Chebyshev Polynomials for Calculating the Mean Line of Advance of a Manoeuvring Air Target

Robert L. Carling
Defence R&D Canada – Valcartier,
2459 Pie XI Boulevard North,
Val-Belair, Quebec,
G3J 1X5,
CANADA.
Tel: (418) 844-4000-4432
E-mail: robert.carling@drdc-rddc.gc.ca

Abstract

An aeroplane that bombs a warship or attacks it with anti-ship missiles usually chooses a manoeuvring trajectory in order to confuse the tracking radars on board the ship. The classical threat evaluation model assumes that the air target moves in a straight line and ranks this air threat according to values calculated for the target's closest point of approach and its time to reach the closest point of approach. Unfortunately, this kind of threat evaluation model leads to rapid inversion of the threat levels within a short time period thus preventing the Above Water Warfare Officer from making a consistent assessment about the most lethal threat attacking his ship. This article describes an approach to solving the threat level instability problem by using the mean line of advance of the manoeuvring threat. Details of the algorithm for calculating the mean line of advance are given here as well as some examples of aeroplane trajectories for which the mean line of advance was calculated using a Sun computer.

1. Introduction

The classical threat evaluation system of a naval command and control system assumes that the air targets fly in a straight line. The closest point of approach (CPA) and the time to reach the CPA (TCPA) are then calculated from the instantaneous velocity vectors of the air threats. Since, for manoeuvring targets the CPA and TCPA are continually oscillating between large and smaller values, two manoeuvring targets that attack a ship with almost the same range will produce ranked threat levels that never clearly indicate over a reasonable time interval which threat of the two is the most highly ranked. In other words, if the manoeuvres of the threats give threat 1 a high threat level and give threat 2 a low threat level at a given instant of time, then subsequent manoeuvres performed immediately afterwards will give threat 1 a low threat level and threat 2 a high threat level, thus inverting the threat rankings. Therefore, the manoeuvres of the two air threats cause instability in the threat ranking. This kind of threat level stabilisation problem is described in detail in the references [Paradis *et al*., 1997] and [Carling, 1999].

It is believed that this problem can be solved by constructing a threat evaluation model based on values of CPA and TCPA coming from the mean line of advance of the air threat. The mean line of advance is the straight line giving the most probable direction of advance of the air threat. Strictly speaking in mathematical terms, if x(t), y(t) are continuously differentiable functions on

$[t_1,t_2]$ representing the x and y components of displacement of an air threat as a function of time then the mean line of advance of the air threat is defined to be the line:

$$lx(t) = at + b; ly(t) = ct + d$$

such that the integrals

$$\int_{t_1}^{t_2} (x(t) - (at + b))^2 \, dt; \int_{t_1}^{t_2} (y(t) - (ct + d))^2 \, dt$$

are minimised. In general, the linear functions lx(t), ly(t) obtained by minimising the above integrals do not tightly adhere to the original curves x(t), y(t). However, the above integral does define a norm on the normed space of square integrable functions $L_2[t_1,t_2]$. Instead of using an integral norm, the author used a uniform norm on the normed space of continuously differentiable functions $C^{(1)}[t_1,t_2]$. The uniform norm of a continuously differentiable function x(t) is defined by

$$\| x \|_\infty = \max_{t_1 \le t \le t_2} | x(t) |.$$

Electrical engineers have successfully approximated the frequency response of various digital filters in terms of Chebyshev polynomials by using the uniform norm (see [Parks and Burrus, 1987]). The form of the frequency response of the filters considered in [Parks and Burrus, 1987] appeared to be quite general in nature. The author thought, therefore, that it would be possible to approximate arbitrary aeroplane trajectories by straight lines using the uniform norm to measure the accuracy of the approximation.

The x(t) and y(t) coordinates of a manoeuvring aeroplane are continuous functions on some time interval $[t_1,t_2]$ and their x and y velocities $x^{(1)}(t)$, $y^{(1)}(t)$ are also continuous functions on $[t_1,t_2]$. However, the ship's radar tracker does not provide continuous updates of x(t), y(t), $x^{(1)}(t)$, $y^{(1)}(t)$ but only discrete updates, depending on the radar antenna scan rate. The author had therefore to devise an interpolation scheme that generated values of x(t), y(t) on the whole interval $[t_1,t_2]$ given a discrete number of n data points $(x(t_i), y(t_i))$, i = 1,…,n belonging to the interval $[t_1,t_2]$. The author interpolated the discrete values $(x(t_i),y(t_i))$, i = 1,…,n using Chebyshev polynomials and then levelled the Chebyshev error curve using eighth degree polynomials. This produced continuously differentiable functions p(t), q(t) such that $p(t_i) = x(t_i)$, i =1,…,n , $q(t_i) = y(t_i)$, i =1,…,n and for u satisfying $t_i \le u \le t_{i+1}$, p(u) varied in an almost linear fashion from $x(t_i)$ to $x(t_{i+1})$ and q(u) varied in an almost linear fashion from $y(t_i)$ to $y(t_{i+1})$. It is believed that in this preliminary model, an almost linear movement of the aeroplane between each one of the radar detection points is a reasonable approximation.

The mean line of advance was calculated using an iterative algorithm called the exchange algorithm. In order to make this algorithm execute successfully for arbitrary continuously differentiable functions x(t), y(t) on arbitrary time intervals $[t_1, t_2]$, it is necessary to have an accurate estimate of the function values between the given radar data points. In fact, a slight difference in the interpolation scheme causes a variation in the coefficients of the mean line of advance obtained. In addition, if the interpolation does not estimate the function values accurately at the intermediate points, the exchange algorithm may fail to calculate the mean line of advance.

Although a good approximation p(t) has been found for x(t), $p^{(1)}(t)$ does not necessarily approximate $x^{(1)}(t)$. The exchange algorithm starts off with a candidate line lx(t)=at+b. This candidate line is updated by calculating the point at which |p(t)-lx(t)| attains its maximum value. Since this point cannot be found by solving $p^{(1)}(t) = lx^{(1)}(t)$, a binary search is performed in the neighbourhood of the global maximum in order to find its precise value. The updated lines then converge uniformly to the line l(t) for which

$$\| x - l \|_{\infty}$$

is a minimum.

If two threats are flying in aircraft formation at the same speed to attack a ship and one of the threats is a few kilometres further from the ship than the other, then it is believed that the variation of the mean line of advance of the further threat cannot give it a time of closest point of approach (TCPA) that is smaller than the TCPA calculated from the mean line of advance of the nearer threat. If the two manoeuvring threats are therefore separated by a few kilometres and have comparable CPA, it is unlikely that the threat ranking will be inversed because the TCPA of the further threat is smaller than that of the nearer threat. However, if the two threats are closer together then it is necessary to have an accurate estimate of the deviation of the mean line of advance in order to know when the increase of the mean line of advance velocity of the further threat is sufficiently greater than that of the nearer threat to cause an inversion of the threat rankings. These ideas will be used in the design of a stabilised threat-ranking model.

The algorithm described in this article calculates the mean line of advance of a manoeuvring air target over an eight second time interval for a radar that updates the air track every second. Algorithms for calculating the mean line of advance over arbitrary time intervals and for more realistic radar update rates such as two-second and four-second update rates will require a generalisation of the algorithms described in this article.

2. **Interpolation using Chebyshev Polynomials**

Chebyshev polynomials have been successfully used to approximate the frequency response of digital filters over a finite discrete frequency set. This motivated the author to try to approximate the x and y time series of an air radar track over the infinite continuous interval from 0 to m seconds where m can be either 7 or 8 seconds. In the theory of approximation of continuously differentiable functions by Chebyshev polynomials on an interval [-1,1] see [Hamming, 1973], the function f(t) is represented by a linear combination of Chebyshev polynomials as given by the finite sum below.

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{n} a_k T_k(t)$$

Here the a_k are defined by the following integrals

$$a_k = \frac{2}{p} \int_0^p f(\cos q) \cos kq \, dq$$

and the $T_k(t)$ are defined recursively by the following relations.

$$T_0(t) = 1; T_1(t) = t;$$
$$T_{n+1}(t) - 2tT_n(t) + T_{n-1}(t) = 0$$

Since the function f(t) must be approximated on the interval [0,m] rather than the interval [-1,1], the transformation p(t) = (-2t)/m + 1 is used to map the interval [0,m] into the interval [-1,1]. Define therefore a function g on [-1,1] having the values of f multiplied by the constant 2/m. Thus, g(t) = (2/m)f((-m/2)t+m/2). The coefficients c_k for the function g are therefore given by

$$c_k = \frac{2}{\pi}\int_0^{\pi} g(\cos\theta)\cos k\theta \, d\theta .$$

Inside the interval [-1,1], the function g is only known at the points –3/4, -1/2, -1/4, 0, 1/4,1/2, 3/4. Now generate a partition of [0,π] by defining the points

$$0 = \alpha_0 < \alpha_1 < \alpha_2 < \ldots < \alpha_{m-1} < \alpha_m = \pi$$

where α_1 = arc cos(3/4); α_2 = arc cos(1/2); α_{m-1} = arc cos(-3/4); and α_m = arc cos(-1). In order to calculate c_0, the integral

$$\int_0^{\pi} g(\cos\theta)d\theta$$

is replaced by the sum

$$\sum_{j=0}^{m-1}\int_{\alpha_j}^{\alpha_{j+1}} g(\cos\theta)d\theta$$

where on each $[\alpha_j,\alpha_{j+1}]$ the function g(cos θ) has been replaced by the linear approximation

$$g(\cos\alpha_j) + \frac{g(\cos\alpha_{j+1}) - g(\cos\alpha_j)}{\alpha_{j+1}-\alpha_j}(\theta - \alpha_j)$$

or

$$d_j + w_j(\theta - \alpha_j); d_j = g(\cos\alpha_j); w_j = \frac{d_{j+1}-d_j}{\alpha_{j+1}-\alpha_j}.$$

Finally, therefore, a closed loop expression exists for c_0

$$c_0 = \frac{2}{\pi}\sum_{j=0}^{m-1}\left(\frac{d_j + d_{j+1}}{2}\right)(\alpha_{j+1}-\alpha_j)$$

where $d_j = g(\cos \alpha_j)$.

In order to find a closed loop expression for c_1, integration by parts is performed on the integral expression for c_1 giving

$$c_1 = \frac{1}{\pi}\sum_{j=0}^{m-1}\left(\frac{b_{j+1}+b_j}{2}\right)(\alpha_{j+1}-\alpha_j) - \frac{1}{4\pi}\sum_{j=0}^{m-1} s_j(\cos 2\alpha_{j+1} - \cos 2\alpha_j)$$

where

$$b_j = g^{(1)}(\cos\alpha_j); s_j = \frac{b_{j+1}-b_j}{\alpha_{j+1}-\alpha_j}.$$

Integration by parts is applied to the integral expression for $c_n, n \geq 2$ to give
where s_j is defined above. The useful approximations to the function g have been found by programming the above expressions for c_0, c_1 and c_n and comparing linear combinations of the

$$c_n = \frac{1}{np}\left(\frac{1}{(n-1)^2}\sum_{j=0}^{m-1} s_j(\cos(n-1)a_{j+1} - \cos(n-1)a_j) - \frac{1}{(n+1)^2}\sum_{j=0}^{m-1} s_j(\cos(n+1)a_{j+1} - \cos(n+1)a_j)\right)$$

$$P_2(t) = \frac{c_0}{2} + c_1T_1(t) + c_2T_2(t)$$

$$P_3(t) = \frac{c_0}{2} + c_1T_1(t) + c_2T_2(t) + c_3T_3(t)$$

Chebyshev polynomials involving these coefficients with the function values g(t). It was found that the best approximations were obtained with the following low order polynomials.

Now, the c_i are calculated on $[0,\pi]$ and both $T_n(t)$ and g(t) are defined on [-1,1]. At first, therefore, the polynomials $P_n(t)$, n = 2,3 exist on [-1,1] only, but under the transformation, q(t) = (m/2)t, $P_n(t)$ can be extended to [-m/2,m/2] and the error term E(t) is put equal to g(t)-P_n((m/2)t). Here the function g(t) is redefined on [-m/2,m/2] by the formula g(t) = (2/m)f(t+m/2) so that E(t) will be a well defined function on [-m/2,m/2]. Generally speaking, the form of |E(t)| is given by the graph below.

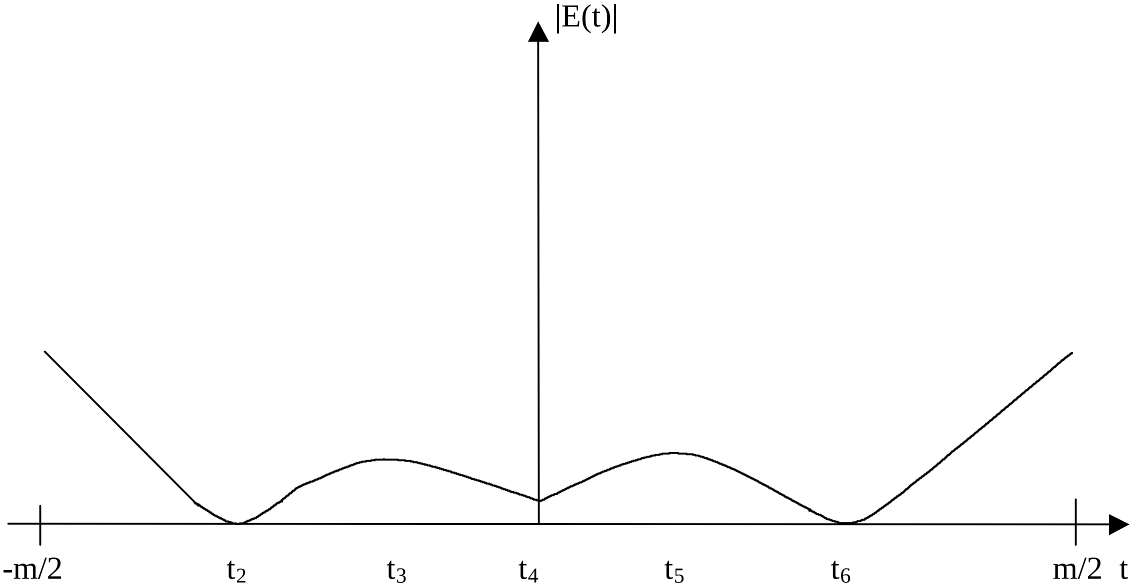

Figure 1 – Graph of the Chebyshev error curve |E(t)|

In Figure 1 above

$$t_1 = -m/2 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 = m/2.$$

t_1, t_7 are points where |E(t)| attains a maximal endpoint value, t_2, t_6 are zeros of |E(t)|, t_3, t_5 are points where |E(t)| has a local maximum, and t_4 is a point where |E(t)| attains a local minimum. The following theorem can be used to level this error curve.

Theorem 1

Let a Chebyshev error curve E(t) which is known at n+1 points α_0, α_1, α_2, …,α_n be specified by the real numbers $E(\alpha_0)$, $E(\alpha_1)$, $E(\alpha_2)$,…,$E(\alpha_n)$. A polynomial approximation $P_n(t)$ can then be constructed by defining

$$P_n(t) = a_0 + a_1 t + ... + a_n t^n$$

$$\frac{1}{(a_1 - a_0)(a_1 - a_2)...(a_1 - a_i)}, i = 0,...,n$$

where each a_i, i = 0,…,n is a function of $E(\alpha_k)$, k = 0,…,i and of products

and certain symmetric functions of $\alpha_0,\alpha_1,\ldots,\alpha_i$.

Proof

In order to calculate the coefficients a_i, the linear system of n+1 equations in n+1 unknowns $P_n(\alpha_i)=E(\alpha_i)$, i = 0,…,n has to be solved. This is accomplished numerically by using Gaussian elimination. The author gives here some of the coefficients for the case when n=6 since the others have a similar representation.

$$a_6 = \sum_{k=1}^{6} \frac{E(a_k) - E(a_0)}{\prod_{\substack{j=0,\\ j\neq k}}^{6} (a_k - a_j)}$$

$$a_5 = -a_6 (\sum_{j=0}^{5} a_j) + \sum_{k=1}^{5} \frac{E(a_k) - E(a_0)}{\prod_{\substack{j=0,\\ j\neq k}}^{5} (a_k - a_j)}$$

$$a_4 = -a_5 (\sum_{j=0}^{4} a_j) - a_6 (\sum_{j,k\in\{0,1,2,3,4\}} a_j a_k) + \sum_{k=1}^{4} \frac{E(a_k) - E(a_0)}{\prod_{\substack{j=0,\\ j\neq k}}^{4} (a_k - a_j)}$$

$$a_3 = -a_4 (\sum_{j=0}^{3} a_j) - a_5 (\sum_{j,k\in\{0,1,2,3\}} a_j a_k) - a_6 (\sum_{i,j,k\in\{0,1,2,3\}} a_i a_j a_k) + \sum_{k=1}^{3} \frac{E(a_k) - E(a_0)}{\prod_{\substack{j=0,\\ j\neq k}}^{3} (a_k - a_j)}$$

Here the notation

$$\sum_{j,k\in\{0,1,2,3\}} a_j a_k$$

means that

$$\sum_{j,k\in\{0,1,2,3\}} a_j a_k = a_0^2 + a_1^2 + a_2^2 + a_3^2 + a_0 a_1 + a_0 a_2 + a_0 a_3 + a_1 a_2 + a_1 a_3 + a_2 a_3.$$

The idea of generating a polynomial function that takes on prescribed values $E(\alpha_i)$ at each of the points α_i in the interval [0,m] is similar to the problem of Lagrange interpolation. Now the error between the original function g(t) and the Lagrange interpolating polynomial can be quite large at points of [0,m] not equal to the α_i. In order to keep this error small everywhere in the interval

$$a = l + m\cos(\frac{[2(n-i)+1]p}{2(n+1)}), i = 0,1,...,n \quad n = 7$$

[0,m], the latter is subdivided into three subintervals and the α_i are taken to be the zeros of an appropriate eighth degree Chebyshev polynomial. These zeros are defined by the formula
where λ is the midpoint of [-m/2,-m/6] or [-m/6,m/6] or [m/6,m/2] and μ=m/6. Hence levelling of the Chebyshev error curve takes place in an interval of length m/3 that contains eight zeros α_i. On average, therefore, the points α_i are separated by approximately m/24 seconds. It was assumed that the function g(t) changed linearly from each of the points α_i to the next point α_{i+1}. Now E(t) is defined to be g(t)-T_2(t) where T_2(t) is a Chebyshev polynomial of order 2. Hence g(t) is locally a quadratic polynomial. The following theorem whose proof may be found on page 35 of [Powell, 1981] can be used to estimate the values of E(t) in the interval [0,m].

Theorem 2

For any set of distinct interpolation points $\{t_i, i = 0,1,\ldots,n\}$ in [a,b] and for any $f \in C^{(n+1)}$[a, b], let p be the nth degree polynomial satisfying $p(t_i)=f(t_i)$,i = 0,…,n. Then, for any t in [a,b], the error e(t) = f(t)-p(t), $a \le t \le b$, has the value

$$e(t) = \frac{1}{(n+1)!}\prod_{i=0}^{n}(t-t_i)\, f^{(n+1)}(x)$$

where ξ is a point of [a,b] that depends on t.

This theorem may be applied to the function E(t) on a subinterval of length m/3. On this subinterval E(t) behaves like a quadratic polynomial in t and hence when it is interpolated by a polynomial of degree eight, the error is uniformly equal to 0 in this subinterval. The above theorem may be used on each of the intervals [-m/2, -m/6], [-m/6, m/6] and [m/6, m/2] to generate polynomials P(t), R(t), Q(t) approximating E(t) on each of these intervals. Thus

$$E(t) = P(t), \qquad -m/2 \le t \le -m/6$$
$$g(t) - T_2(t) = P(t), \qquad -m/2 \le t \le -m/6$$
$$g(t) = T_2(t) + P(t), \qquad -m/2 \le t \le -m/6$$
$$E(t) = R(t), \qquad -m/6 \le t \le m/6$$
$$g(t) - T_2(t) = R(t), \qquad -m/6 \le t \le m/6$$
$$g(t) = T_2(t) + R(t), \qquad -m/6 \le t \le m/6$$
$$E(t) = Q(t), \qquad m/6 \le t \le m/2$$
$$g(t) - T_2(t) = Q(t), \qquad m/6 \le t \le m/2$$
$$g(t) = T_2(t) + Q(t), \qquad m/6 \le t \le m/2.$$

The function g(t) is now known at each point t in the interval [-m/2,m/2] and can be used to calculate the mean line of advance of the function f(t). In particular, f(t) can be the x component of the threat’s displacement x(t) or its y component y(t).

3. **The Mean Line of Advance Algorithm**

3.1 ***General Overview***

The mean line of advance of a function f(t) defined on [-m/2,m/2] is calculated by a one-point iterative exchange algorithm. In general, an attempt is made to approximate a continuously differentiable function f(t) defined on an interval [a, b] by a straight-line l(t) defined on the same interval [a, b]. Here a and b are two real numbers satisfying a<b. The set of all continuously

differentiable functions on [a, b] denoted by $C^{(1)}[a, b]$ is a linear space because the sum of two continuously differentiable functions is continuously differentiable and the product of a real scalar with a continuously differentiable function is also continuously differentiable. In addition it can be made into a normed linear space by introducing the norm

$$\| f \|_\infty = \max_{a \le t \le b} | f(t) |, \qquad f \in C^{(1)}[a,b]$$

The above norm satisfies the properties

$$(1) \| f \|_\infty \ge 0 \ \ and \ \ \| f \|_\infty = 0 \ \ iff \ \ f = 0 \ \ for \ \ f \in C^{(1)}[a,b]$$

$$(2) \| lf \|_\infty = | l | \| f \|_\infty \qquad l \ \ real, f \in C^{(1)}[a,b]$$

$$(3) \| f + g \|_\infty \le \| f \|_\infty + \| g \|_\infty \quad f, g \in C^{(1)}[a,b]$$

and hence makes $C^{(1)}$ [a, b] into a normed linear space. The set of all lines in $C^{(1)}[a, b]$ denoted by

$$L = \{ d_0 + d_1 t \mid t \in R \ \ for\ fixed\ d_0, d_1 \in R \}$$

is a two-dimensional subspace of $C^{(1)}[a, b]$. It is a subspace of $C^{(1)}[a, b]$ because the sum of two lines is a line and the product of real scalar λ with a line gives another line. It is a two-dimensional subspace because {1,t} is a basis of the subspace. It is shown on pages 137-139 of [Davis, 1963] that if f belongs to a normed linear space E, S is a finite dimensional linear subspace of E and f does not belong to S, then there always exists an element s_0 of S that is closest to f. In other words, there always exists an element $s_0 \in$ S such that

$$\| f - s_0 \|_\infty = \min_{s \in S} \| f - s \|_\infty .$$

It is then shown on pages 142-145 of [Davis, 1963] that this element s_0 is unique. In particular, if E = $C^{(1)}[a, b]$, S=L, then for any continuously differentiable function $f \in C^{(1)}[a, b]$, there always exists a unique line $l_0 \in$ L which is the best linear approximation in the uniform norm to f, i.e.,

$$\| f - l_0 \|_\infty = \min_{l \in L} \| f - l \|_\infty .$$

This line minimises the worst-case difference between itself and the given function f. On page 77 of [Powell, 1981], the following theorem, which characterises the best approximation in the uniform norm, is stated and proved.

Theorem 3

Let A be a (n+1)-dimensional linear subspace of C[a, b] that satisfies the Haar condition, and let f be any function in C[a, b]. Then p* is the best minimax approximation from A to f, if and only if there exists (n+2) points $\{\xi_i^*; i = 0,1,\ldots,n+1\}$, such that the conditions

$$a \le x_0^* < x_1^* < \ldots < x_{n+1}^* \le b,$$

$$| f(x_i^*) - p^*(x_i^*) \| = \| f - p^* \|_\infty \quad i = 0,1,\ldots,n+1,$$

and

$$f(x_{i+1}^*) - p^*(x_{i+1}^*) = -[f(x_i^*) - p^*(x_i^*)] \quad i = 0,1,\ldots,n$$

are obtained.

The functions in A satisfying the Haar condition are continuous functions that satisfy the same properties concerning zeros as nth degree polynomials, i.e., if an nth degree polynomial has more than n zeros then it is identically zero. In the case of the two-dimensional subspace L of $C^{(1)}$[a,

$$a \le x_0 < x_1 < x_2 \le b$$

$$| f(x_i) - l_0(x_i) | = \| f - l_0 \|_\infty \qquad i = 0,1,2$$

b], l_0 is the best minimax approximation from L to f for any f belonging to $C^{(1)}$[a, b] if and only if there exists 3 points $\{\xi_0,\xi_1,\xi_2\}$such that the conditions
and

$$f(x_i) - l_0(x_i) = -(f(x_{i+1}) - l_0(x_{i+1})), \quad i = 0,1$$

are obtained. This latter criterion is used to verify the output from computer programs. A candidate line l_0 becomes the best linear approximation in the uniform norm to a given function f belonging to $C^{(1)}$[a, b] when three points ξ_0, ξ_1, ξ_2 can be found in the interval [-m/2,m/2] at which $|f-l_0|$ achieves its maximum and at these points f(x)-l(x) changes alternately in sign.

3.2 ***The exchange algorithm***

An iterative exchange algorithm calculates the best linear approximation in the uniform norm numerically. The recommended way of starting this algorithm, see [Powell, 1981], page 91 is to define

$$t_i = \frac{a+b}{2} + (\frac{b-a}{2})\cos(\frac{(n+1-i)p}{(n+1)}), \qquad i = 0,1,\ldots,n+1$$

when iteration takes place in $C^{(1)}$[a, b]. In the case that a = -m/2, b = m/2, the iterations become

$$t_i = \frac{m}{2}\cos(\frac{(2-i)p}{2}), \quad i = 0,1,2.$$

It is then necessary to solve the equations

$$\begin{aligned} f(t_0) - d_0 - d_1 t_0 &= h \\ f(t_1) - d_0 - d_1 t_1 &= -h \qquad \textit{first condition} \\ f(t_2) - d_0 - d_1 t_2 &= h \end{aligned}$$

and

$$\begin{aligned} f(t_0) - d_0 - d_1 t_0 &= -h \\ f(t_1) - d_0 - d_1 t_1 &= h \qquad \text{sec}\textit{ond condition} \\ f(t_2) - d_0 - d_1 t_2 &= -h. \end{aligned}$$

The solution of the first set of equations is

$$d_1 = \frac{f(t_2) - f(t_0)}{t_2 - t_0}, h = \frac{d_1(t_1 - t_0) - [f(t_1) - f(t_0)]}{2}, d_0 = f(t_0) - h - d_1 t_0.$$

If the value of h in the above equation is negative, this signifies that the first set of equations does not correspond to the physical reality. In other words, the approximating line initially passes over the curve f(t) instead of going under it. In this case, the second set of equations is solved giving the solutions

$$d_1 = \frac{f(t_2) - f(t_0)}{t_2 - t_0}, h = \frac{f(t_1) - f(t_0) - [d_1(t_1 - t_0)]}{2}, d_0 = f(t_0) + h - d_1 t_0.$$

It is to be noted that if h is negative in the first set of equations then it will be positive in the second set because the expression for h in the second set is exactly the negative of the expression for h in the first set. The values of d_0 and d_1 calculated here determine the coefficients of the mean line of advance at the current iteration. In order to update the values of d_0 and d_1, the error function and its derivative are defined by the following equations.

$$e(t) = f(t) - (d_0 + d_1 t); e^{(1)}(t) = f^{(1)}(t) - d_1$$

A partition P of grid length Δt is placed in the interval [-m/2,m/2]. It consists of the points p_0, p_1, p_2,…,p_n where

$$p_n = p_0 + n\Delta t$$

Denote

$$\max_{i\in\{0,1,2,\ldots,n\}} | e(p_i) |$$

by B. Then either $e(p_k) = B$ or $-e(p_k) = B$ for some k∈{0,1,2,…,n}. A search will be made in the interval [-m/2,m/2] to find λ∈[-m/2,m/2] such that

$$| e(l) | = \max_{-m/2\le t\le m/2} | e(t) |.$$

$$e(t) = e(p_j) + e^{(1)}(c)(t - p_j)$$

$$\max_{-m/2\le t\le m/2} | f^{(1)}(t) - d_1 |$$

$$e(p_j) \le B - D_1\Delta t$$

In order to do this, it is to be noted that if $e(p_k) = B$ and t∈[p_{j-1},p_{j+1}] where j≠k, then
for c∈[p_j,t] or c∈[t,p_j] by the mean value theorem for differentiable functions. Now denote
by D_1. If the centre point of the interval [p_{j-1},p_{j+1}] satisfies the condition
then for any t∈[p_{j-1},p_{j+1}], e(t) satisfies
since

$$e(t) \le B - D_1\Delta t + e^{(1)}(c)(t - p_j)$$

$$\le B$$

$$| e(t) | > B$$

$$| t - p_j | \le \Delta t, | e^{(1)}(c) | \le D_1.$$

It, therefore, follows that if one wishes to find a t∈[-m/2,m/2] such that
then one should only search intervals of the form [p_{j-1},p_{j+1}] whose centre point p_j satisfies
In each of the intervals satisfying the above criterion, a binary search is conducted to find the

$$e(p_j) > B - D_1\Delta t.$$

largest value of |e(t)| in the subinterval [p_{j-1}, p_{j+1}]. In the case that $-e(p_i) = B$, consider the

$$\bar{e}(t) = -e(t) = d_0 + d_1 t - f(t)$$

function
and calculate
The condition becomes if
then

$$D_2 = \max_{t\in[-m/2,m/2]} | d_1 - f^{(1)}(t) | = D_1.$$

$$\bar{e}(p_j) \le B - D_1\Delta t$$

$$\bar{e}(t) \le B$$

for all $t \in [p_{j-1},p_{j+1}]$. Now, the following algorithm generates a sequence of points which converges to a local maximum of $|e(t)|$ in the interval $[p_{j-1},p_{j+1}]$

A1: Compare $|e(p_{j-1})|$ with $|e(p_{j+1})|$.

If $|e(p_{j-1})| > |e(p_{j+1})|$, put $z_0 = p_{j-1}, z_1 = (p_{j-1}+ p_{j+1})/2$

Else if $|e(p_{j+1})|>|e(p_{j-1})|$, put $z_0=p_{j+1}$, $z_1=(p_{j-1}+p_{j+1})/2$

A2: For n=0,1,2,3,4,5,…

Compare $|e(z_n)|$ with $|e(z_{n+1})|$.

If $|e(z_n)| > |e(z_{n+1})|$, put $z_{n+2} = z_n$, $z_{n+3} = (z_n+ z_{n+1})/2$

Else if $|e(z_{n+1})|>|e(z_n)|$, put $z_{n+2}=z_{n+1}$, $z_{n+3} = (z_n+z_{n+1})/2$

Put tol = $|e(z_{n+3}) – e(z_{n+2})|$ and tolz = $|z_{n+3}-z_{n+2}|$. If either tol<.001 or tolz<.001, stop the iteration at z_{n+2} and put

$$| e(z_{n+2}) |= \max_{p_{j-1}\le t\le p_{j+1}} | e(t) |.$$

z_{n+2} becomes, therefore, the point in $[p_{j-1},p_{j+1}]$ at which $|e(t)|$ attains its maximum. The sequence z_n defined above has the property that

$$| z_n - z_{n+k} |< \frac{1}{2^n} \textit{ for all } k \in Z^+.$$

This means that the sequence is by the nature of its definition a Cauchy sequence and hence converges to a real limit. The maximum value of $|e(t)|$ is then calculated in each interval $[p_{j-1},p_{j+1}]$ for which

$$e(p_j) > B - D_1\Delta t.$$

The global maximum of $|e(t)|$ over the whole interval [-m/2,m/2] is now calculated by finding the largest value of

$$\max_{p_{j-1}\le t\le p_{j+1}} | e(t) |$$

for all j satisfying

$$e(p_j) > B - D_1\Delta t.$$

Denote this largest value by $|e(z^*)|$. If the condition

$$\bar{e}(p_j) \le B - D_1\Delta t$$

is not satisfied, then a similar Cauchy sequence must be generated in $[p_{j-1},p_{j+1}]$ with limit t_j^*.

$$| \bar{e}(t_j^*) |$$

is then calculated for each p_j such that

$$\bar{e}(p_j) > B - D_1\Delta t.$$

Denote the largest of these calculated values by

$$| \bar{e}(t^*) |.$$

Finally, a comparison is made between the value of B, the values of $|e(z^*)|$ and

$$| \bar{e}(t^*) |$$

in order to find the largest one. The largest value and the t*or z* at which it is attained is therefore the value of

$$| e(l) | = \max_{-m/2\le t\le m/2} | e(t) |.$$

The value of λ found in the binary search is now used to update the reference set $\{t_0,t_1,t_2\}$. If the first condition is valid
with

$$e(l) = \max_{-m/2 \le t \le m/2} |e(t)|$$

$$e(t_0) = h; e(t_1) = -h; e(t_2) = h;$$

$$t_0 < l < t_1$$

and
then λ replaces t_0 in the reference set. If the first two conditions hold but
then λ replaces t_2 in the reference set. If the first condition holds but instead
then whether

$$t_1 < l < t_2$$

$$-e(l) = \max_{-m/2 \le t \le m/2} |e(t)|$$

$$t_0 < l < t_1$$

$$t_1 < l < t_2$$

or
λ replaces t_1 in the reference set. If the second condition holds instead of the first condition then similar rules can be deduced for introducing λ into the reference set. It is important to remember that λ has to be introduced into the reference set so that the alternating sign property on consecutive elements is preserved. If the second condition holds

$$e(t_0) = -h; e(t_1) = h; e(t_2) = -h$$

and

$$e(l) = \max_{-m/2 \le t \le m/2} |e(t)|$$

and

$$l < t_0$$

then the reference set becomes $\{\lambda,t_0,t_1\}$. Since $e(\lambda)>0$, $e(t_0)<0$ and $e(t_1)>0$, the error function e(t) changes sign alternatively on different members of the reference set. If the second condition holds and

$$-e(l) = \max_{-m/2 \le t \le m/2} |e(t)|$$

and

$$l < t_0$$

then λ replaces t_0 in the reference set. Similarly, if the first condition holds and

$$l < t_0$$

or

$$l > t_2$$

then similar rules hold for introducing λ into the reference set based on the observation that the error function e(t) must always alternate in sign on consecutive elements of the reference set.

$$|e(l)| = \max_{-m/2 \le t \le m/2} |e(t)|.$$

The binary search on the function e(t) produces a λ such that

This λ is used to update the reference set $\{t_0,t_1,t_2\}$. The new values of the reference set are submitted to the two systems of linear equations to obtain updates of d_0 and d_1, the coefficients of the best linear approximation to f(t). The solution of these linear equations also produces an update to h that is an estimate of the levelled reference error between f(t) and the current line $l(t)= d_0+d_1t$. The function e(t) is now redefined to be

$$e(t) = f(t) - (d_0 + d_1 t)$$

where d_0,d_1 are the updated values of the coefficients of l(t). The procedure described in the preceding pages is then applied to this new e(t). In other words, a binary search is implemented on the current e(t) to find a λ such that

$$| e(l) | = \max_{-m/2 \le t \le m/2} | e(t) |$$

and this λ is used to update the reference set in such a way that the value of e(t) on consecutive elements is alternatively positive and negative. The new values of the reference set are again submitted to the two systems of linear equations to obtain updates of d_0 and d_1, the coefficients of the best linear approximation to f(t). The coefficients d_0 and d_1 are then used to redefine e(t). Each time one goes through the iteration one generates a sequence h_n of levelled reference errors, a sequence of maxima $|e(\lambda_n)|$ and a sequence of line coefficients $d_0^{(n)},d_1^{(n)}$. It turns out that h_n is an increasing sequence

$$h_1 < h_2 < \ldots < h_{n-1} < h_n < \ldots <$$

and if l(t) is the unique best linear approximation in the uniform norm to f(t) on [-m/2, m/2], the sequence h_n converges to $||f-l||_\infty$ in a finite number of steps. The fact that h_n converges to $||f-l||_\infty$ is shown on pages 85-87 of [Powell, 1981]. In addition, there exists a subsequence $|e(\lambda_{nj})|$ such that

$$| e(l_{n_1}) | > | e(l_{n_2}) | > \ldots . > | e(l_{n_{j-1}}) | > | e(l_{n_j}) | > \ldots >$$

and this subsequence converges to $||f-l||_\infty$ in a finite number of steps. The fact that there is always a subsequence of $|e(\lambda_n)|$ that is decreasing and converges to $||f-l||_\infty$ is illustrated on pages 88-90 of [Powell, 1981] and proved on pages 101-102 of [Powell, 1981]. Finally, the sequence of lines

$$l^{(n)}(t) = d_0^{(n)} + d_1^{(n)} t$$

converges uniformly on [-m/2,m/2] to the unique best linear approximation l(t) of f(t) for any continuous function f(t) on [-m/2,m/2]. This fact is proved on pages 99-102 of [Powell, 1981]. If the function f is twice continuously differentiable then the convergence of the iterations

$$l^{(n)}(t) = d_0^{(n)} + d_1^{(n)} t$$

to l(t) is quadratic. In the case of the naval air defence problem, the x and y target accelerations are continuous everywhere except at those points where a radial acceleration is applied to the target trajectory to make it turn suddenly. At each one of these points, the second derivative is piecewise continuous, while the first derivative and the original function $x^{(1)}(t)$, x(t) or $y^{(1)}(t)$,y(t) are continuous everywhere. Therefore, the functions x(t) or y(t) are twice continuously differentiable except at a finite number of points and hence one would expect fairly rapid convergence. In fact, the algorithm converges most of the time within six iterations. The resulting line obtained is called the best linear approximation in the uniform norm and it is the author's way of calculating the mean line of advance of the x and y position time series of a manoeuvring target.

4. **Mean Line of Advance Calculations for Naval Attack Scenarios**

The algorithms described in sections two and three of this article were programmed in C on a Sun Sparc 5 workstation in order to calculate the mean line of advance of a manoeuvring aeroplane. Aeroplane position and velocity data were generated for a scenario where a single aeroplane flies from east to west at speeds varying from 400 to 600 m/s in order to attack a ship that is situated in the west. The x component of aeroplane position increased steadily as the aeroplane moved towards the ship. The y component of aeroplane position varied in the northerly and southerly directions as the aeroplane performed manoeuvres. The mean lines of advance of the y time series of aeroplane position over several time intervals beginning at time = 1 second and ending at time = 9 seconds are shown in Figures 2 to 5. The mean line of advance

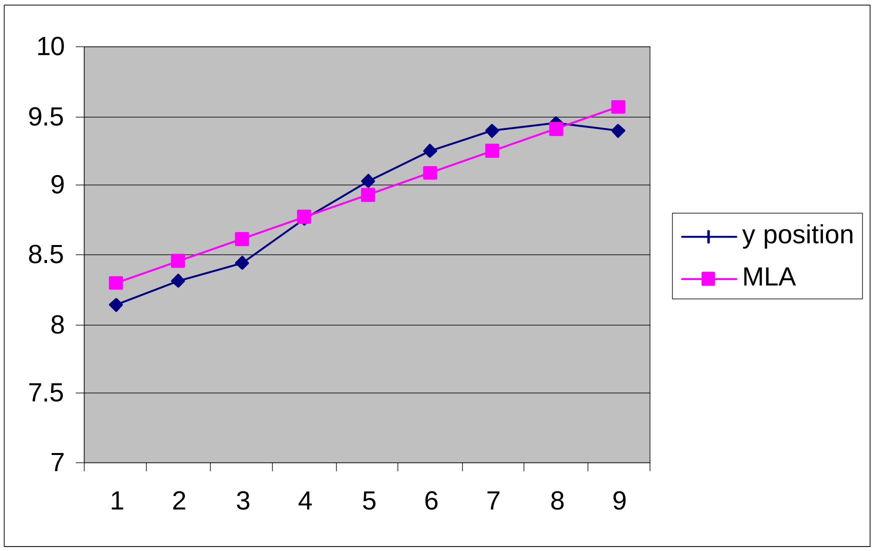

Figure 2 – Mean line of advance of the y time series of position

calculation takes place each time after one new data point is added to the x or y position data set and the data point corresponding to x or y position eight seconds ago is removed. Thus, the mean

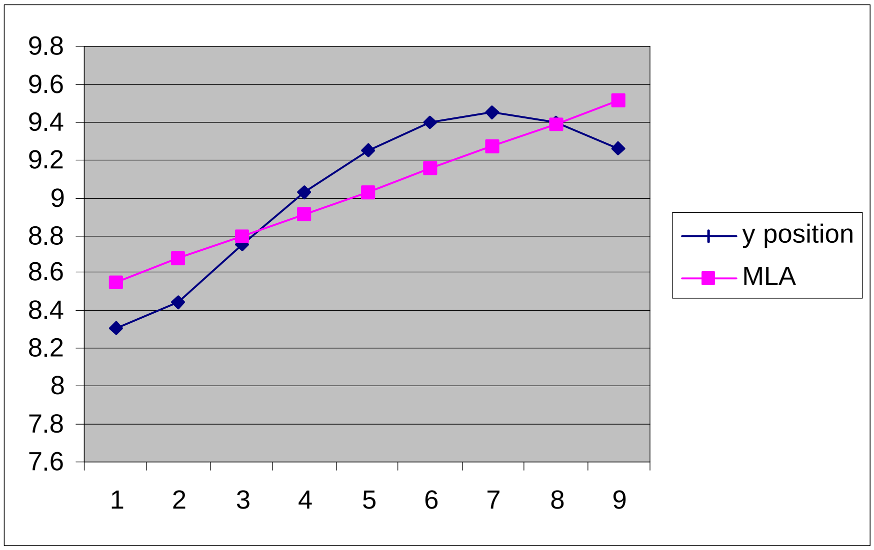

Figure 3 – Mean line of advance of the y time series of position

line of advance is calculated over a moving time window of length 8 seconds. In Figure 2, the y position does not vary much so that the mean line of advance is close to the original curve. In Figure 3, the y position varies more than in the previous figure and hence the levelled reference error between the y position and the mean line of advance increases.

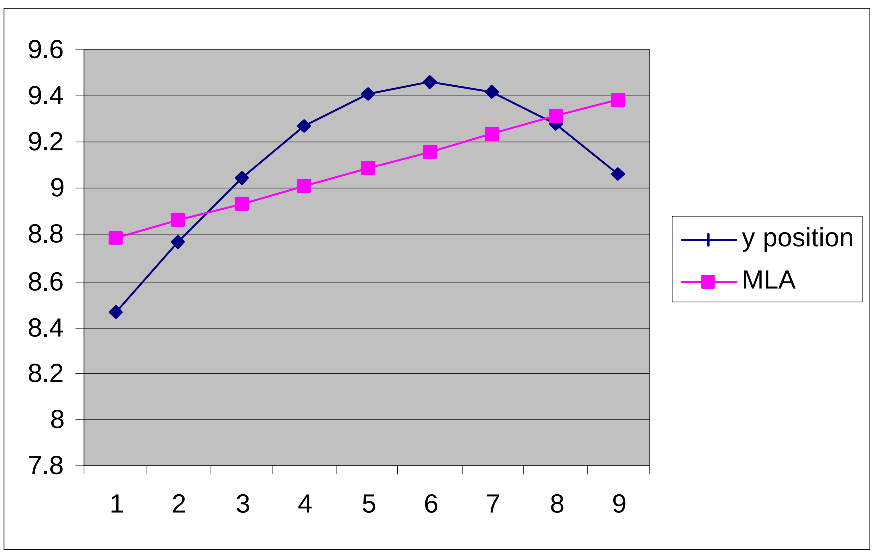

Figure 4 – Mean line of advance of the y time series of position

In Figure 4, the y position of the aeroplane takes the form of a curve without any straight-line segments in it at all. In this case, the levelled reference error between the y position and the mean line of advance is large and the latter is an increasing function of time.

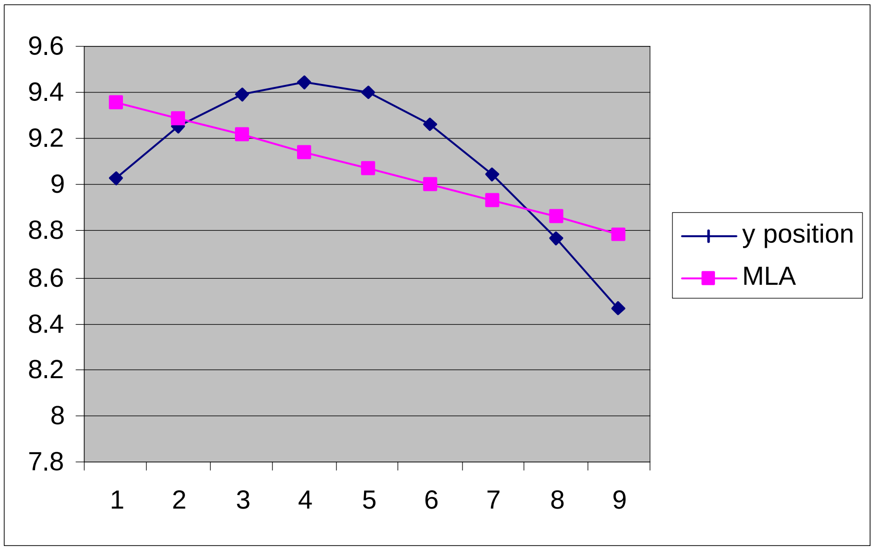

Figure 5 – Mean line of advance of the y time series of position

In Figure 5, several new position data points have been added to the data set and several old position data points have been discarded since the time of the situation shown in Figure 4. The new incoming points in Figure 5 have caused the mean line of advance to become a decreasing function of time.

5. **Conclusion**

In this paper, an algorithm has been described for calculating the mean line of advance of a continuously differentiable function f(t) defined on an interval such as [1,9]. The algorithm consists of two parts. Since the trackers of tactical surveillance radars only provide updates to the target track at discrete times such as every two seconds or every four seconds it was necessary to devise an interpolation algorithm using Chebyshev polynomials in order to express the target x and y time series as continuous functions of time. The second part consists of an iterative exchange algorithm that produces a finite sequence of lines converging uniformly to the mean line of advance. In the second part, it is necessary to calculate the maximum of the difference of the function values f(t) and those of the current line l(t) in the sequence. Even though a continuous approximation p(t) has been found for f(t), it is not true that $p^{(1)}(t)$ is a good approximation of $f^{(1)}(t)$. Hence, one cannot solve the equation $p^{(1)}(t) = l^{(1)}(t)$ to calculate the points at which f(t)-l(t) attains its maximum value. Instead, a binary search of the function p(t)-l(t) is conducted around points that are relatively close to the global maximum in order to determine which point is the global maximum.

The mean line of advance was tested on manoeuvring aeroplane trajectories for an aeroplane travelling at speeds between 400 and 600 m/s. For trajectories that are almost straight lines, the uniform norm mean line of advance tightly adheres to the original trajectory. If the manoeuvres are slight there is a small levelled reference error between the mean line of advance and the aeroplane trajectory. If the manoeuvres consist of many curves, the levelled reference error between the mean line of advance and the original trajectory can be quite large. In fact, the author noticed that if the mean line of advance is calculated for a trajectory consisting of many curves then it may have a positive slope as shown in Figure 4. However, after four more points are added to the data set and the four data points with the oldest timestamps are removed, the mean line of advance can change direction and can now have a negative slope as shown in Figure 5. Thus, the mean line of advance of a multiple curve trajectory can itself oscillate, although it does not oscillate to the same extent as the original curve. These last comments make it necessary to estimate the upper and lower bounds of the variation of the mean line of advance so that if two air threats are flying almost wing tip to wing tip the variation of velocity in each of the mean lines of advance does not cause an inversion of their threat levels.

References

1. Paradis, S, Chalmers, B., Carling, R., Roy, J., and Bossé, É., "Towards a Generic Model for Situation and Threat Assessment", DREV-R-9622, May 1997.
2. Carling, R. L., "Naval Situation Assessment Using a Real-Time Knowledge-Based System", *Naval Engineers Journal*, May 1999, Vol. 111, No. 3, pp.173-187.
3. Parks, T.W., and Burrus, C.S., "Digital Filter Design", Wiley, New York, 1987, pp. 83-106.
4. Hamming, R.W., "Numerical Methods for Scientists and Engineers", Dover Publications, Mineola, New York, 1973, pp. 470-494.
5. Powell, M.J.D., "Approximation Theory and Methods", Cambridge University Press, Cambridge, 1981.
6. Davis, P.J., "Interpolation and Approximation", Blaisdell Publishing Company, Waltham, Massachusetts, 1963.