Global Journal of
Information Technology

Volume 05, Issue 1, (2015) 25-40

www.awer-center/gjit

Moving target trajectory estimation using Kalman, curve fitting and Anfis methods

Timur İnan *, Instructor at Istanbul Arel University, Kucukcekmece, Istanbul 34295, Turkey.

Suggested Citation:
İnan, T. (2015). Moving target trajectory estimation using kalman, curve fitting and anfis methods. *Global Journal of Information Technology*. *5*(1), 25-40.

Received 18 January, 2015; revised 16 March, 2015; accepted 27 March, 2015.
Selection and peer review under responsibility of Prof. Dr. Adem Karahoca, Bahcesehir University, Turkey

Abstract

Estimation of the possible position of the moving targets after a few steps has great significance especially in terms of defense industry. If a shoot aiming at the target is planned, the issue of estimation of forward position of the target gains importance in terms of accurate strike of the bullet at the target. In target tracking, impact of three different methods as motion estimation method on various motion types has been examined in our study. Motion types have been examined in four different types, which are rectilinear motion, circular motion, sinusoidal motion and curvilinear motion. On the other hand, estimation methods have been examined under three different titles. These are Kalman estimation method, curve fitting method and Anfis method. Different motion types have been examined with different estimation methods and the results obtained have been presented.

Keywords: robotics, image processing, trajectory tracking, target tracking, estimation, Anfis, Kalman, curve fitting.

*ADDRESS FOR CORRESPONDENCE: **Timur İnan,** Instructor at Istanbul Arel University, Kucukcekmece, Istanbul 34295, Turkey. *E-mail address*: timurinan@arel.edu.tr / Tel.: +90-537-669-6627

1. Introduction

This study aims at introducing a system, which can define the target, track the target, designate its possible location in the course of time using various estimation methods; testing motion methods in motion types using this system, and presenting the estimation types giving more favorable or more adverse results as far as the motion types are concerned.

The system has three main parts which are target tracking system, pan-tilt camera and microcontroller to place the camera to the center of the target.

The robot, performing the tracking work, is a robot known as biaxial pan-tilt camera; the robot consists of two servo motors, a camera and connection plates. Thanks to the image processing software, running on the computer, operations such as target designation, positioning, and tracking are carried out. MATLAB software has been used for the purpose of image processing. In order to ensure communication between the image processing software and the robot, a microcontroller takes place in between as shown in Figure1.

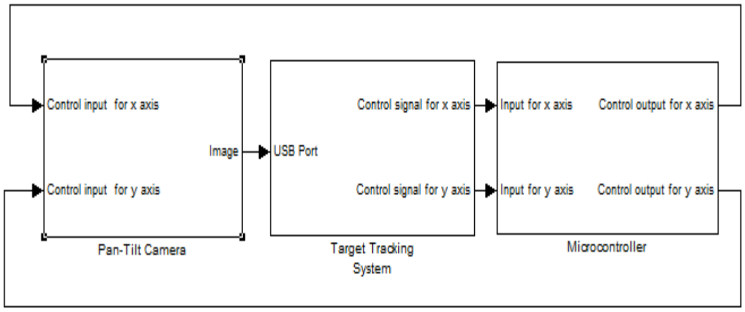


Figure 1. System components.

The platform uses the positioning information, obtained from the computer, to position the servomotors; thereby accurate operation of the system and positioning of the camera at the midpoint of the target are possible.

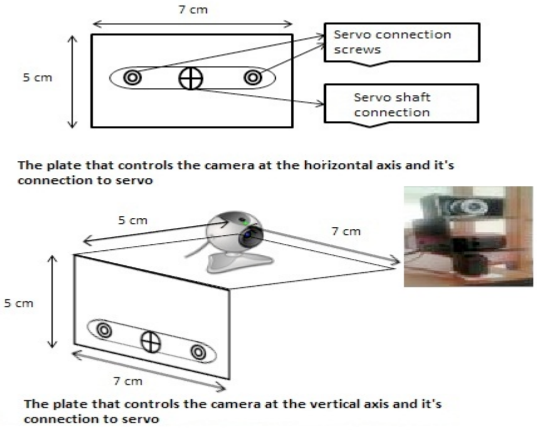

7 cm
5 cm
Servo connection screws
Servo shaft connection
The plate that controls the camera at the horizontal axis and it's connection to servo
5 cm
7 cm
5 cm
7 cm
The plate that controls the camera at the vertical axis and it's connection to servo

Figure 2. Servo connection plates

The camera at the top of the platform is connected to the computer via usb port, system gets the image from the camera via usb port.

As seen of Figure2, the servo motors on the platform are used for positioning the plates to position the camera to the center of the target definition on the last frame. A microcontroller for positioning the camera is connected to the computer via another usb port; the image processing software computes the right rotation amounts by calculating denavit-hertenberg parameters [9].

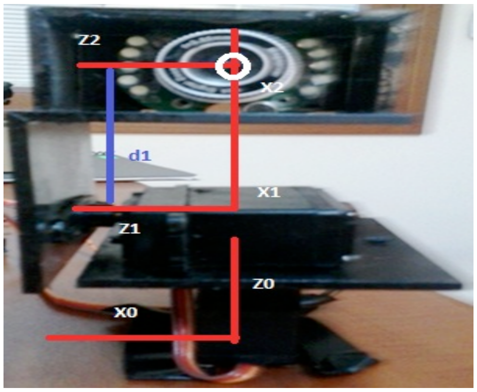


Figure 3. Joints of pan-tilt camera

For controlling the pan-tilt camera, rotation for horizontal and vertical axis must be mathematically described.

Table 1. Denavit-Hertenberg parameters of pan-tilt camera

i	α_i	a_1	d_i	θ_i
1	π/2	0	d_1	q_1
2	0	0	0	q_2

$$ {}^{0}_{2}x = {}^{0}_{1}R(q_1)\,{}^{1}_{2}R(q_2)\begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}\cos q_1 & 0 & \sin q_1\\ \sin q_1 & 0 & \cos q_1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\cos q_2 & \sin q_2 & 0\\ \sin q_2 & \cos q_2 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}cosq_1 * cosq_2\\ sin\,q_1 * cosq_2\\ sinq_2\end{bmatrix} \quad (1)$$

$$ {}^{0}_{2}y = {}^{0}_{1}R(q_1)\,{}^{1}_{2}R(q_2)\begin{bmatrix}1\\0\\0\end{bmatrix} = \begin{bmatrix}\cos q_1 & 0 & \sin q_1\\ \sin q_1 & 0 & \cos q_1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\cos q_2 & \sin q_2 & 0\\ \sin q_2 & \cos q_2 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\1\\0\end{bmatrix} = \begin{bmatrix}cosq_1 * sinq_2\\ sinq_1 * sinq_2\\ cosq_2\end{bmatrix} \quad (2)$$

$$ {}^{0}_{2}z = {}^{0}_{1}R(q_1)\,{}^{1}_{2}R(q_2)\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}\cos q_1 & 0 & \sin q_1\\ \sin q_1 & 0 & \cos q_1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\cos q_2 & \sin q_2 & 0\\ \sin q_2 & \cos q_2 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} = \begin{bmatrix}sinq_1\\ cos\,q_2\\ 0\end{bmatrix} (3)$$

For any 3D position a point can be described as;

$$ {}^{0}_{T}p = ({}^{0}_{T}x \;\; {}^{0}_{T}y \; {}^{0}_{T}z\,)^{\mathrm{T}} \quad (4)$$

$$ \alpha = atan2\{{}^{0}_{T}y, {}^{0}_{T}x\,\} \quad (5)$$

$$ \beta = atan2\{{}^{0}_{T}z\,, \sqrt{{}^{0}_{T}x^2 + {}^{0}_{T}y^2}\} \; (6)$$

Where α is the angle for horizontal axis and β is the angle for vertical axis.

For horizontal axis; $\mathrm{pwm_{on}} = 1000\mu s + \alpha * \left(\frac{1000\,\mu s}{180^{\circ}}\right), pwm_{off} = 20000\mu s - \alpha * (\frac{1000\mu s}{180^{\circ}})$ (7)

For vertical axis; $\mathrm{pwm_{on}} = 1000\mu s + \beta * \left(\frac{1000\mu s}{180^{\circ}}\right), pwm_{off} = 20000\mu s - \beta * (\frac{1000\,\mu s}{180^{\circ}})$ (8)

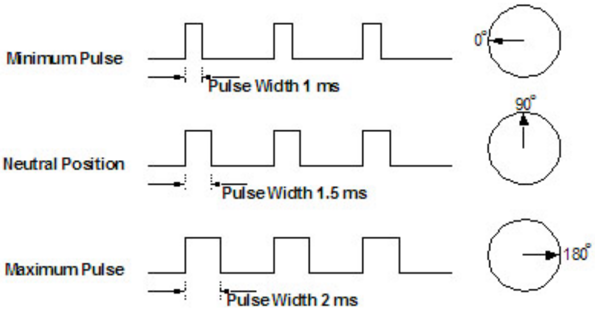


Figure 4. Pulse width modulation controlling of servo [12].

For placing the servos to the right angle for accurate tracking of the target, pulse width modulation is used. Servos place their shafts between 0 and 180 degrees according to the width of the pulse, this pulse is 1ms for 0° and 2ms for 180°, so each angle is 1ms/180. Microcontroller takes the angle information from the tracking system and generates a pulse width modulated wave to place the servos.

For identification of the target, two-dimensional normalized correlation method is used. Using photos of the target taken from various angles, the target can be identified. Photos of the target are used as templates to calculate normalized cross correlation.

Images taken from the camera are cross correlated with the templates which are the target's photos taken before.

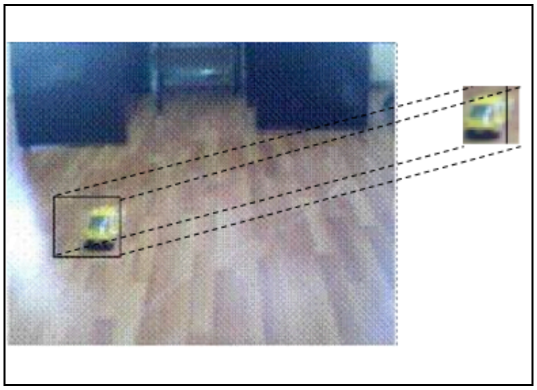


Figure 5. System tracking the target

The frames taken from the camera is formatted in YUY2160X120 format, target images which are used templates are different sized images, but as they are templates, they are small sized images.

If we consider the incoming frame as a 160x120 matrix and name it A and if we name the template image as B, if we define M_a as the image row number, N_a as the image row number, M_b as the template column number, N_b as the template row number, the calculation can be defined at the equation 9.

$$C(i,j) = \sum_{m=0}^{M_a-1} \sum_{n=0}^{N_a-1} A(m,n).conj(B(m+i,n+j)) \qquad (9)$$

Where 0≤i<Ma+Mb-1 and 0≤j<Na+Nb-1. [13]

For every template image, the template and incoming frame from the camera is cross correlated and if the correlation coefficient is satisfactory, the target is identified in the image.

The image taken from camera and templates are normalized as the equation 10 shows.

$$c(u,v) = \frac{\sum_{x,y}[f(x,y)-\bar{f}_{u,\mu}][t(x-u,y-v)-\bar{f}]}{\left\{\sum_{x,y}\left[[f(x,y)-\bar{f}_{u,\mu}]^2\right]\sum_{x,y}[t(x-u,y-v)-\bar{f}]^2\right\}^{0.5}} \qquad (10)$$

Where, f is the image, $\bar{t}$ is the mean of the template, $\bar{f}_{u,\mu}$ and is the mean of f (x, y) in the region under the template. c is the correlation coefficients [11, 12].

While finding the satisfactory correlation coefficient, a floating number is used which is obtained by the user from the system interface. If the number is low, the sensivity of the target definition is low, if the number is high, then the sensivity of the target definition is high.

As a result of designation and tracking of the target, the locations of the target in the course of time are recorded by the tracking system. The information recorded is shown to the user as a graphic later on. The graphical user interface of the system can be seen on Figure 6.

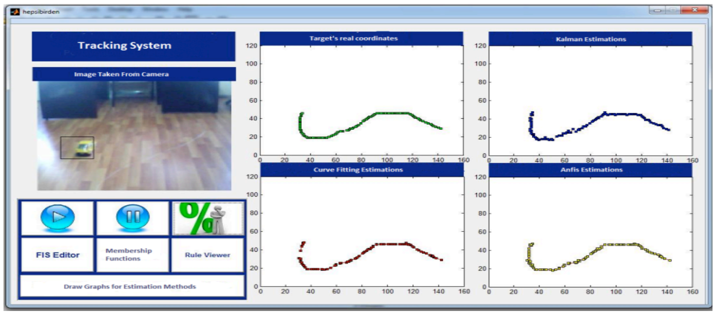


Figure 6. User interface of the system

2. Estimation methods used

As target estimation method; Kalman estimation, curve fitting and Anfis methods are examined and what kind of results various motion types will yield with particular estimation methods are stated. The motion estimation methods used are as below:

2.1. Kalman estimation method

Kalman estimation method is an estimation method, which is used in many areas, including image processing. Kalman estimation method can produce estimations for the next step of the system using system state-space model, operation and measurement noise covariance information [7].

Kalman method becomes capable of estimating the next position of the system with the help of the inputs given, state-space and noise levels.

2.2. Curve fitting method

Curve fitting methods suggest many different methods in terms of possible next location of the target. Use of these methods changes depending on the shape of the curve, drawn by the orbit of the target. Use of the accurate method in position estimation with curve fitting method is pretty significant in terms of obtainment of an accurate result [8].

In this study four different moving types examined, for all motion types polynomial function is used as the fitting function.

2.3. Anfis method

Fuzzy logic method is a method suggested by taking inspiration from decision making mechanism of human beings. Artificial neural network is a method which can teach machines how to behave in certain situations inspiring from human learning system and human brain working system.

Anfis is a method which uses both fuzzy logic and artificial neural network to get good sides of these methods.

In terms of our study, the possible next position of the target is unclear. The issue of solution of this uncertainty with a decision system is an important topic in terms of operation of many other systems as in our system [7].

At this point, one of the methods, which can be used in terms of how uncertainty will be estimated in decision processes and can be made a part of the decision processes in general manner, is Anfis method [2, 3].

The steps, which have to be applied while building up an Anfis application, are as below:

- Establishment of input or inputs; output or outputs.
- Establishment of training datasets.
- Establishment of learning datasets.
- Trying Anfis system.
- Applying the system to the real world.

4. Motion analysis for estimation methods

For all the estimation methods used, to calculate the estimated coordinates, the laws of motion equations are used.

$$x = x_0 + v_x dt \qquad (11)$$

$$y = y_0 + v_y dt \qquad (12)$$

$$v_x = v_{x_0} + a_x dt \qquad (13)$$

$$v_y = v_{y_0} + a_y dt \qquad (14)$$

Where,

a_x is the acceleration on x axis, a_y is the acceleration on y axis.

4.1. Motion analysis for kalman method

Kalman estimation method is a method that is used in many applications like image processing, signal processing and control systems.

For examining the motion, first the motion must be mathematically described, so the state space model of the motion must be defined. For defining the state space model a 6x6 matrix created. Because the variable number of the motion equation as described in (9), (10), (11) and (12) are 6, and if the relation between the variables defined in the same equations, the state space model of the motion can be defined [4].

$$A = \begin{bmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (15)$$

Where, dt=1,

Kalman method has two steps while computing the estimated coordinates, first is computing the predicted state and covariance. The second is computing the estimated state using the estimated coordinates and the measurement values.

For computing the predicted state and covariance two matrices used, first is a matrix that holds predicted coordinates, velocities and accelerations is a six rows and one column matrix;

$$\text{predicted state} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (16)$$

The matrix holds zero in all fields for the initial condition and the matrix is updated by Kalman filter in every iteration.

The second matrix is a six rows and six columns matrix that holds prediction covariance,

$$\text{prediction covariance} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

After computing the predicted state and covariance, Kalman uses them to compute the estimated coordinates. For computing estimated coordinates Kalman method needs to know the current measurement which is a two rows and one column matrix,

$$\text{current measurement} = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix} \quad (18)$$

Current measurement values come from the target tracking system.

Kalman uses current measurement, prediction covariance, predicted state and Kalman gain, measurement matrix to compute the estimated coordinates which ise a two rows and one column matrix,

$$\mathrm{es} = \mathrm{ps} + \mathrm{kg} * (\mathrm{m} - \mathrm{mm} * \mathrm{ps}) \quad (19)$$

Where es is estimated state,

ps is predicted state,kg is Kalman gain, m is measurement, mm is measurement matrix.

mm is the measurement matrix which used to find how similar the current measurement with the predicted state is two rows and six columns matrix,

$$\text{measurement matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

Kalman gain is a six rows and two columns matrix used for computing estimated state.

$$\text{kalman gain} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (21)$$

$$kalman\ gain = \left(\frac{\mathrm{mm} * \mathrm{pc}' * \mathrm{mm}'}{\mathrm{mm} * \mathrm{pc}'}\right)' \quad (22)$$

Where pc’ is transpose of prediction covariance, mm’ is transpose of measurement matrix, mm is measurement matrix [1].

4.2. Motion analysis for curve fitting method

For examining motion for curve fitting not like the other estimation methods, for preparing a curve data, satisfactory times of motion coordinates must be logged. For this study, after the system start, the system logs 5 coordinates for preparing a curve data for creating the motion curve equation. After 5 times of logging the system starts estimating by the equation of curve fitting part of the system. And while logging new data the system renews the equation of curve fitting by the new acquired coordinates [6].

X coordinates and y coordinates of the target is logged and the last 5 coordinates are imported to the xData and yData matrices;

$$\mathrm{xData} = ([\mathrm{lxp}_{a-4} \quad \mathrm{lxp}_{a-3} \quad \mathrm{lxp}_{a-2} \quad \mathrm{lxp}_{a-1} \quad \mathrm{lxp}_{a}]) \qquad (23)$$

$$\mathrm{yData} = ([\mathrm{lyp}_{a-4} \quad \mathrm{lyp}_{a-3} \quad \mathrm{lyp}_{a-2} \quad \mathrm{lyp}_{a-1} \quad \mathrm{lyp}_{a}]) \qquad (24)$$

Where lxp is logged x points of the target, lyp is the logged y points of the target and a is the initial iteration number.

For calculating the fit function MATLAB’s fit function is used. Fit function requires xData, yData, fitting type and options parameters.

```
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Normalize = 'on';
[fitresult, gof] = fit( xData, yData, ft, opts );
```

Fitting type is the parameter which obtains the type of fitting while computing the fitting function.

For this study fitting type ‘poly1’ is selected because in the experiments this type got the most accurate results.

4.3. Motion analysis for anfis method

For examining motion for Anfis method, first the motion must be trained to the system for all possible x and y coordinates and also for different speeds and acceleration. In Figure 5, the target and it’s possible next points can be seen [3].

Where ever the target is positioned, according to the calculation of the target’s maximum speed and the frame acquiring time of the system the target can move to 20 pixels right, left , up or down so the training data can be created by accepting the center of the target (0,0) then defining all possible coordinates, -20 pixel to 20 pixel x and y side of the target. This means 41 pixels for x and 41 pixels for y side.

For calculating estimated coordinates by Anfis method, input for fuzzy inference system must be obtained then the output of fuzzy inference system can be calculated. The inputs for fuzzy inference system are the possible coordinates of the target beginning from (0, 0) coordinates.

```
for $a_x$=-30,-19,…..,30
    for $a_y$=-30,-19,…….,30
        Log the x,y, $a_x$ and $a_y$ values,
        Calculate possible x and y coordinates,
    end inner loop
end outer loop
```

Where, a_x is the acceleration on x axis, a_y is the acceleration on y axis, x,y, a_x and a_y values are stored in a matrix which has 3721 rows and 4 columns. This matrix used for generating training data. The matrix is called allpossiblecoordinates.

$$\text{allpossiblecoordinates} = \begin{bmatrix} x & y & a_x & a_y \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \qquad (25)$$

All possible next coordinates are calculated by using a state space matrix which has four rows and four columns,

$$\text{state space matrix} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (26)$$

In every iteration x and y coordinates, a_x a_y values are assigned into a multiplicity matrix which is used to calculate estimated coordinates with state space matrix, the multiplicity matrix is a four rows and one column matrix,

$$\text{multiplicity matrix} = \begin{bmatrix} \text{X point} \\ \text{Y point} \\ a_x \\ a_y \end{bmatrix} (27)$$

$$ec = ssm * mm \qquad (28)$$

Where ec is estimated coordinates, ssm is state space matrix and mm is multiplicity matrix.

After finding every estimated coordinates for every position and for every acceleration types, for using Anfis method, input and output for fuzzy inference system are determined.

To obtain input and output for fuzzy inference system, MATLAB's genfis1, Anfis and evalfis methods are used. Genfis1 method creates input for fuzzy inference system, Anfis method creates output for fuzzy inference system using estimated coordinates and input for fuzzy inference system.

$$training\ data = [apc(:,3:4), ec(:,3)] \quad (29)$$

Where apc is all possible coordinates and ec is estimated coordinates.

For generating input for fuzzy inference system, genfis1 method uses training data, number of membership functions and membership type, for this study number of membership functions is selected 20. Membership function type is selected 'gbell' function.

$$infis = genfis1(trndata, numMfs, MfType) \quad (30)$$

Where numMfs is number of membership functions, MfType is Membership function type.

Then output of fuzzy inference system is generated by using Anfis method. Anfis method uses training data, input of fuzzy inference system and member ship numbers.

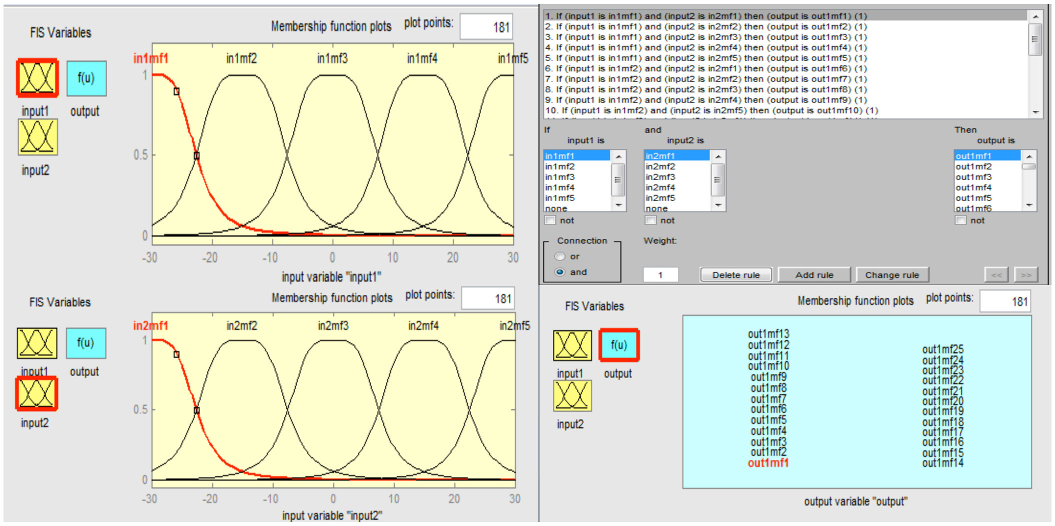
$$\text{outfis} = \text{anfis}(\text{trndata}, \text{infis}, \text{numMfs}) \quad (31)$$

After calculating input and outputs of the system for x coordinates, the same process is applied for y coordinates [5].

$$\text{training data} = [\text{apc}(:,3:4), \text{ec}(:,4)] \quad (32)$$

$$\text{infis2} = \text{genfis1}(\text{trndata}, \text{numMfs}, \text{MfType}) \quad (33)$$

$$\text{outfis2} = \text{anfis}(\text{trndata}, \text{infis}, \text{numMfs}) \quad (34)$$



1. If (input1 is in1mf1) and (input2 is in2mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf1) and (input2 is in2mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf1) and (input2 is in2mf3) then (output is out1mf3) (1)
4. If (input1 is in1mf1) and (input2 is in2mf4) then (output is out1mf4) (1)
5. If (input1 is in1mf1) and (input2 is in2mf5) then (output is out1mf5) (1)
6. If (input1 is in1mf2) and (input2 is in2mf1) then (output is out1mf6) (1)
7. If (input1 is in1mf2) and (input2 is in2mf2) then (output is out1mf7) (1)
8. If (input1 is in1mf2) and (input2 is in2mf3) then (output is out1mf8) (1)
9. If (input1 is in1mf2) and (input2 is in2mf4) then (output is out1mf9) (1)
10. If (input1 is in1mf2) and (input2 is in2mf5) then (output is out1mf10) (1)

If input1 is: in1mf1, in1mf2, in1mf3, in1mf4, in1mf5, none; □ not

and input2 is: in2mf1, in2mf2, in2mf3, in2mf4, in2mf5, none; □ not

Then output is: out1mf1, out1mf2, out1mf3, out1mf4, out1mf5, out1mf6; □ not

out1mf13, out1mf12, out1mf11, out1mf10, out1mf9, out1mf8, out1mf7, out1mf6, out1mf5, out1mf4, out1mf3, out1mf2, out1mf1

out1mf25, out1mf24, out1mf23, out1mf22, out1mf21, out1mf20, out1mf19, out1mf18, out1mf17, out1mf16, out1mf15, out1mf14

Figure 7. Membership functions for inputs and output

5. Experimental results

The fact that the motion types have been examined under four main headings in our study had been mentioned in part one. These motion types have been established as rectilinear motion, curvilinear motion, circular motion and sinusoidal motion. How estimation methods respond to various motion types of the system, and whether the estimation method is proper or improper for particular motion types are given in this part in the light of experimental data.

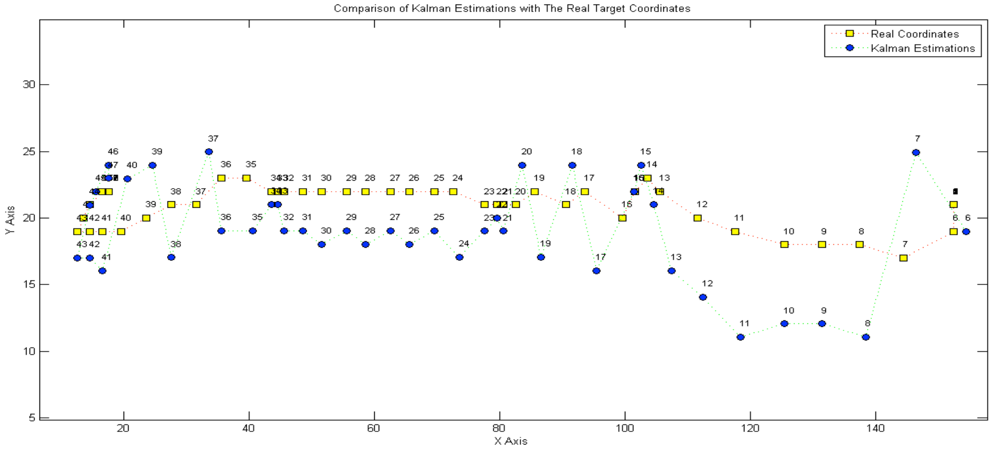


Figure 8. Examination results of kalman for rectilinear motion

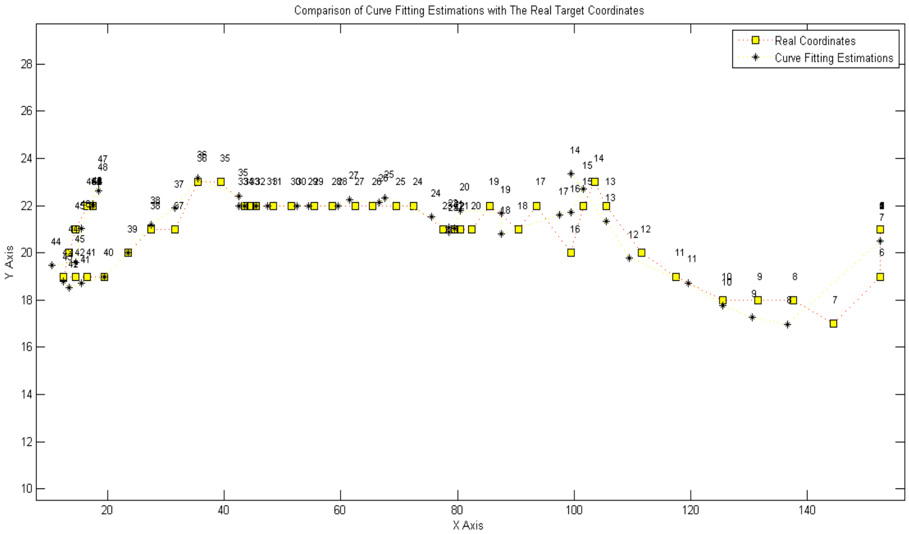


Figure 9. Examination result of curve fitting for rectilinear motion

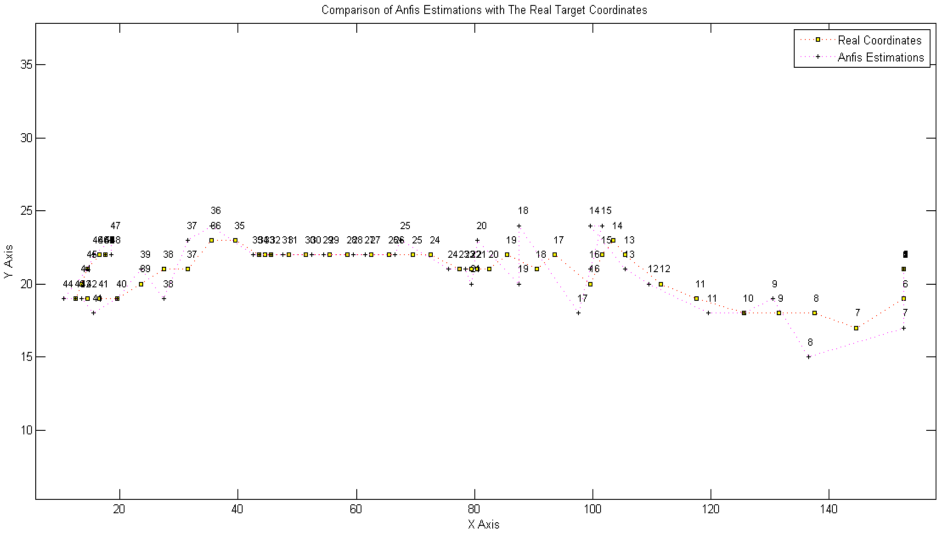


Figure 10. Examination result of anfis for rectilinear motion

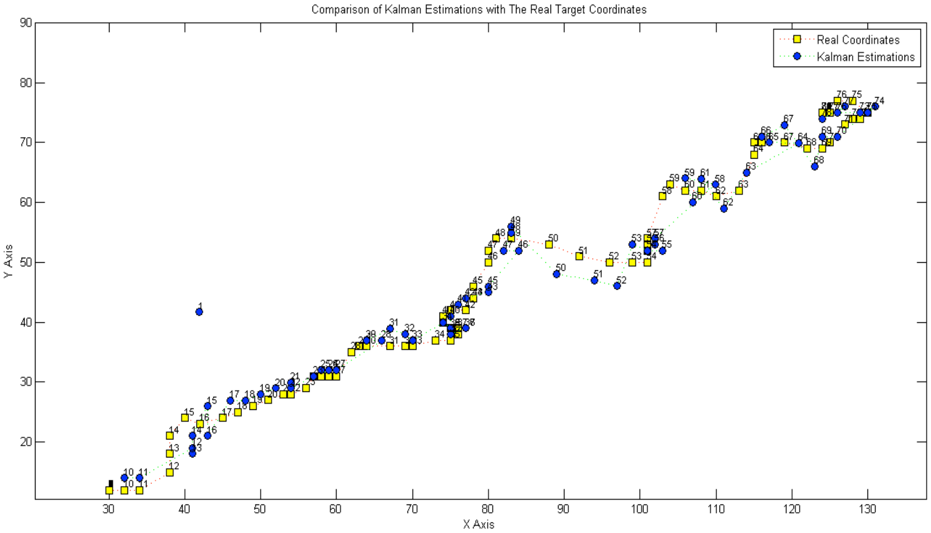


Figure 11. Examination result of kalman for curvilinear motion

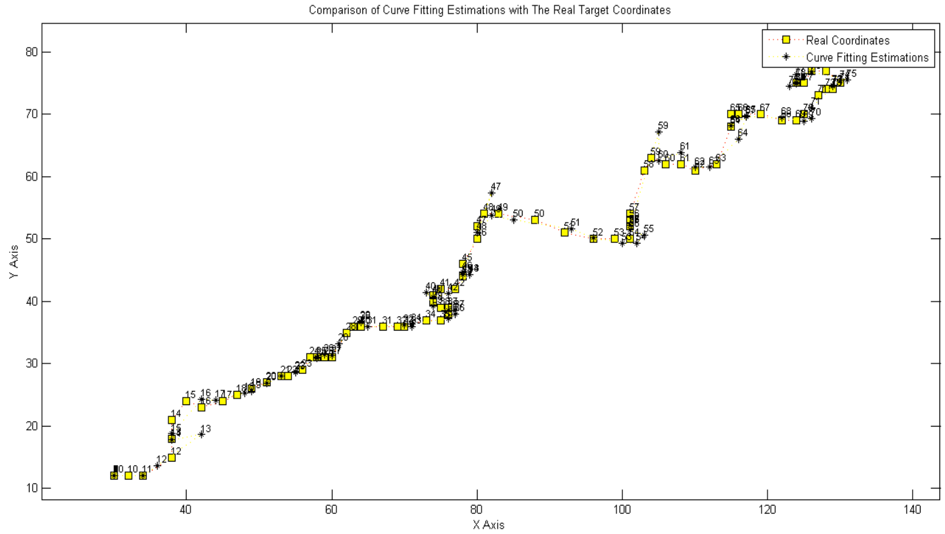


Figure 12. Examination result of curve fitting for curvilinear motion

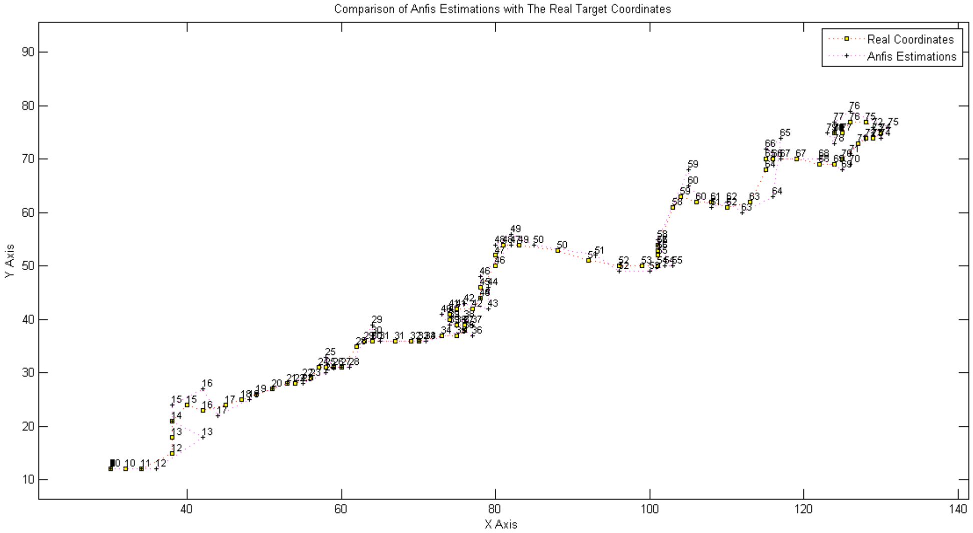


Figure 13. Examination result of anfis for curvilinear motion

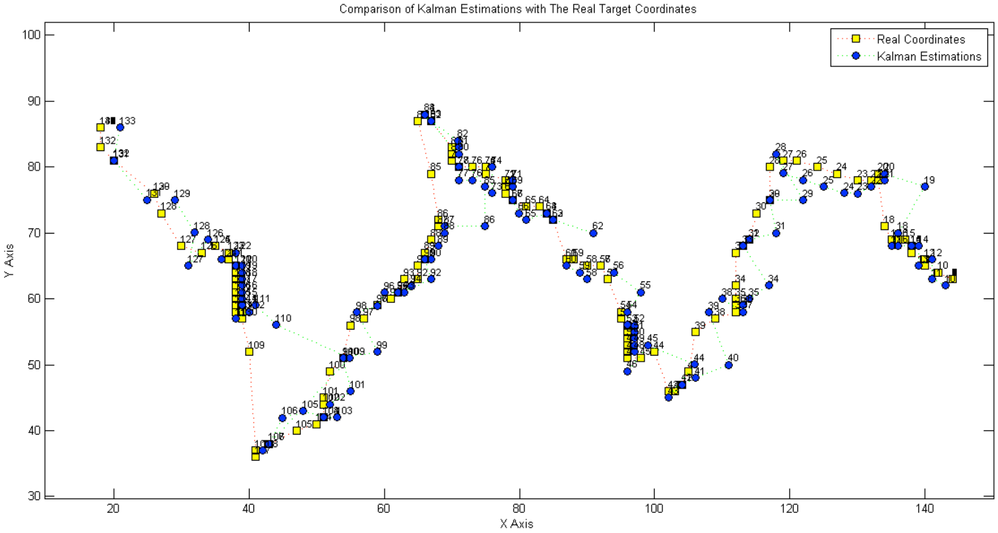


Figure 14. Examination result of kalman for sinusoidal motion

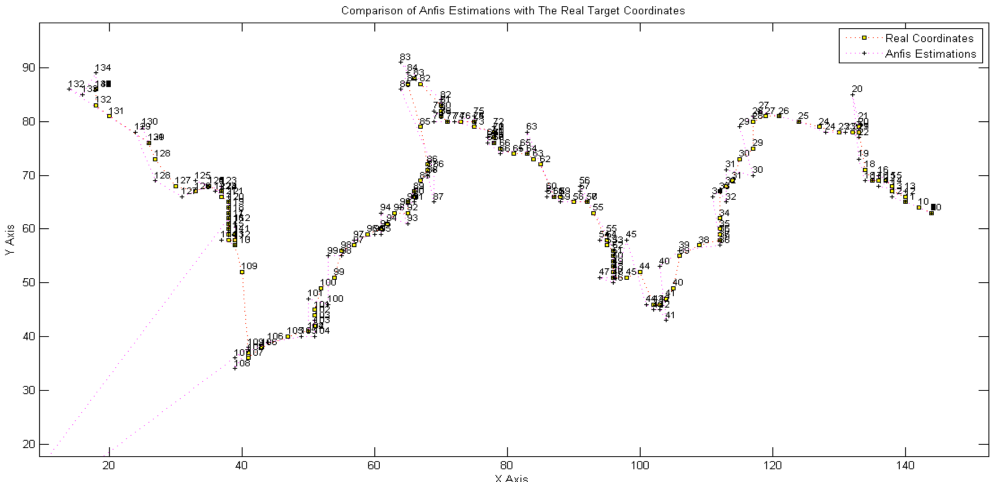


Figure 15. Examination of anfis for sinusoidal motion

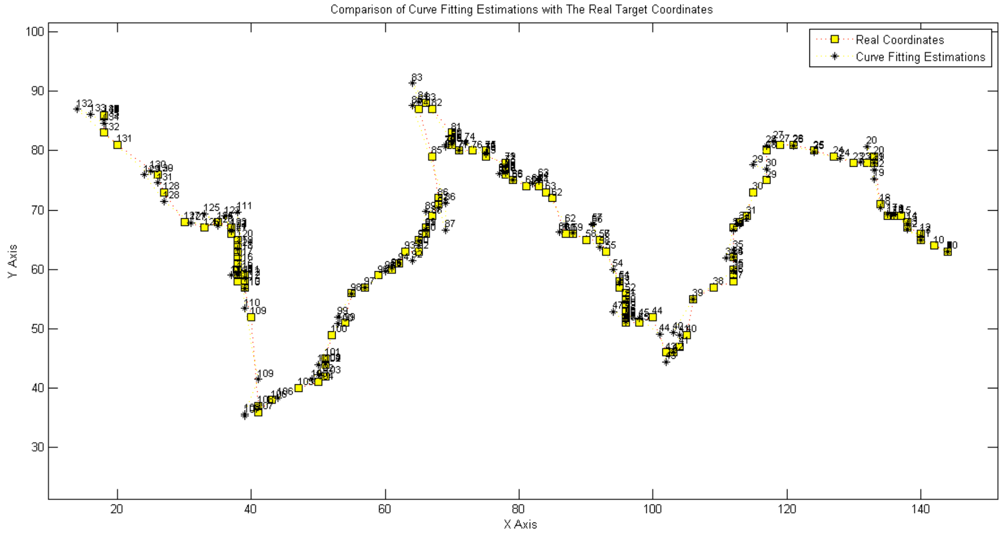


Figure 16. Examination result of curve fitting for sinusoidal motion

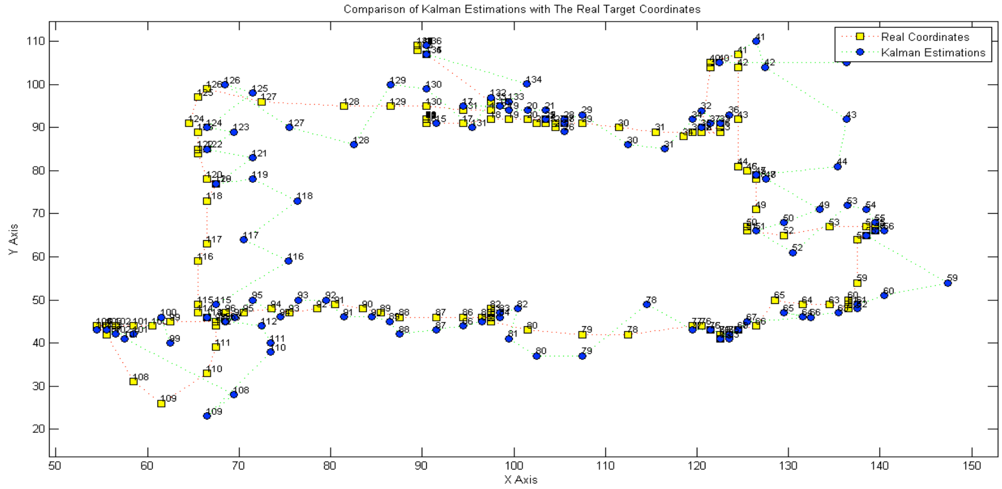


Figure 17. Examination result of kalman for circular motion

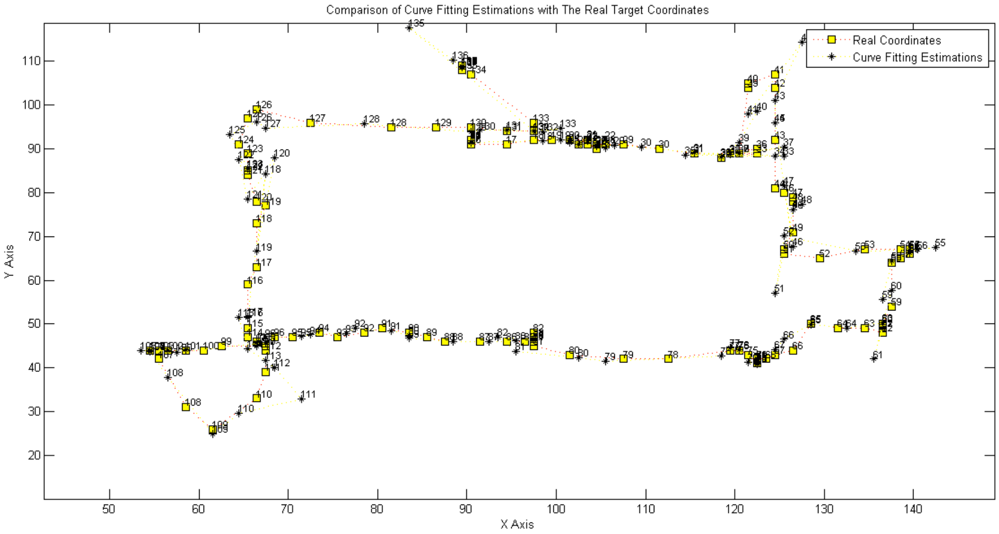


Figure 18. Examination result of curve fitting for circular motion

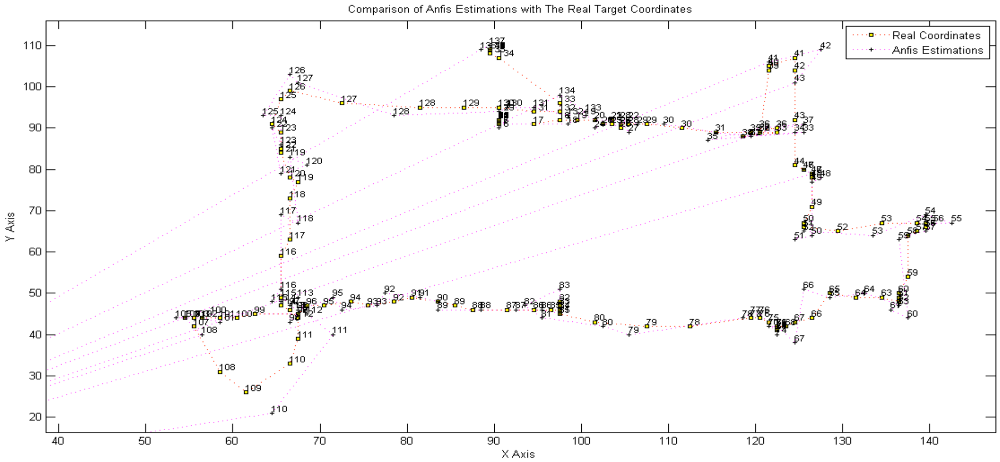


Figure 19. Examination result of anfis for circular motion

5.1. Comparison of the methods

Totally for three estimation method four different type of motion has been tried in real time and all the experiments have been recorded, for examining how the estimation of the motion is like the real motion, mean error and exact estimation number of the methods have been calculated and noted.

For obtaining the accuracy of the methods the mean error value of each estimation type for each motion type have been shown in graphical bars.

The mean error values of estimation types are shown over the graphs and exact estimations of the estimation types are shown over the graphs as well for comparing the accuracy of the methods for each motion types.

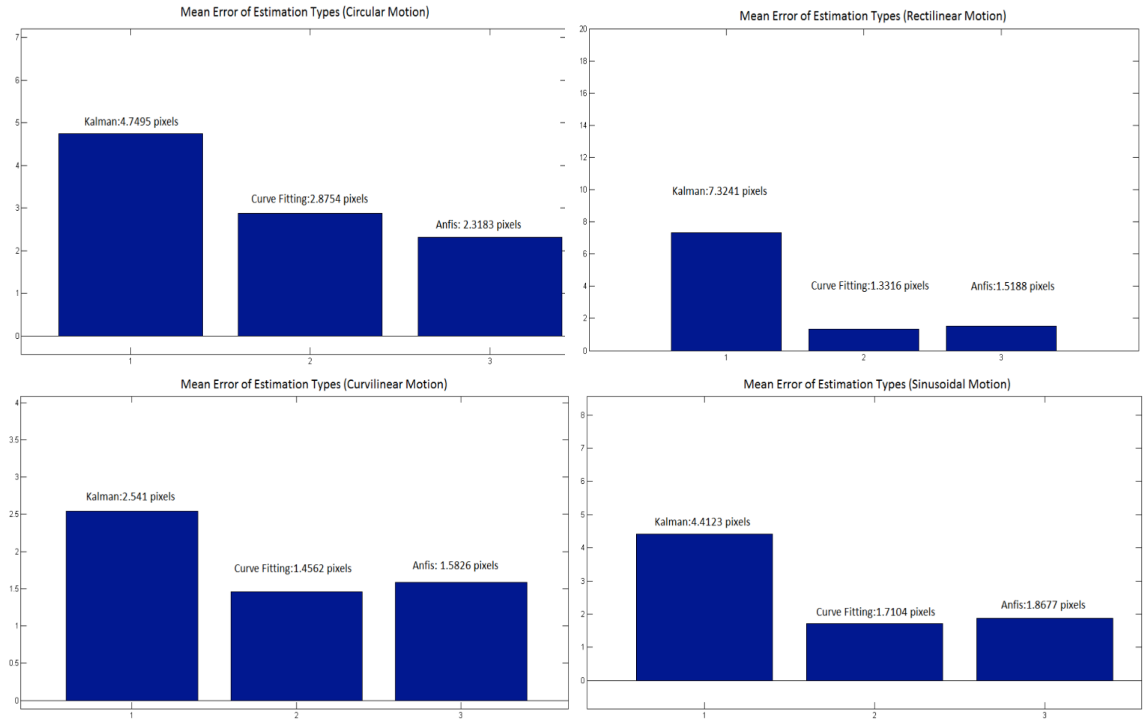


Figure 20. Mean error of estimation types for all motion types

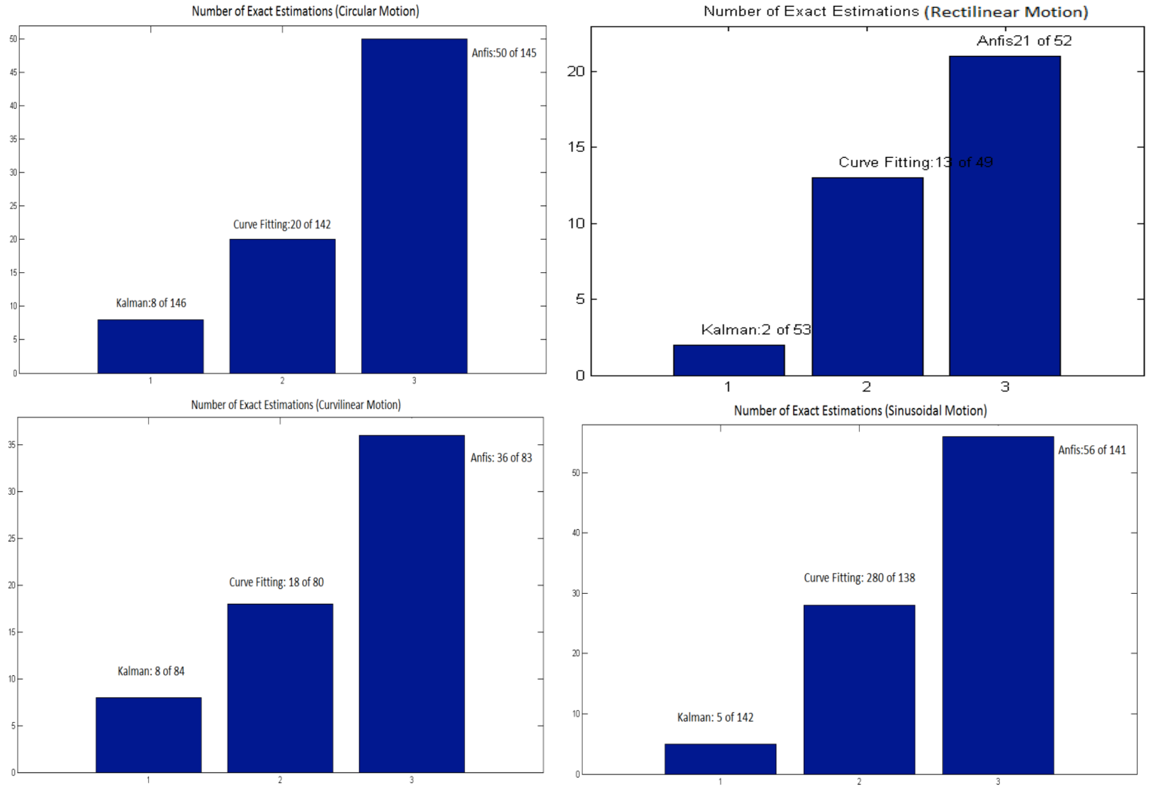


Figure 21. Number of exact estimations for all motion types

6. Conclusions

In this study, the results have been examined using three different motion estimation methods for four different motion types. As a result of the examination made, it has been found out that Anfis method gives better results when compared with the other methods. It has been observed that curve fitting estimation method gives less accurate results when compared with Anfis method; however it obtains a high success rate. Kalman method has been found out to be the least successful method when compared with the other methods.

As a result, this study can identify a target, track the target and can record the orbit of the target. With the help of the recorded data, the system infers from the coordinate data obtained, and starts to make estimations after five steps for curve fitting method; after two steps for Kalman and Anfis methods. The results obtained are satisfactory for all motion types and in terms of all estimation methods.

References

[1] Simon, H. (1996). Adaptive Filter Theory. Upper Saddle River, NJ: Prentice-Hall, Inc.

[2] Jang, J.S. R. (1991). Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm, Proc. of the Ninth National Conf. on Artificial Intelligence (AAAI-91), 762-767.

[3] Jang, J.S. R. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference Systems, *IEEE Transactions on Systems, Man, and Cybernetics, 23*(3), 665-685.

[4] MATLAB. / Kalman Prediction/Control System Toolbox/Functions/LQR-LQG Design/Kalman.

[5] MATLAB. / FuzzyLogicToolbox/Functions/Advanced Fuzzy Inference Systems/anfis.

[6] MATLAB. / CurveFittingToolbox/Functions/Fit Postprocessing/confint.

[7] Barimani, N., Moshiri B., & Teshnehlab M. (2012). State space modelling and short-term traffic speed prediction using kalman filter based on ANFIS. *IACSIT International journal of engineering and technology, 4*(2).

[8] Sillen, L. G. (1956). Some graphical methods for determining equilibrium constants II. On curve fitting methods for two-variable data. *Acta chemical scandinavica 10*. 186-202

[9] Corwin, Jr. et al. (1975). Method and apparatus for programming a computer operated robot arm. United States patent.

[10] Lewis, J. P., Fast Normalized Cross-Correlation, Industrial Light & Magic.

[11] MATLAB/Computer Vision System Toolbox.

[12]http://www.egr.msu.edu/classes/ece480/capstone/spring12/group06/Documents/Application%20Note%20-%20Eric%20Mitchell%20%28Servo%20Control%29.pdf?attredirects=0&d=1