A Linear Smoothing vs. Coordinating turn Model for Projectile Tracking[1]

Ljudmil V. Bojilov, Central Laboratory for Parallel Processing, Bulgarien Academy of Sciences, acad. ”G. Bonchev” str., bl. 25A, Sofia, Bulgaria, e-mail: bojilov@bas.bg

Abstract. In this paper an attempt is made to construct an algorithm for tracking fired projectile in order to determine the exact coordinates of the cannon launching this projectile. Two approaches have been used: a) a linear smoothing algorithm with fixing point and b) a coordinated turn model of target moving. The assumed sensor in this study is agile beam (or electronically scanned array) radar. For use in simulation, an original model of projectile trajectory is derived implementing point mass model of shell flight. In ballistics, the point mass model is one of the simplest model of projectile flight, but it fits very well the real trajectories in the firs few seconds of the flight, so we accept this model to be appropriate enough.. The main goal of the paper is via intensive numerical experiments to investigate the limits of the used approaches in determining the coordinates of firing cannon.

Keywords: target tracking, trajectory model, external ballistics.

1.Introduction. The importance of the problem under consideration is obvious. Destructive power of the artillery is well known and the only way to protect ourselves from this power is to destroy the enemy artillery as soon as possible. The distance between our radar and enemy artillery have to be of order of several tenths of kilometers - close enough to successfully tracks such little targets as projectiles, and not so close to fall into the range enemy shells. There are several limitations in determining the exact coordinates of the hostile batteries: a) the projectile has to be detected immediately after launching, b) the tracking process has to take no more than couple of seconds, and c) the error of estimated coordinates has to be much less than the error of the cannon fire. The first two limitations impose the need of using a sensor with very high speed of scanning, such as agile beam radars. These two limitations are closely connected with the third one. If the projectile is detected in the first second of flight and the tracking process takes the next one or two seconds, we have a chance to receive acceptable results, because the projectile trajectory at this first few seconds is very close to straight line. If, however, the projectile is detected 5 or 6 seconds after launching, we have to track a target moving on complicated nonlinear trajectory and afterwards to extrapolate this trajectory to the ground. Evidently, the obtained results with high probability will be unacceptable.

In practical design stages, particular attention has to be paid to the problem of time allocation between search of new targets and track update of existing targets [Blackman'86], but this issue is out of the scope of this paper.

This paper is organized as follows. In the next section, the problem formulation is presented as well as brief description of the methods used in this paper. In section 3 the tracking algorithms compared here are described. In the first paragraph of section 4 a projectile trajectory model for simulation is derived. In the second paragraph of this section numerical results of our experiments are presented and discussed. The paper ends with brief conclusions.-- peculiarity speciality

2.Problem formulation. The specific task to track flying projectile posed the question of the tracking method to be used. Peculiarity of the task is that the parameters of the projectile trajectory have to be estimated in a very short time and to extrapolate this trajectory to its starting point on the ground. The most promising idea is to try to obtain good enough estimation of the coordinates and velocity at the first detection point and to follow the direction opposite to the estimated velocity vector up to the ground, possibly with some correction.

The estimation problems can be divided into three subgroups [2,3]:

a) prediction, when estimating $\hat{x}(k|j)$, the relation $k > j$ holds;

[1] This work is partially supported by the Bulgarian Ministry of Education and Science under grants I-1205/2002 and I-1202/2002 and by Center of Excellence BIS21 grant ICA1-2000-70016.

b) filtration, when estimating $\hat{x}(k|j)$, the relation $k = j$ holds; and

c) smoothing or retrodiction, when estimating $\hat{x}(k|j)$, the relation $k < j$ holds.

On the other hand, the smoothing estimation problem itself can be divided into another three subgroups:

a) fixed point smoothing, when k is fixed and $j = k+1, k+2, \ldots$;

b) fixed lag smoothing, when k varies and $j = k + L$, where L is the lag;

c) fixed interval smoothing, when $j = N$ (the data interval is up to N) and $k = 0, 1, \ldots, N$.

Evidently, the fixed point smoothing approach is most suitable for our problem. In this case, the time index k will coincide with one of the nearest points after the first detection of the target. After receiving every frame of data the algorithm will give the next more and more precise estimation of state vector at time k. It is clear that the processing of this algorithm can be implemented *on-line*. The same results can be obtained if we use the standard Kalman filter backward, i.e. in the inverse time after receiving all the data. But this approach can be implemented *off-line* only. For comparison, we include the inverse time Kalman Filter in our numerical experiments.

Another approach for solving the stated problem is the coordinated turn model with predetermined turn angle rate ω [3]. The planar equations of an object moving with velocity with constant magnitude and with constant angular rate are

$$\ddot{x} = -\omega\dot{y} \quad \text{and} \quad \ddot{y} = \omega\dot{x}$$

with a state vector

$$X \equiv [x, \dot{x}, y, \dot{y}]. \tag{1}$$

The corresponding transition matrix of the process will be of the form

$$\begin{vmatrix} 1 & \dfrac{\sin\omega T}{T} & 0 & -\dfrac{1-\cos\omega T}{T} \\ 0 & \cos\omega T & 0 & -\sin\omega T \\ 0 & \dfrac{1-\cos\omega T}{T} & 1 & \dfrac{\sin\omega T}{T} \\ 0 & \sin\omega T & 0 & \cos\omega T \end{vmatrix} \tag{2}$$

The tuning parameter in this case is the turn rate angle ω. Using the derived in this paper ballistics model of the projectile we find the value of the angle rate for the first few seconds to be

$$\omega = 0.5^{\circ} / sec \quad \text{or} \quad \omega = 0.0087 rad / sec .$$

This approach, however, can be implemented in the inverse time frame, i.e. *off-line*. Starting from the last received data we proceed to the beginning of the data set, obtaining the state vector estimation in the first detection point. The last step is to extrapolate velocity vector up to the ground.

3.Compared algorithms. Next follows a brief description of particular algorithms compared in this work. These algorithms are: a) fixed point smoothing, b) inverse time Kalman filter, c) coordinated turn model inverse time KF, and d) fixed point smoothing with additional turn at the end of the processing. Two points initiation is accepted and converted measurements Kalman filter is implemented in any of the algorithms.

3.1. Fixed point smoothing algorithm. After track initiation (first two points) a standard Kalman filter is started. We choose the third point of the trajectory as the fixed of our smoothing algorithm, i.e. $k = 3$. At time j the KF cycle ends with the estimation

$$\hat{x}(j|j) = \hat{x}(j|j-1) + W(j)\nu(j), \tag{3}$$

where W is KF gain at time j and ν corresponding innovation.

Next, the smoothing step follows [2]

$$\hat{x}(k|j) = \hat{x}(k|j-1) + B(j)[\hat{x}(j|j) - \hat{x}(j|j-1)], \tag{4}$$

where

$$B(j) = \prod_{i=k}^{j-1} A(i), \quad \text{and} \quad A(i) = P(i|i)\Phi'(i+1,i)P^{-1}(i+1|i).$$

In the last equation, $P(i|i)$ and $P(i+1|i)$ are the Kalman filter state estimation and state prediction covariance matrices respectively and $\Phi(i+1|i)$ is a system transition matrix. After each iteration we receive more and more precise estimation at point $k=3$ of coordinates and velocity of the projectile. The algorithm ends with simple extrapolation of the trajectory to the ground starting from estimated coordinates and following direction, opposite to the estimated velocity.

3.2 Inverse time Kalman filter. This is simply converted measurements KF which is started from the last received data, i.e. in the inverse time manner.

3.3 Coordinated turn KF. In this case the flight model of the projectile is characterized by: a) nearly constant velocity and b) coordinated turn mode with nearly constant turn rate [3]. Augmenting the state vector (1) by one more component - the turn rate ω

$$X \equiv [x,\dot{x},y,\dot{y},\omega] \tag{5}$$

the projectile trajectory model is given by

$$x(k)=\begin{vmatrix} 1 & \frac{\sin\omega T}{T} & 0 & -\frac{1-\cos\omega T}{T} & 0 \\ 0 & \cos\omega T & 0 & -\sin\omega T & 0 \\ 0 & \frac{1-\cos\omega T}{T} & 1 & \frac{\sin\omega T}{T} & 0 \\ 0 & \sin\omega T & 0 & \cos\omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}.x(k-1)+\begin{vmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \\ 0 & 0 \end{vmatrix}.v(k-1)+\begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{vmatrix}.u(k) \tag{6}$$

where u is control input for the turn rate, v - process noise vector and T is sampling time.

As in the previous algorithm, this algorithm can be implemented *off-line*, i.e. after receiving the last data from the target.

3.4 Fixed point smoothing with additional turn. This algorithm is described above fixed point smoothing algorithm with an additional turn of estimated velocity (more precisely, the vector opposite to the estimated velocity) at some appropriated angle.

4. Simulation experiments. Next follows the short description of ballistics trajectory model and numerical results from numerous experiments with presented above algorithms.

4.1 Ballistics trajectory simulation. Since we consider the first few seconds of the projectile flight we accept the most simple ballistics model - the point mass model [4]. In this model only the air resistance is taking into account what is a very good approximation for high speed projectiles - more than twice the speed of sound (with Mach number more than 2). In this study we consider planar case only, i.e. the motion of the projectile in a vertical plane with coordinates (x,z). With this assumptions the drag force is given by

$$F_D=\frac{1}{2}C_D.S.\rho.V^2 \tag{7}$$

where C_D is drag coefficient, $S=\frac{\pi.d}{4}$ is the projectile cross-section (here d is the projectile caliber) , $\rho=1.25\,{}^{kg}\!/_{m^3}$ is the air density and V is the projectile velocity. The physical dimension of the drag force from expression (7) is *Newtons(N)*. This expression can be rewritten in the form

$$F_D=m\dot{V}=-\frac{1}{8}C_D.\rho.\pi.d^2.V^2$$

and for the projectile velocity can be derived the next differential equation

$$\dot{V}=-kV^2\,, \tag{8}$$

where the constant k is given by

$$k=\frac{1}{8m}C_D.\rho.\pi.d^2\,. \tag{9}$$

And now, if do not taking into account the earth gravity, after integrating the equation (8) we obtain the expression for a projectile velocity

$$V(t)=\frac{V_0}{k.V_0.t+1}\,, \tag{10}$$

where V_0 is so called muzzle velocity, i.e. the velocity at which the projectile leaves the cannon barrel. For the two components of the velocity (10) - horizontal and vertical, including gravity, we have

$$V_x(t)=\frac{V_0\cos\theta}{k.V_0.t+1} \quad \text{and} \quad V_z(t)=\frac{V_0\sin\theta}{k.V_0.t+1}-g.t\,,$$

where θ is angle between the velocity vector and the (x,y) plane and g is acceleration of gravity.

As matter of fact, the angle θ depends on time - $\theta=artg\dfrac{V_z(t)}{V_x(t)}$. But in the modeling algorithm we assume that for the very short interval of time, say Δt, this angle remains constant and at the end of this interval it is changed abruptly, i.e. the angle is piecewise constant. The simulation algorithm itself contains the next steps:

- The design parameters: Δt - time step interval; V_0 - muzzle velocity; θ_0 - angle of departure;
- The first step

$$x(1)=x_0+\Delta t.V_0.\cos\theta_0\,, \qquad z(1)=z_0+\Delta t.V_0.\sin\theta_0\,;$$

- The steps of the j^{-th} iteration of the algorithm
  - $t(j)=t(j-1)+\Delta t$;
  - $V_x(j)=\dfrac{V_0.\cos\theta(j-1)}{k.V_0.t(j)+1}$, $\qquad V_z(j)=\dfrac{V_0.\sin\theta(j-1)}{k.V_0 t(j)+1}-g.t(j)$;
  - $\theta(j)=artg\dfrac{V_z(j)}{V_x(j)}$;
  - $x(j)=x(j-1)+\Delta t.V_x(j)$, $\qquad z(j)=z(j-1)+\Delta t.V_z(j)$.

The choice of time interval Δt in our trajectory simulation algorithm is connected with the sampling time of the Kalman filter which ranges from $0.001s$ to $0.003s$, and as the most natural, we accept the two intervals to be equal. As for the sampling interval, we make use of expression presented in [1, pp. 335-336] relating the sampling interval to the prediction error

$$T=0.4P_D\left(\frac{\sigma_0\sqrt{\tau_m}}{\sigma_m}\right)^{0.4}\frac{v_0^{2.4}}{1+0.5v_0^2}\,,$$

where T is sampling interval; σ_0 is observation standard deviation; τ_m is maneuver time constant; σ_m is maneuver standard deviation; $\sigma_p=\sqrt{p_{11}(k+1|k)}$ (here $p_{11}(k+1|k)$ is the first diagonal element of $P(k+1|k)$), and $v_0=\dfrac{\sigma_p}{\sigma_0}$ is standard deviation reduction ratio. To achieve good results in estimating coordinates of the firing cannon we have to ensure considerable reduction of the prediction error (σ_p in our case) compared to the observation error - σ_0, say 10 times, i.e. for the standard deviation reduction error we have $v_0=0.1$. In our case, maneuver of the target is due to acceleration of gravity g. Taking the projection of g on direction normal to the trajectory we have $\sigma_m=6.93$. For maneuver time constant we accept the value of one. In all the experiments we use two values of distance error - $10m$ and $20m$. For these two values of observation error, and accepting $P_D\cong 1$, we obtain for sampling time interval $T=1.83E-3s$ and $T=2.42E-3s$, respectively.

4.2 Numerical results. We have performed our tests with three values of sampling time: $0.001s$, $0.002s$ and $0.003s$, and constant duration of tracking - 1second. The overall frame of experiments includes three different calibers: 155mm, 120mm and 100mm shells. For any single caliber - three different values of sampling time are implemented, filling up the tables below. All these tests are performed for angle of departure $\theta=45^0$. In each one table the results of performance of four described above algorithms are included by using next notations: Smoothing algorithm - **Sm**; Inverted time KF - **IKF**; Coordinated turn algorithm - **CT** and Smoothing algorithm plus one step coordinated turn - **SmCT**. Each one value in the tables below is averaged over 1000 Monte Carlo runs.

The next three tables contain data concerning 45^0 angle of departure.

Table 1. Mean error and standard deviation (in meters) for 155 mm caliber shell.

Compared algorithms	T = 0.001		T = 0.002		T = 0.003	
	mean error	standard deviation	mean error	standard deviation	mean error	standard deviation
Sm	8.895	4.93	8.8	7.1	9.15	9.14
IKF	9.03	4.87	8.81	7.08	9.29	9.24
CT	3.777	4.99	3.5	7.13	4.03	9.24
SmCT	0.21	4.92	0.114	7.05	0.41	9.09

Table 2. Mean error and standard deviation (in meters) for 120 mm caliber shell.

Compared algorithms	T = 0.001		T = 0.002		T = 0.003	
	mean error	standard deviation	mean error	standard deviation	mean error	standard deviation
Sm	8.83	4.92	8.8	7.25	9.3	9.27
IKF	8.96	4.92	9.1	7.18	9.52	9.36
CT	3.68	5.11	3.85	7.38	4.29	9.36
SmCT	0.13	4.91	0.1	7.23	0.54	9.19

Table 3. Mean error and standard deviation (in meters) for 100 mm caliber shell.

Compared algorithms	T = 0.001		T = 0.002		T = 0.003	
	mean error	standard deviation	mean error	standard deviation	mean error	standard deviation
Sm	8.88	5.04	9.2	7.33	8.97	9.24
IKF	9	5.01	9.36	7.25	9.17	8.9
CT	3.76	5.11	4.1	7.25	3.93	8.9
SmCT	0.18	5.04	0.5	7.3	0.23	9.21

The next two tables contain data concerning 40^0 and 35^0 angle of departure respectively and for 155 mm caliber shell.

Table 4. Mean error and standard deviation (in meters) for 40^0 angle of departure..

Compared algorithms	T = 0.001		T = 0.002		T = 0.003	
	mean error	standard deviation	mean error	standard deviation	mean error	standard deviation
Sm	12.45	6.36	11.95	8.78	12.2	11.26
IKF	12.53	6.24	12	9	12.48	11.29
CT	6.23	6.36	5.63	9.04	6.08	11.26
SmCT	1.84	6.34	1.32	8.72	1.49	11.2

Table 5. Mean error and standard deviation (in meters) for 35^0 angle of departure.

Compared algorithms	T = 0.001		T = 0.002		T = 0.003	
	mean error	standard deviation	mean error	standard deviation	mean error	standard deviation
Sm	17.93	7.92	17.85	11.46	17.43	14.34
IKF	17.95	7.82	17.8	11.14	17.96	14.51
CT	10.11	8.04	9.9	11.2	10.03	14.4
SmCT	4.13	7.74	4.03	11.2	3.58	14.02

The results from numerical experiments show that for all calibers between 155-100 *mm* and angle of departure 45^0 the coordinate estimation of enemy cannon position is very good. The worst results were obtained for sampling time interval of T = 0.003 *s* (and hence, with number of scans 334), even though this estimation precision with mean error less than one meter and standard deviation less than 10 meters is highly acceptable. The smoothing algorithm with the single step coordinated turn at the end of the processing is leading in terms of mean error, while the standard deviation for all compared algorithms is almost the same. This result can be expected taking in mind that each one of these algorithms contains standard Kalman filter cycle .

It can be noticed surprisingly weak dependence of the estimation parameters on projectile caliber. We have made additional experiment with 75 *mm* projectile, 45^0 angle of departure and T = 0.001*s*. The results are very close to those of the first three tables. We have included in our experiments the mentioned above range of calibers only because the cannons with less than 100 mm caliber are rarely used for long range gunfire.

The last two tables contain results from experiments with 40^0 and 35^0 angles of departure. It can be seen that the estimation parameters are inversely proportional to angle of departure and for angle of 35^0 the results are the worst. However, even this estimation accuracy is acceptable in some extend taking into consideration that for 155 *mm* cannon and at firing distance of 10 kilometers the fire accuracy is determined by ellipse with 45 *m* lateral and 130 *m* longitudinal semi axes. For these two cases smoothing algorithm with additional single coordinated turn once again outperform the other algorithms in terms of mean error and, as before, the four algorithms give almost equals standard deviations. For testing very low angle of departure we have made additional experiment with angle of 20^0 for 155 *mm* shell and T = 0.001*s*. All four algorithms give very close values of standard deviation - about 24-25 *m*, but as to mean error, the leading algorithm gives value of 8 *m*, while the others - 60-83 *m*. Even in this case the estimation accuracy, especially of the leading algorithm, is better than the fire accuracy of the cannon.

And the last array of experiments concerns the increasing value of distance error of the sensor. We repeat the experiments with all initial conditions as in Table 1 with only exception - $\sigma_r = 20m$ instead $\sigma_r = 10m$. In the obtained results the mean errors are practically the same as in Table 1, and as to standard deviations, they are slightly increased - with no more than 15%.

5. Conclusions. In this paper a numerical investigation is presented of four tracking algorithms described in section 3. Compared algorithms are used for tracking fired projectile at a distance of about 30000 *m*. For ballistics trajectory simulation (more precisely, the first few seconds of the flied) an original algorithm is derived in subsection 4.1. The final aim of the tracking is to estimate the coordinates of firing cannon. In our experimental frame the motion of the projectile is considered in a vertical plane with coordinates (x,z) and so, the firing cannon have a single coordinate - x. According to the obtained experimental results the most promising algorithm is the linear smoothing algorithm with additional coordinated turn step at the end of the processing. This algorithm outperformed the others in terms of mean error, while the standard deviations obtained in the experiments were almost equal for all tested algorithms. As it can be expected, the estimation accuracy is inversely proportional to the angle of departure, but the experiments with very low angle of departure show that even for this case the estimation precision is better than the firing accuracy of the cannon.

REFERENCES:

1. Blackman, S., Multi-Target Tracking with Radar Applications, Artech House, Norwood, MA, 1986.

2. Meditch, J.S, Stochastic optimal linear estimation and control, McGraw Hill, New York,1969.

3. Bar-Shalom, Y., X.R. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, Norwood, MA, 1993.

4. Bray, Derek, External Ballistics course, http://www.rmcs.cranfield.ac.uk/aeroxtra/, Cranfield University Aerospace Dept., DAPS, Royal Military College of Science Shrivenham, Swindon Wilts., SN6 8LA