends with the increase in the standard deviation of the noise in the target images.

Table 1. Performance metrics for two FL based centroid tracking algorithms

Standard deviation of target noise →	Std=1	Std=3	Std=5
	ANFIS based centroid tracking algorithm		
% fit errors			
FE-x	0.859	0.917	1.304
FE-y	0.909	1.071	1.160
RMSPE	0.4795	0.5403	0.668
RMSVE	0.206	0.225	0.250
	FLKF centroid tracking algorithm		
FE-x	**0.5687**	**0.6308**	**1.061**
FE-y	**0.6484**	**0.7924**	**0.8664**
RMSPE	**0.3305**	**0.3881**	**0.5248**
RMSVE	**0.07454**	**0.07807**	**0.1223**

However, these metrics have lower values for the FLKF centroid tracking compared to the ANFIS based algorithm. Also, the ANFIS requires two step procedure: training, and then evaluation. For training of ANFIS, the error and error derivatives are used as input, and the target output is used as the output. Figures 3 and 4 show the time histories of state errors, and measurement residuals for the CTAFLKF. Since, these are within their theoretical boundaries, the performance of the centroid tracking filtering algorithms can be considered as very satisfactory. Similar plots were also obtained for the ANFIS based KF filtering tracking algorithm, but not produced here. Hence, it is seen that both the FL based centroid tracking algorithms perform very satisfactorily and hence, can be further utilized for image data fusion and building AI based sensor data fusion and target tracking systems.

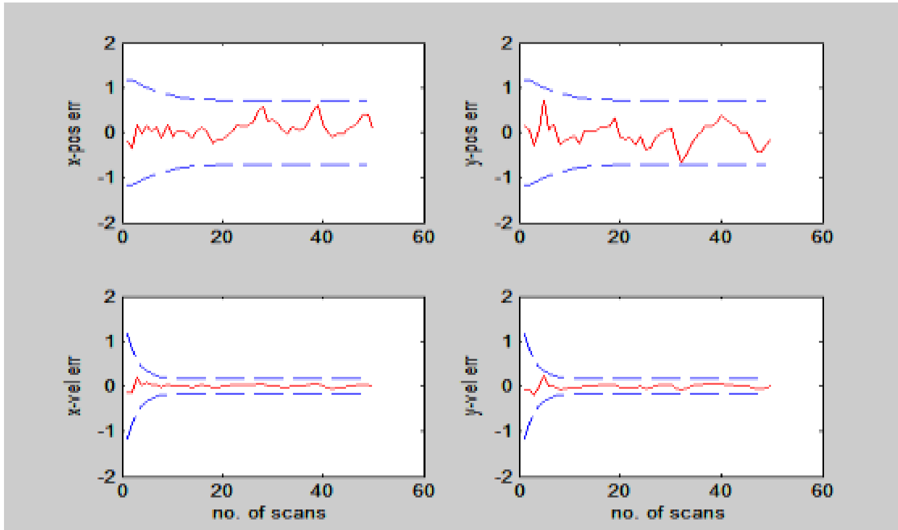

Figure 3 State errors with bounds: FLKF

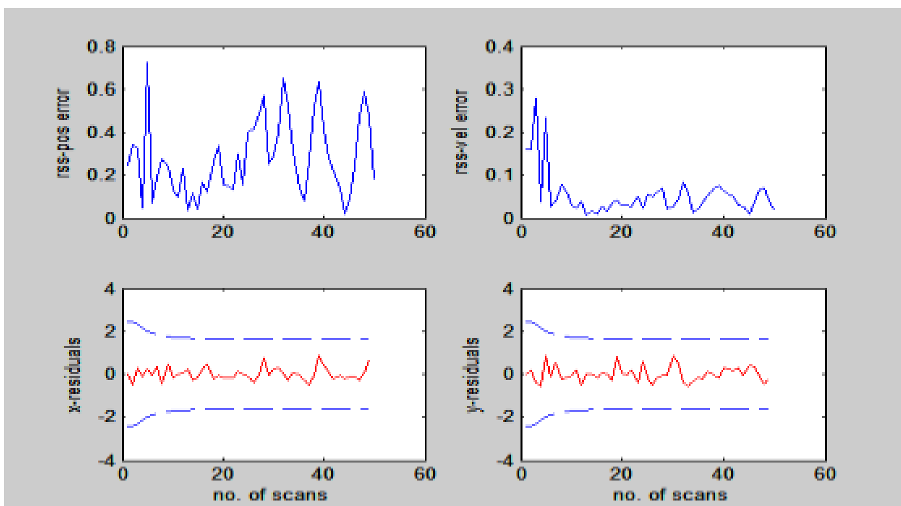

Figure 4 RSS-position & velocity errors and measurement residuals: FLKF

CONCLUDING REMARKS

Two fuzzy logic based centroid tracking algorithms have been presented and evaluated for their performance using synthetic images of a target in a noisy background field. The basic tracking algorithm used is the well-known Kalman filter in the NNDA setting. Although, the performance of the ANFIS based centroid tracking is slightly worse than that of the FL based KF (FLKF), the overall performances of both the algorithms is found to be very satisfactory based on the numerical values of the performances metrics and the behavior of the time histories of the errors, which are all found to be within their respective theoretical bounds. This development can be further extended to image fusion and building AI based systems for centroid tracking of moving objects. Also, these tracking algorithms can be implemented for real time applications and tested using FPGA (field programmable gate arrays) and DSP (digital signal processing) hardware.

REFERENCES

[1]. Chen, Y-Q, Gao, X-D, Huang, J-Y, and Wen Q. Numerical study of keyhole centroid infrared image based on fixed threshold method during high power fiber laser welding.

[2]. Cho, J-S., Kim, D-J., and Park, D-J. Robust centroid target tracker based on new distance features in cluttered image sequences. IEICE Trans. Inf. & Syst., Vol. E83-D, No. 12, Dec. 2000.

[3]. Naidu, V. P. S., Girija G., and J. R. Raol. Data fusion for identity estimation and tracking of centroid using imaging sensor data. Defense Science Journal, Vol. 57, No. 5, pp. 639-652, Sept., 2007.

[4]. Gao, X., Ding, D., Bai, T., and Katayama, S. Weld Pool Image Centroid algorithm for seam tracking in arc welding process. IEEE paper, published Proc. of International Workshop on Imaging Systems and Techniques, Shenzhen, China, May 11-12, 2009.

[5]. Nascimento, J. C., Abrantes, A. J., and Marques, J.S. An algorithm for centroid-based tracking of moving objects. IEEE International Conference on Acoustic, Speech, and Signal Processing, Vol. 6, pp. 3305-3308, 15-19, March, Phoenix, AZ., USA, 1999.

[6]. Rodjito, P. Position tracking and motion prediction using fuzzy logic. Dept. of Computer Science, Honors Theses. Paper 520. http://digitalcommons.colby.edu/honorstheses/520.

[7]. Raol, J. R. Multisensor Data Fusion with MATLAB. CRC Press, Florida, USA, 2010.