The Nucleus, **41** (1-4) 2004 : 7-17

The Nucleus
A Quarterly Scientific Journal of
Pakistan Atomic Energy Commission
NCLEAM, ISSN 0029-5698

COMPARISON OF TECHNIQUES FOR PREDICTION OF FUTURE POSITIONS IN TRAJECTORY TRACKING OF AN OBJECT BY DC DRIVES

QAZI S.M. ZIA-UL-HAQUE and N.R. JADOON

KINPOE, Karachi Nuclear Power Complex, Paradise Point, Karachi, Pakistan

(Received February 21, 2003 and accepted in revised form November 23, 2004)

Trajectory tracking means to follow the path or trajectory of a moving object. Trajectory tracking finds application in various fields of war and peace. The location of the object can be represented by its rectangular as well as spherical coordinates. The system performing the task was a prototype model of an anti aircraft gun. To point the target by the gun we needed to track only the two spherical coordinates of its position i.e. the angle of azimuth and the vertical angle. To be in coherence with the object, the knowledge of future position was required in advance. However it was not possible to have this knowledge. Good estimates of the future positions could be made from the knowledge of motion so far of the object, thus a good estimation technique was required. The estimation techniques used here included conventional numerical techniques and modern adaptive filtering techniques as well. The paper is based only upon the results of estimation techniques applied. For discussion, only the results for the angle of azimuth are used in the paper. Conventional numerical techniques were found suitable when the object to be tracked moves with smaller degree of non-linearity in its motion and at the transients in terms of error magnitude but are poorer in terms of computing time and steady state error reduction. The adaptive filtering techniques on the other hand are poorer in transient error magnitude but are good in terms of computing time and steady state error reduction.

Keywords : Prediction, Motion control, Trajectory tracking, DC drives, Filters

1. Introduction

The project was to make use of computer to automize the trajectory tracking of moving objects. The moving object follows a trajectory that is not predefined for the system that is trying to track it. It needs a well-equipped data acquisition system for sensing its current position, a sophisticated algorithm to predict its future position and then a control mechanism to reach the predicted position in coherence with the object.

Trajectory tracking finds its application in so many areas. For example, to automize a movie camera to automatically focus some particular object needs to first recognize and secondly track the trajectory of the object. The second phase is one of the applications of the project. Another field of the project is air defense where to encounter the intruding enemy aircrafts, anti aircraft guns are used. To decrease the error and enhance the efficiency of these guns, they can be automized through computers which may control the gun to track the trajectory of the intruding air craft. Another area of the applications of the project is the modern robotic industry. An intelligent robot design may also need to sense some moving object, to know/understand its motion and to follow or catch the object. To design such a robot it requires including trajectory tracking of the object so as to provide the feature of following the moving object.

The task of trajectory tracking was accomplished by controlling the two angles θ (azimuth) and ϕ (vertical) of a model of anti-aircraft gun using digital PID (Proportional-Integral-Derivative) controllers. Tracking the two angles leads the model to focus the object during its motion.

The overall system used in the project was a sort of embedded control systems. Embedded control systems are one of the important areas in control engineering. Basically an embedded control system consists of three important components.

1. Master Controller, which in our system was a PC, which decides or receives the desired trajectory parameter and PID coefficients from user and supplies them to the slave controller.
2. Slave Controller, which was a control card consisting of two separate digital PID

[*] Corresponding author : smzhaq@yahoo.com

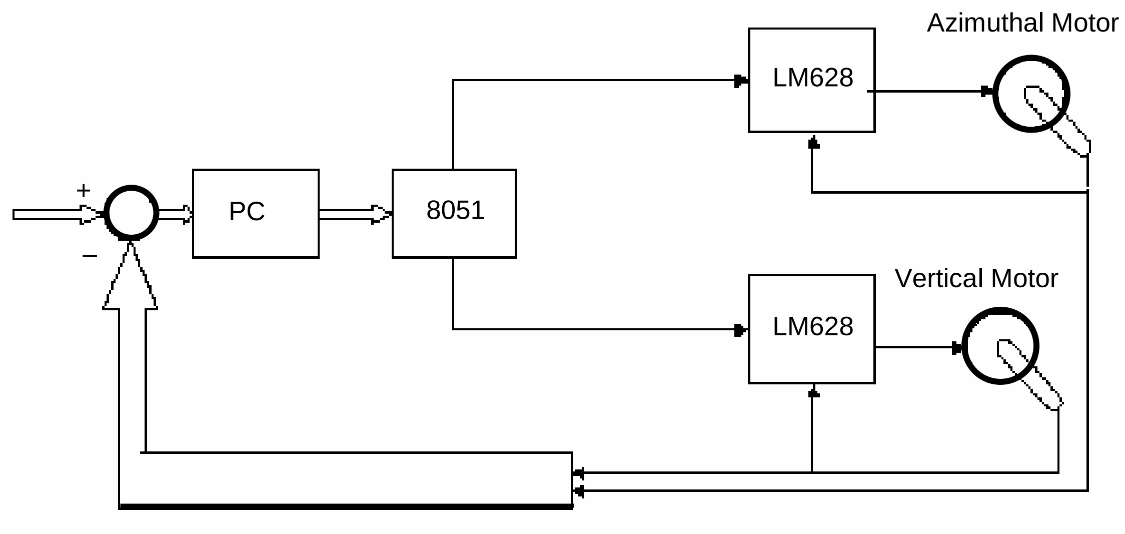

Figure 1. Block diagram of the system.

controllers for the two angles. The control card had a serial communication link with the PC.

3. Plant, a model consisting of two motors driving two separate gears to track the two angles θ (azimuth) and ϕ (vertical).

Figure 1 shows the block diagram of the complete system. There are three loops in the system. The first, the outer loop is the master loop taking in the input data points as the reference signal and the position as output. The output (position) is compared with the desired position and the error (desired position minus actual current position obtained by the block shown as Σ in figure 1) is provided to PC which generates the required velocity signal for next sample. The two inner loops on the other hand have required velocity as the reference signal from master controller (through the 8051 microcontroller) and have the actual velocity as the output. The sampling time in the outer loop is larger and is maintained to 1 second by MATLAB clock. The inner loops have a sampling time equal to 1.024 milliseconds maintained through operating frequency of LM628 precision motion controllers [6] provided by the quartz crystal.

2. Simulation of Moving Object

A problem with us was that we did not have any physical object available for the trajectory tracking so it was needed to be simulated through some mean. For this purpose the MATLAB function ODE45 was used. The function is very much useful to solve the state space for a particular time span with certain initial conditions. Since the moving target can be represented in state space with position, velocity and acceleration as the state variables, the function can be used to generate the position at specific times. Since for a physical object the motion is always continuous, the data points generated with the above method has been made to match closely to a physical object. The motion of a physical object may be with constant position, constant velocity or at most with constant acceleration. If at some instant the acceleration changes, it will most probably change at a constant rate and will remain constant for a certain period after change. Keeping the above facts in mind, a 4th order differential equation as

$$\left.\begin{array}{l} x_1' = x_2 \\ x_2' = x_3 \\ x_3' = x_4 \ \& \\ x_4' = C\,(\text{constant}) \end{array}\right\} \tag{1}$$

with the state variables

x_1 = position

x_2 = velocity

x_3 = acceleration &

x_4 = rate of change of acceleration

can represent a physical object. The state space can be represented as

$$\underline{X}' = A\,\underline{X} + \underline{B} \qquad (2)$$

Where,

$$\underline{X}' = \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \end{bmatrix}, \quad \underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \underline{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \& \quad \underline{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ C \end{bmatrix}$$

Equation 2 was defined in the function m file (standard type of files to write programs in MATLAB) to be used by the ODE45 a built in function of MATLAB to solve ordinary differential equation by Rangi Kuta method 45). Calling the ODE45 with this function m file as ODEFUN results in a matrix whose columns represent position, velocity, acceleration and its rate of change respectively. Choosing the first column of this matrix provided a sequence of data points for position. The approach was applied to the three axes (X, Y & Z) separately to generate the Cartesian coordinates of the position. The motion in any dimension could be simulated with constant position, velocity or acceleration by providing all the higher state variables (for example velocity, acceleration and rate of change of acceleration for constant position, or acceleration and its rate of change for constant velocity or rate of change of acceleration for constant acceleration) equal to zero initially. A change in the motion was produced at any time by first generating the data points upto that instant and then introducing the current states with the required change as initial conditions for next generation. The sequence Cartesian coordinates of points of trajectory were returned as x, y and z vectors. These were converted into spherical coordinates to get the angle of azimuth and the vertical angles θ & ϕ respectively, for tracking. Tracking the angles was performed in a loop repeatedly taking in the Cartesian coordinates of data points, calculating the θ & ϕ for current and future position and supplying the desired trajectory parameters to the slave controllers for tracking.

3. Theory of Operation

The main areas of the project include the estimation of position of object at the next sampling instant and tuning the digital PID filter. Tuning of PID filter is trivial and thus is omitted from the discussion. The paper emphasizes only upon the results obtained from the use of various estimation techniques.

We wished our system to track the object in coherence with it for which we needed the knowledge of the position of the moving object at the next sampling instant. Since we didn't have any future vision capability so that we may know the future position thus we had to apply some appropriate estimation technique to achieve close estimations for good results. As the estimates would be closed, the more accurate the tracking would be. For the estimation of future positions, we have used following two types of techniques in the project:

3.1. Curve fitting or polynomial approximation techniques

Curve fitting or polynomial approximation techniques are classical approaches for the estimation of the function value at different points where the actual value is unknown. This is made by approximating a polynomial on the known values of function. The curve fitting techniques are based upon the 'Weierstrass Approximation Theorem' [1]. According to the theorem

"If f is defined and continuous on *[a, b]*, and $\varepsilon > 0$ is given, then there exists a polynomial P, defined on *[a, b]* with the property that

$|f(x) - P(x)| < \varepsilon$ for all $x \in [a, b]$"

If the function value is approximated within the region [a, b], the method of approximation is called Interpolation and if the value is approximated at a point before a or beyond b, the approximation is called Extrapolation [1]**.** Since we always used the techniques for the time instants of future, we performed extrapolation**.** There are a number of methods for curve fitting. We have used the following in our project for estimation.

3.1.1. Lagrange polynomial

Lagrange polynomial uses the data points known to find the polynomial coefficients. Once the coefficients are estimated, the function value at any point can be approximated from the polynomial [1, 2].

For the implementation of Lagrange Polynomial Approximation, a function was developed in MATLAB. The function returns a polynomial of order provided as the input argument from the data points supplied. The coefficients of the polynomial are returned which can be used to approximate the function value at any point using *POLYVAL ( )*, the built in function of MATLAB.

3.1.2. *Least square estimation*

The Least Square Estimation method is usually implemented upon the data acquired by the sensors for their calibration. It is used to fit a best curve to the response of the sensors as the measurement values may have errors included in them. The criterion for the best fitting is *"minimization of the sum of the squares of the errors"*. In addition to giving a unique result for a given set of data, the least squares method is also in accord with the maximum likelihood principle in statistics. If the data is linear, the first-degree polynomial is sufficient to fit but in many cases the data obtained is non-linear so we need to fit them some function other than the first-degree polynomial. Because polynomials can be readily manipulated, fitting such functions to data that do not plot linearly is common [2].

To implement the least square estimation in the programming in MATLAB the built in function *POLYFIT( )* has been used which determines the coefficients of the required degree polynomial through Least Square Estimation. Once obtaining the coefficients, the function *POLYVAL( )* can be used to find the approximation of function at any point.

3.2. *Estimation using adaptive filtering techniques*

An adaptive filter has an *adaptation algorithm* that is meant to monitor the environment and vary the filter transfer function accordingly. The algorithm starts from a set of initial conditions, that may correspond to complete ignorance about the environment, and, based in the actual signals received, attempts to find the optimum filter design [3]. In a stationary environment, the filter is expected to converge to the Wiener filter. In a non-stationary environment, the filter is expected to track time variations and vary its filter coefficients accordingly. As a result, there is no such thing as a unique optimal solution to the adaptive filtering problem. Adaptive filters have to do without a priori statistical information, but instead usually have to draw all their information from only one given realization of the process, *i.e.* one sequence of time samples. Nevertheless, there are then many options as to what information is extracted, how it is gathered, and how it is used in the algorithm. In the stationary case, for example, ergodicity may be invoked to compute the signal statistics through time averaging. Time averaging obviously is no longer a useful tool in a non-stationary environment. Given only one realization of the process, the adaptation algorithm will have to operate with 'instantaneous' estimates of the signal statistics. Again, such estimates may be obtained in various ways. Therefore, we have a *'kit of tools'* rather than a unique solution. This results in a variety of algorithms. Each alternative algorithm offers desirable features of its own. A selection has been taken from the wealth of adaptive filtering algorithms developed in the literature.

A pragmatic choice is to use an *FIR filter* (Finite Impulse Response filter), where the filter output is formed as a linear combination of delayed input samples, *i.e.*

$$y_k = w_0 u_k + w_1 u_{k-1} + w_2 u_{k-2} + \cdot \cdot \cdot + w_{N-1} u_{k-N+1} \quad (3)$$

with y_k the filter output at time k, w_i, $i = 0\dots\dots\dots N-1$ the filter weights (to be 'adapted') and u_k the filter input at time k, as shown in figure 2.

This choice leads to tractable mathematics and fairly simple algorithms. In particular, the optimization problem can be made to have a cost function with a single turning point (unimodal). Thus any algorithm that locates a minimum, say, is guaranteed to have located the global minimum. Furthermore the resulting filter is unconditionally stable [3].

The generalization to adaptive *IIR* (Infinite Impulse Response) *filters* is nontrivial, for it leads to stability problems as well as non-unimodal optimization problems so we concentrated on adaptive FIR filter algorithms [3].

3.2.1. *Prediction*

In many applications, it is desirable to construct the desired signal from the filter input signal, for example consider

$$d_k = u_{k+1} \quad (4)$$

with d_k the desired signal at time k. This is referred to as the *forward linear prediction problem*, *i.e.* we try to predict one time step forward in time. When

$$d_k = u_{k-N}\dots \quad (5)$$

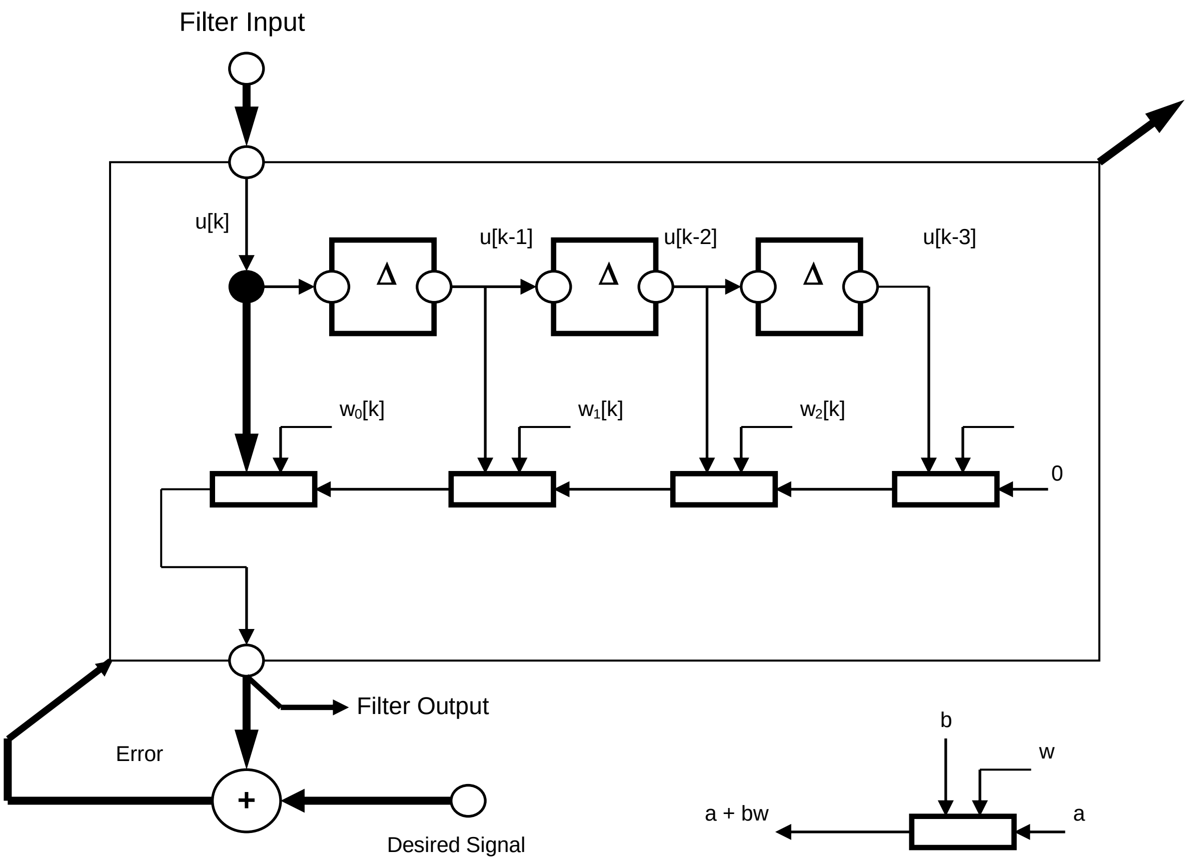

Figure 2. Prototype adaptive FIR filtering scheme (redrawn from [3]) .

we have the *backward linear prediction problem*. If e_k is the error signal at time *k*, one then has (for forward linear prediction)

$$e_k = u_{k+1} - w_0\ u_k - w_1 \cdot u_{k-1} - \cdot\cdot\cdot\cdot . w_{N-1} \cdot u_{k-N+1} \qquad (6)$$

or

$$u_{k+1} = e_k + w_0\ u_k + w_1\ u_{k-1} + \ldots\ldots + w_{N-1}\ u_{k-+1} \qquad (7)$$

This represents a so-called *autoregressive (AR) model* of the input signal u_{k+1} (similar equations hold for backward linear prediction) [3].

Figure 3 shows how the filter weights are updated. To predict the position at next sampling instant for our desired task of trajectory tracking, an estimate of the current value was estimated with the current weights and the positions acquired up to the previous instant (present position delayed one instant by delay element Δ). The estimate is then compared with the actual position acquired at the current instant that acts as the desired signal here. The error (found as actual current position minus the estimate by the block labeled Σ) was then used to modify the weights and the estimate for the position at next sampling instant was calculated with the modified weights and the positions acquired up to the current instant.

3.3. *The recursive least square (RLS) predictors*

RLS, typically exponentially weighted (EW-RLS), is a commonly considered adaptive Wiener technique. Adaptive prediction with RLS has been analyzed for zero bandwidth [4] and finite bandwidth [5] chirped signals. These previous results consider the performance of RLS for prediction and identification problems in non-stationary environments with simple models defined by a single parameter and static or deterministically time-variant receive vector autocorrelation matrix. These models do not apply to the interference canceling problem characterizing a receive antenna array in communications, and such closed-form tracking

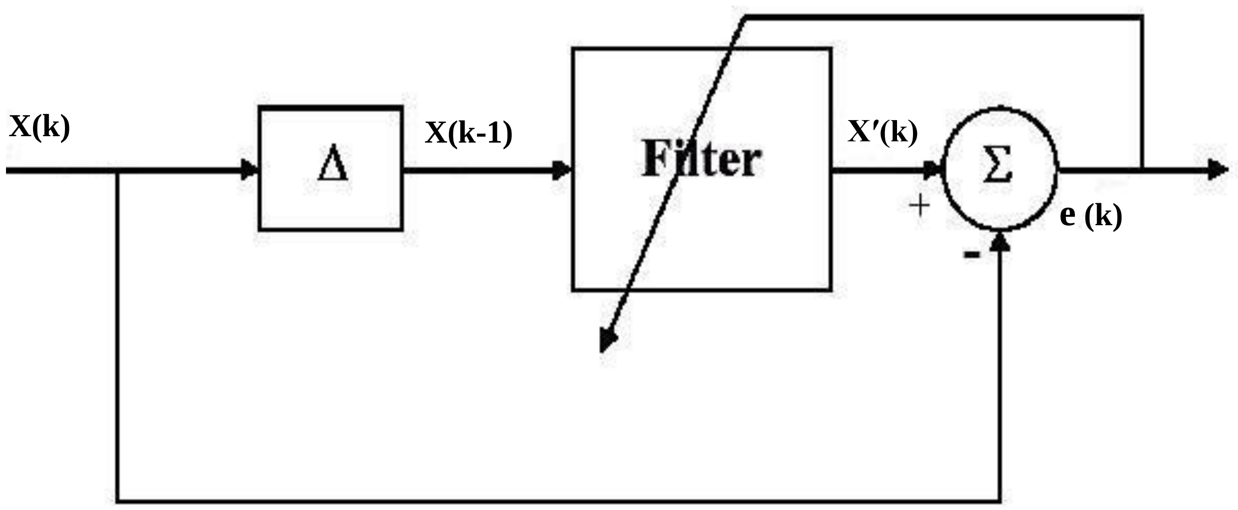

Figure 3. Updating weights for adaptive prediction

analyses cannot be found for the interference canceling problem.

4. Experiments and Results

To achieve the trajectory tracking with the estimate of next position, we operated the system to move in velocity mode. With the difference in the current position of the system and the estimate of position of the object for next sampling instant, the required velocity to reach the estimated position in coherence with the object can be calculated and provided to the system to move with. The other mode of operation is the position mode. In this mode the system could be operated to attain the future positions by directly providing the position to the controller. This may result in the system to stop at the desired position in some cases. To track for the next position, the system would then have to start from rest, and thus there would be jerks in motion and greater torque required because of inertia. The approach of using velocity mode provided good results. The system showed a smooth motion and tracked the object at almost every sampling instant except the instants where the object suddenly changed the behavior of its motion as the estimates diverted more from the actual position at these instants because the estimation uses previous behaviour of motion for the guess and a change in motion caused the guess to be wrong.

The estimation techniques discussed above were applied to the system to estimate the future positions. Since the spherical coordinates are highly non linear being $\sin^{-1}$ and $\cos^{-1}$ functions of the Cartesian coordinates and therefore, need polynomials of very high degree to fit, thus are expensive in terms of time. Thus, to apply the conventional numerical techniques, we chose the Cartesian coordinates of the position to be estimated rather than the spherical coordinates. The Cartesian coordinates were then converted to θ and ϕ, two of the spherical coordinates. Since a physical object moves with either a constant velocity or constant acceleration and can only change the motion in a continuous fashion, we considered the extreme case when the movement may be with an acceleration varying with a constant rate of change which will be closely approximated with at most a fourth degree polynomial. In case of adaptive predictors the technique was applied directly to the two spherical coordinates θ and ϕ. Since the angular motion is also usually continuous so the estimation technique was applied to the angular positions as the filters are applicable also for highly nonlinear data as well as data with smaller degree of nonlinearity.

Figures 4 to 10 reflect the results from the implementation of these techniques to the system for trajectory tracking. In all these figures, the points marked as '*****' are the actual positions of the moving object. The points marked as '**+**' are the estimates of the position of object at the next sampling instant. The points marked as 'o' are the positions obtained by the system tracking the object.

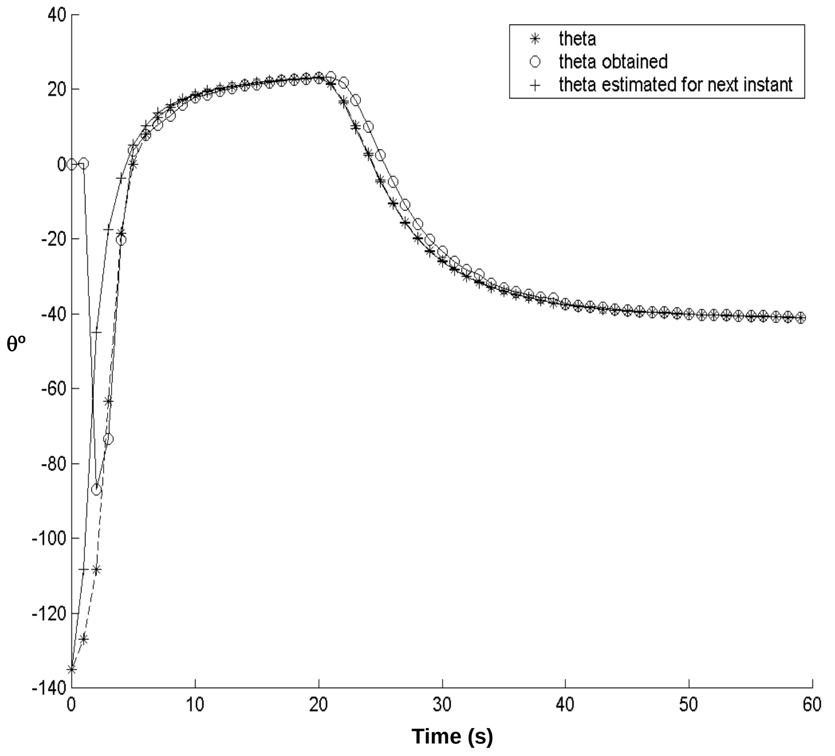

Figure 4. Angle of azimuth obtained through lagrange polynomial of 2nd degree .

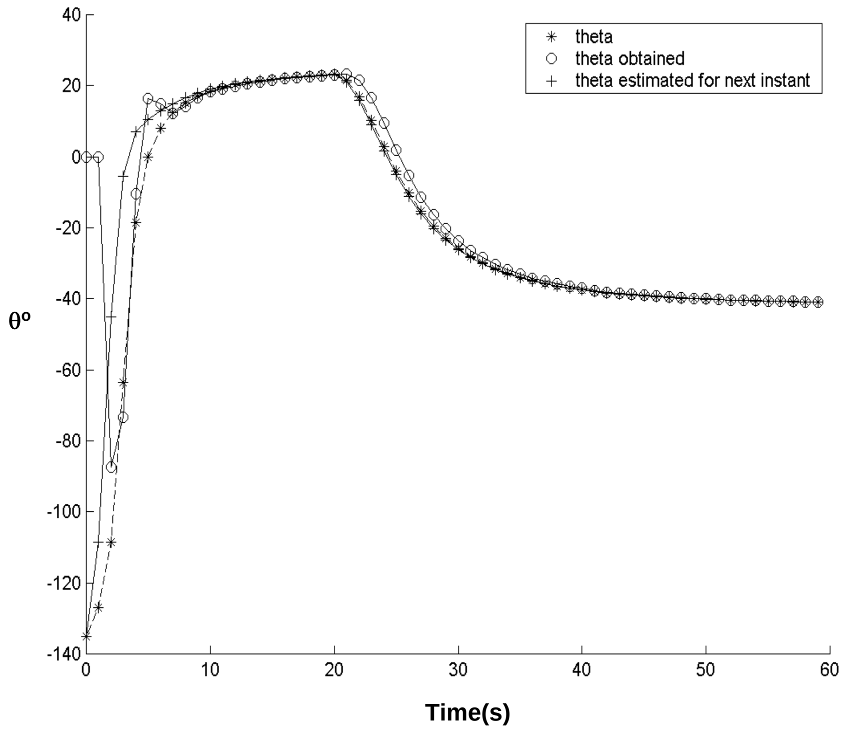

Figure 5. Angle of azimuth obtained through lagrange polynomial of 4th degree.

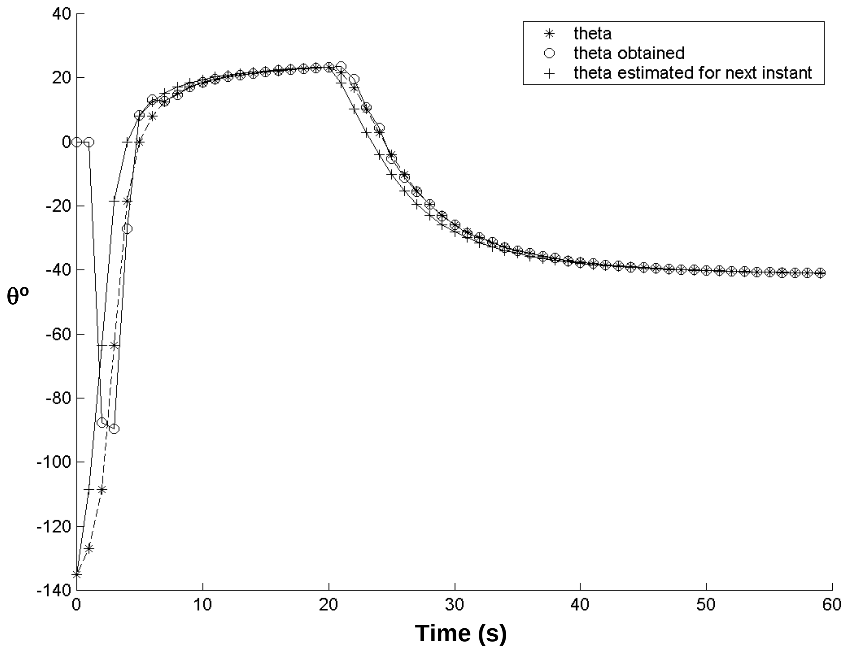

Figure 6. Angle of azimuth obtained through lest square estimation with 2nd degree polynomial.

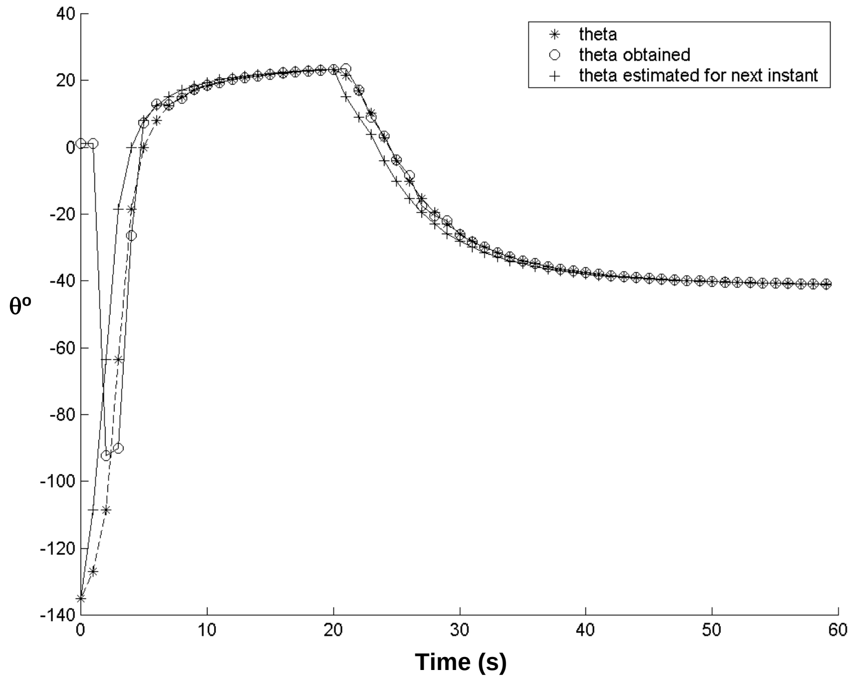

Figure 7. Angle of azimuth obtained through least square estimation with 4th degree polynomial.

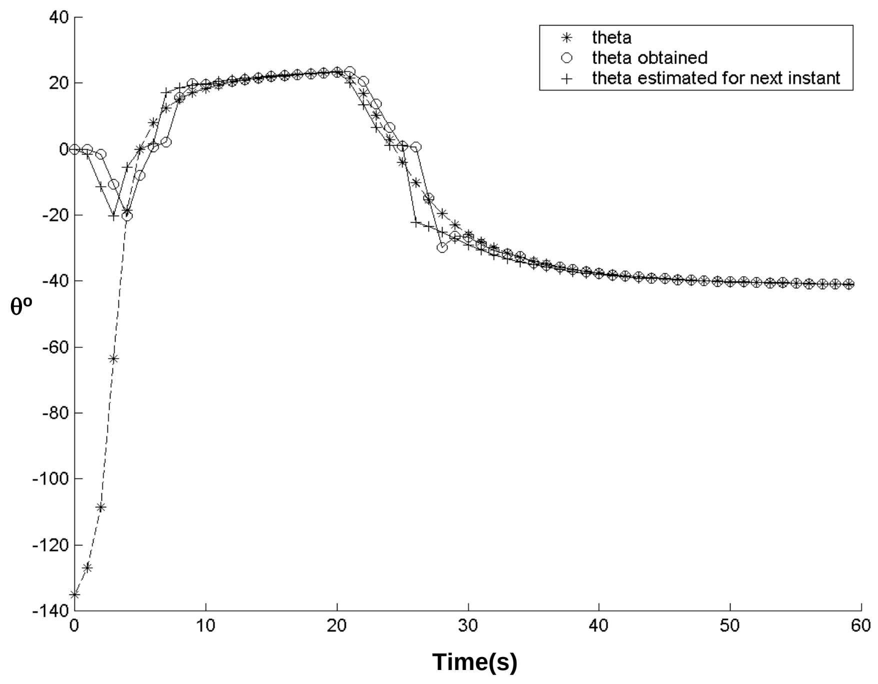

Figure 8. Angle of azimuth obtained through RLS predictor of order 1.

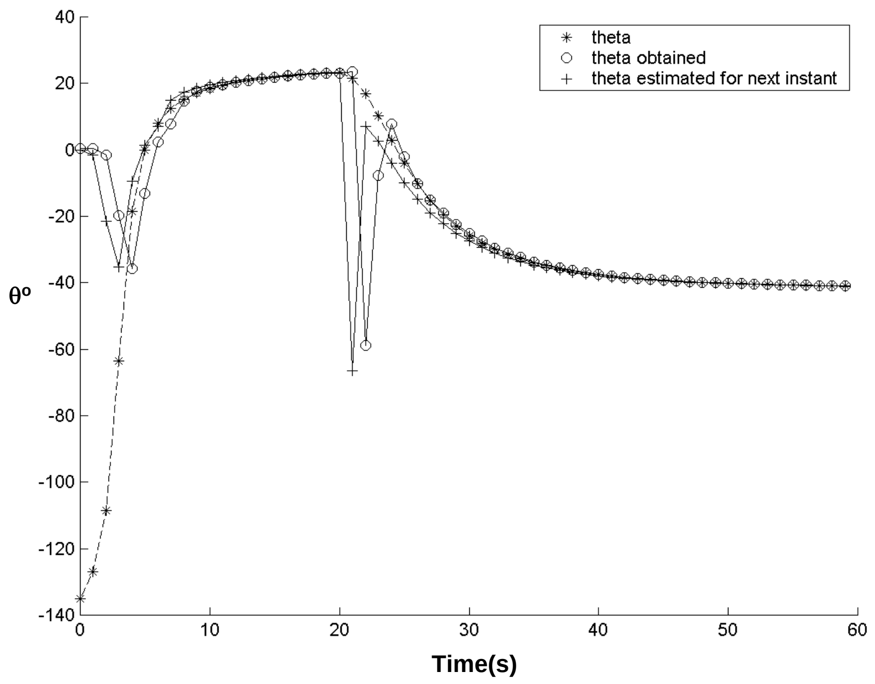

Figure 9. Angle of azimuth obtained through RLS predictor of order 2.

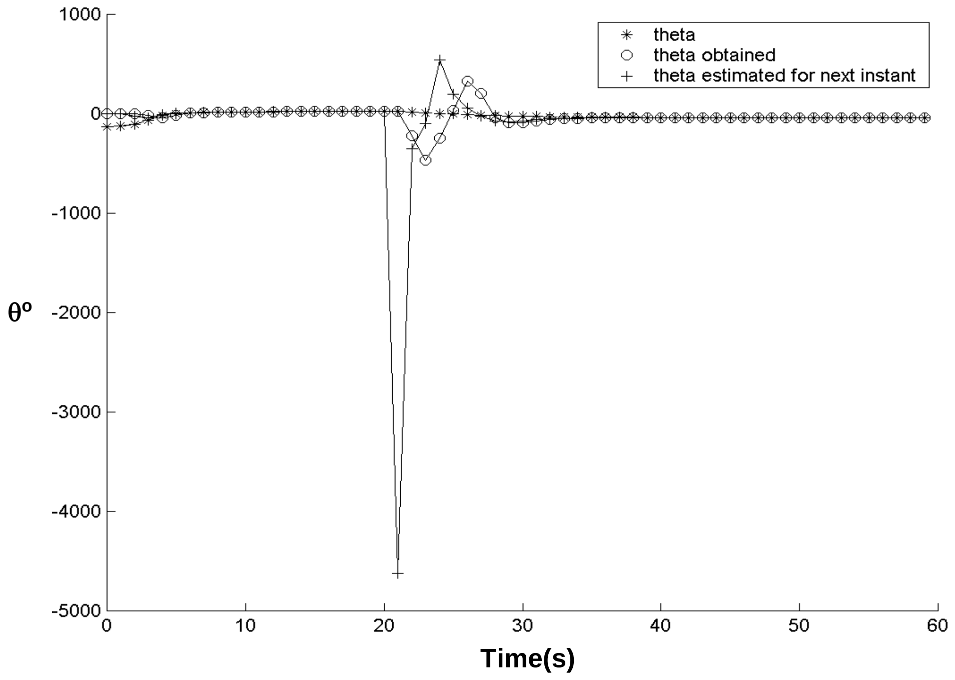

Figure 10. Angle of azimuth obtained through RLS predictor of order 3.

5. Discussion

Looking at the results we see that the Lagrange Polynomials work well as long as the object moves with a constant fashion but start missing at the instants where the object changes its behavior of motion and takes time to come back to the original trajectory. Also since there is no check or measure to minimize the error, the estimates are not very much closed to the original trajectory.

The polynomial fitted through Least Square Estimation tries to best fit the curve to the motion and gets closer estimates by minimizing the least square error. As a result the Least Square Estimation provides good estimates as compared to the Lagrange Polynomials. The estimates are much closed to the actual trajectory as long as the object does not change its behavior of motion. At the time the behavior of motion changes there is slight missing of the estimates from the actual trajectory. In case of the 4th degree polynomial, the error is larger and the estimates miss the original trajectory for a time longer than that in case of the 2nd order polynomial. This is because the 4th order polynomial uses more points from the previous behavior and uses them for a longer time as compared to the 2nd order polynomial.

The RLS predictors also provide good results after they got tuned up. Looking at the results we see that the 2nd order predictor provides good prediction as compared to the 1st and 3rd order predictors. We see that the estimates are closed to the original trajectory as long as the motion has an unchanged behavior. At the instants where the fashion changes the predictors start missing the actual trajectory. Looking at the results we see that at the instants where a change occurs in motion, the higher order predictors estimate more away from the actual trajectory as compared to the lower order predictor but come back sooner to the original trajectory. Of the three predictors used, the second order filter has been found to be fastest in coming back to the original trajectory whereas the error value for it is between those for the 1st and 3rd order filters. This is because of the fact that the second order filter used less number of previous points than the 3rd order filter so the error is smaller than the 3rd order predictor but is larger than the first order predictor as the 1st order predictor uses lesser number of previous points. The faster recovery of the 2nd order predictor to the original trajectory is due to the fact that it uses only two points to predict the next instant position. When the behavior of motion changes, it gets misguided for an instant but at the very next instant it takes both the points it uses from the new fashion. Thus comes back to the original trajectory. The 3rd order predictor on the other hand uses 3 points so it misses the original trajectory for a longer time with a larger error. Here we put some limit to the velocity so as to avoid the complete misguide of the tracking system. As a result the system does not go to track the points very far away from the current trajectory it was tracking as shown near the time instant 20-23 in figure 10.

6. Conclusions

From the above discussion it is reflected that the estimation through the conventional numerical techniques as well as those through the adaptive estimators have been found working well and have merits and demerits of their own. Using the conventional numerical techniques is helpful when the object moves with small variation in the behaviour of its motion. The degree of polynomial is a choice between high and low degrees depending upon the nature of the motion of the object. An object moving with high degree of nonlinearity in its motion can be tracked by using a higher degree polynomial approximation where as the object moving with lower degree of nonlinearity in the motion should be tracked with the estimates from a smaller degree polynomial estimation so as to avoid the misestimates due to the use of more previous values at the instants of sudden changes in the behaviour of motion of the object and to save the time. Between the two methods of polynomial interpolation techniques we found the Least Square Estimation technique much better due to best fitting the curve. Also if there are chances of error in data acquisition, the least square estimation will be a good choice over the Lagrange polynomial approximation. The adaptive predictors on the other hand are good too, if larger error can be tolerated for the sake of quicker return to the original trajectory if misestimates occur due to change in motion. The choice of the order of the predictor filter will again depend upon the tolerance for the error and delay time in coming back to the original trajectory. In case of lower order filter the diversion from the actual trajectory is smaller than higher order predictors but the delay in coming back is larger. Thus smaller order filters should be applied where delay is tolerable over diversion and vice versa. Since the adaptive predictors directly estimate the two angles θ and ϕ the technique is economic in time as less computation is required because only two quantities are estimated and also only one conversion from estimated Cartesian coordinates to spherical coordinates is required for current position and not for the predicted one whereas with numerical techniques we needed to predict three Cartesian coordinates and two conversions from Cartesian to spherical coordinates (i.e. for both the actual and the predicted positions). However if the time cost is less important then the error, the Polynomial fitted through the Least Square Estimation is the best approach over the adaptive predictors due to the tune up delays for the adaptive predictors at the instant of changes.

References

[1] R.L. Burden, J.D. Faires and A.C. Reynolds, “Numerical Analysis”, 1st Edition. Wadsworth, Inc (1979).

[2] C. F. Gerald and P. O. Wheatley, “Applied Numerical Analysis”, 5th Edition Addison-Wesley Publishing Company (1994).

[3] M. Moonen, “Introduction to Adaptive Signal Processing” Department of Electrical Engineering · ESAT/SISTA K.U. Leuven, Belgium url: www.esat.kueleuven.ac.bc/~moonen/asp_course.html

[4] O. Macchi and N. J. Bershad, “Adaptive recovery of a chirped sinusoid in noise, 1. Performance of the RLS algorithm,” IEEE Trans. Acoust., Speech, Signal Processing, Vol. 39 (Mar. 1991), pp. 583–594,.

[5] P. C. Wei, J. R. Zeidler, and W. H. Ku, “Adaptive recovery of a chirped signal using the RLS algorithm,” IEEE Trans. Signal Processing, vol. 45 (Feb. 1997) p. 969–976,.

[6] National Power ICs. “LM628/LM629 Precision Motion Controller. National Semiconductors (1995).