

## DRAG COEFFICIENT CURVE IDENTIFICATION OF PROJECTILES FROM FLIGHT TESTS VIA OPTIMAL DYNAMIC FITTING<sup>1</sup>

Yangquan Chen\*, Changyun Wen\*, Zhiming Gong\* and Mingxuan Sun\*\*

\*School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Republic of Singapore (ecywen@ntuvax.ntu.ac.sg)

\*\*Department of Electrical Engineering, Xi'an Institute of Technology, Xi'an 710032, P.R. China

(Received March 1996; in final form March 1997)

**Abstract:** Extracting a projectile's optimally fitting drag coefficient curve  $C_{df}(M)$  from flight testing data is considered as an optimal tracking control problem (OTCP), where  $C_{df}(M)$  is regarded as the *control profile*, while the flight testing data is the *desired trajectory* to be optimally tracked. Cubic splines with deficiency number 2, which can guarantee the continuity and fitting flexibility of  $C_{df}(M)$ , are employed to parametrize the  $C_{df}(M)$ . This functional minimization problem (OTCP) can then be converted into a multivariable parametric minimization one. The standard Newton-Raphson iteration is formulated. A quasi-Newton-Raphson iterative scheme is proposed to reduce the cost of computation by half, and performs a near-second-order convergence by using the first-order gradient only. A further simplification is proposed by using the approximated Hamiltonian function, which cuts the computing cost by half once more. A practical testing data-reduction result is presented to show that the optimal dynamic fitting is a useful scheme for aerodynamic coefficient *curve* identification. Copyright © 1997 Elsevier Science Ltd

**Keywords:** Optimal control, data reduction, curve fitting, identification.

### 1. INTRODUCTION

In exterior ballistics, the most important factor concerned is the aerodynamic drag coefficient curve  $C_d(M)$ , which plays a critical role in the firing table generation. An accurate firing table is vital to the accuracy of the artillery attacks. Although one can use wind tunnel tests or numerical aerodynamic property prediction codes to obtain the aerodynamic property curve of a projectile, the results suffer from the errors introduced by interference from the wind tunnel wall

or impractical assumptions made in numerical modeling, and so on. Therefore, identifying the aerodynamic properties of real or full-scale flying vehicles from flight testing data is an important area in modern flight dynamics (Chapman and Kirk, 1970; Lieske and Mackenzie, 1972; Bartelsson, 1975; Ross, 1975; Eulrich and Rynaski, 1975; Williamson, 1980; Gupta and Illif, 1982; Fratter and Stengel, 1983; Anderson and Vincent, 1985; Jiang and Chen, 1986; Chen *et al.*, 1992; Linse and Stengel, 1994). Essentially, this is a combination of modern control theory and flight dynamics. Because, in most of the proving grounds, the doppler tracking radar such as TERMA OPOS DR-582 is becoming a common item of equipment, finding

<sup>1</sup> This paper is partially supported by the National Science Foundation of China under project no. 69404004.

the best means of utilizing the measured data is receiving a lot of attention. This paper concentrates on the identification of a single aerodynamic drag coefficient curve of an artillery projectile from velocity data of flight tests measured by tracking radar.

The easiest and most straightforward method is to apply the direct numerical differentiation (DND) method. The results obtained in this way cannot be applied directly, as measurement noise will be amplified during the differentiation of the measured velocity. However, the result is still useful in the sense that it can provide a reasonable initial trial for other advanced approaches. To get the result more accurately, a *differential correction* method was introduced in (Chapman and Kirk, 1970) by incorporating the ballistic model into a Newton-Raphson-like iteration. It is worth pointing out that the existing literature only concentrates on aerodynamic coefficient extraction by parameter identification, see for example (Bartelson, 1975; Williamson, 1980; Gupta and Illif, 1982; Fratter and Stengel, 1983; Larimore *et al.*, 1985; Illif, 1985, 1989; Anderson and Vincent, 1985). Surveys on the parametric identification can be found in (Illif and Maine, 1986; Peter, 1989). In fact, from the intrinsic characteristics of aerodynamics (Tobak and Schiff, 1981), aerodynamic property curves (Mach history) can be regarded as deterministic *control profiles*, and aerodynamic property identification can be considered as an *optimal tracking control problem* (OTCP) where the collected testing data are the desired trajectories to be optimally tracked. However, this kind of *profile optimization problem* cannot be directly solved by numerical methods in standard optimal control because it is a singular optimal control problem. Moreover, it has also been found that the aerodynamic property time history cannot be directly applied (Chen *et al.*, 1992; Chen and Dou, 1993a). This is because the resulting aerodynamic coefficient may be a multi-valued function of the Mach number. This is inconsistent with the theory of aerodynamics, when the ballistic trajectory of a flight test has an ascending part and a descending part. Thus, solving this singular optimal control profile problem provides the motivation to explore new schemes.

The idea of a scheme termed *optimal dynamic fitting*, developed by Chen *et al.* (1992), can be applied to solve the OTCP mentioned above. Its fundamental approach is to convert the functional minimization problem into a multivariable parametric minimization problem, where the control profile is appropriately parametrized and then an effective optimization algorithm is applied. A similar idea was also used in (Messner and

Horowitz, 1993), where, through an integral representation of the control profile, the functional basis is considered as *the regressor* and the influence function is regarded as "*parameters*" to be optimally sought. In this paper, the control profile  $C_{df}(M)$  is parametrized by cubic splines with deficiency number 2. Thus the first-order derivative of  $C_{df}(M)$  is guaranteed to be continuous. To search for the optimal parameters, a standard Newton-Raphson iteration is applied. To reduce the complexity of the algorithm, an idea termed "quasi-Newton-Raphson iteration" is introduced. This can achieve near-second-order convergence and cut the computing cost by half even if only the first-order gradient is used. By utilizing the approximated Hamiltonian function, the computing cost can be further reduced by half or more. In addition, the initial system states for flight testing, which may be uncertain or inaccurate, can be easily identified or corrected, together with the optimal dynamic fitting. As real flight testing was conducted to obtain the data for aerodynamic curve identification, the authors believe that the results obtained are rather convincing in applications including the verification and improvement of design objectives, validation of computational aerodynamic property prediction codes, and so on.

It should also be pointed out that, from the aerodynamic point of view, the identified  $C_{df}(M)$  curve cannot be taken as a zero lift drag coefficient curve or an incidence induced drag coefficient curve. The  $C_{df}(M)$  is just the *fitting drag coefficient curve* with respect to the trajectory model employed, which comprehensively reflects the effects of zero lift drag and the drag induced by the angular motion around the center of mass. Because the generation of the firing table is mainly based on a single ballistic coefficient, a drag law, and some fitting factors, more accurate firing tables can be produced when the  $C_{df}(M)$  identified in this paper is utilized directly.

The remaining parts of this paper are organized as follows. The problem formulation is given in Section 2. Two elements of the optimal dynamic fitting application, i.e., cubic spline parametrization and Newton-Raphson iteration, are presented in Section 3. Some simplifications utilized in this optimal dynamic fitting problem, i.e., the quasi-Newton-Raphson scheme and the approximate Hamiltonian function method, are introduced in Section 4. The correction of inaccurate initial conditions is discussed in Section 5. In Section 6, an actual flight-testing data-reduction result is given to show the effectiveness of the dynamic fitting method. Section 7 concludes this paper. A complete set of actual flight testing data, as well as the

testing conditions, are given in the APPENDIX.

## 2. PROBLEM FORMULATION

For brevity, a 3-DOF point mass ballistic model is used in this paper. A more complete ballistic model (Chen, 1990a,b, 1991) may be used if other aerodynamic coefficients are available, and the method presented here can still be applied. Suppose that at time  $t$ , the position of the projectile  $P$  in the earth coordinate system (ECS) is  $[x(t), y(t), z(t)]^T$ , and its relative velocity vector  $\vec{u}$  w.r.t. ECS is  $[u_x(t), u_y(t), u_z(t)]^T$ . The position of the radar  $R$  in ECS is  $[x_r(t), y_r(t), z_r(t)]^T$ , which is known from the testing setup as shown in Fig. 1.

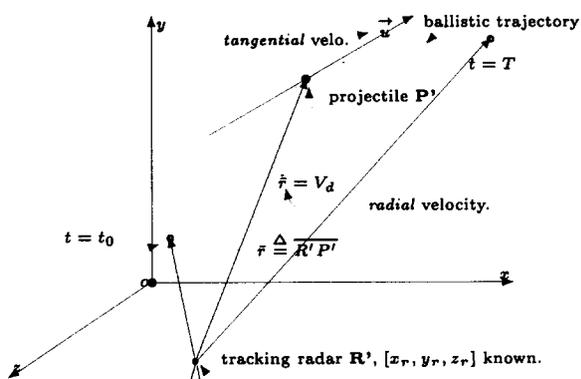


Fig. 1. Illustration of Doppler Radar Tracking

The 3-DOF point mass trajectory model can be described by nonlinear state space equations as follows:

$$\begin{cases} \dot{u}_x = f_1(X(t), C_{df}) = -\frac{\rho s V}{2m} (u_x - w_x) C_{df} \\ \dot{u}_y = f_2(X(t), C_{df}) = -\frac{\rho s V}{2m} u_y C_{df} - g \\ \dot{u}_z = f_3(X(t), C_{df}) = -\frac{\rho s V}{2m} (u_z - w_z) C_{df} \\ \dot{x} = f_4(X(t), C_{df}) = u_x \\ \dot{y} = f_5(X(t), C_{df}) = u_y \\ \dot{z} = f_6(X(t), C_{df}) = u_z \end{cases} \quad (1)$$

where  $X(t) = [u_x, u_y, u_z, x, y, z]^T$  is the state vector of system (1);  $t \in [0, T]$ ,  $T$  is known and the initial state  $X(0)$  may not be exactly known;  $g$  is the gravitational acceleration;  $w_x, w_z$  are the wind components in ECS known from meteorological measurements;  $V$  is projectile's relative velocity w.r.t. the wind and

$$V = \sqrt{(u_x - w_x)^2 + u_y^2 + (u_z - w_z)^2}; \quad (2)$$

$\rho$  is the air density;  $s = \pi d^2/4$  is the reference area of the projectile and  $d$  is the projectile's diameter;  $m$  is the mass of the projectile;  $C_{df}(M)$  is the fitting drag coefficient curve w.r.t. the trajectory

model (1), which is regarded as the control profile. The Mach number  $M \triangleq V/a$ , where  $a$  is the local sonic speed.

The drag coefficient  $C_{df}$  in the 3-DOF point-mass model (1) should be the combined effects of the zero-lift drag coefficient  $C_{d0}$  and the angle-of-attack induced drag coefficient  $C_{d_{\alpha^2}}$ , i.e.,

$$C_{df} = C_{d0} + C_{d_{\alpha^2}} \alpha^2$$

where  $\alpha$  is the angle of attack. To accurately predict  $\alpha$ , all aerodynamic coefficients must be exactly known and a full rigid-body 6-DOF model (Chen, 1991) should be used. In this situation, the curve identification methods of this paper are still applicable. However, the computational cost will be unnecessarily high. As far as the aerodynamic identification problem of this paper is concerned, a full rigid-body 6-DOF model is unnecessary, based on the following observations:

- O1) From the causality considerations, the velocity is directly affected by drag force while the motion around center of mass has an indirect effect on the velocity.
- O2) For a spin-stabilized projectile, the maximal angle of attack along the whole ballistic trajectory is normally small, e.g., around  $10^\circ$ .

It is worth stressing again that the above "control profile"  $C_{df}(M)$  cannot only be taken as a zero lift drag coefficient curve or an incidence-induced drag coefficient curve. The  $C_{df}(M)$  is the *fitting drag coefficient curve* with respect to the ballistic model used. Because the  $C_{df}(M)$  comprehensively reflects the effects of zero lift drag and the drag induced by the angular motion around the center of mass, the 3-DOF point mass model (1) describes the projectile's movement better than the case when the  $C_{df}(M)$  is replaced by a ballistic factor.

Denote  $r(t)$  as the distance between the tracking radar  $R$  and the projectile  $P$ , and  $v_r(t)$  as the velocity data, measured using doppler radar.

$$v_r(t) = \dot{r}(t) \quad (3)$$

To solve the optimization problem conveniently, the projectile's tangential velocity  $\vec{u}$  must be transformed into  $v_d$ , which is in the doppler radar's radial sense, i.e., the  $\vec{RP}$  direction. Let

$$[r_x, r_y, r_z] = [x - x_r, y - y_r, z - z_r]. \quad (4)$$

Then

$$r = \sqrt{r_x^2 + r_y^2 + r_z^2} \quad (5)$$

and  $v_d$  is given to be

$$v_d = (u_x r_x + u_y r_y + u_z r_z) / r \quad (6)$$

which has the same implication as that of  $v_r(t)$ . Equation (6) can be taken as the output equation of system (1). Define the functional performance index as

$$J[C_{df}(M)] = \int_0^T L(C_{df}(M), X(t), t) dt \quad (7)$$

where

$$L(C_{df}(M), X(t), t) = (v_d(t) - v_r(t))^2,$$

and assume a free terminal condition. It can be seen that equations (1), (6), and (7) formulate an optimal tracking control problem (OTCP), where the doppler radar measured velocity data  $v_r(t)$  is the desired trajectory to be optimally tracked, and the *fitting drag coefficient curve*  $C_{df}(M)$  is the *control profile* to be determined by minimizing  $J[C_{df}(M)]$ . The Hamiltonian function  $H(C_{df}(M), X(t), t)$  is

$$H = L + \lambda^T f \quad (8)$$

where  $L$  is given in (7) and  $f = [f_1, f_2, \dots, f_6]^T$  is defined in (1).  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_6]^T$  is the co-state vector which can be obtained from

$$\dot{\lambda} = -\frac{\partial H}{\partial X} = -\frac{\partial L}{\partial X} - \lambda^T \frac{\partial f}{\partial X}, \quad (\lambda(t_f) = 0). \quad (9)$$

Obviously, the above OTCP is a singular one because  $\partial H / \partial C_{df}$  does not explicitly contain the  $C_{df}$  term. Difficulties will arise in the numerical computation for this singular optimal control problem (SOCP). Furthermore, the correction of inaccurate  $X(0)$  will not be easily made using the standard numerical methods for solving optimal control problems. The optimal dynamic fitting method proposed in the following section can overcome the above problems.

### 3. OPTIMAL DYNAMIC FITTING

The optimal dynamic fitting method can be described in Fig. 2.

To apply the dynamic fitting method, the first thing to do is to determine the parametrization of the control profile. Then an efficient parameter-searching method is to be employed. In this paper, the control profile is parametrized by using cubic splines with deficiency number 2, and the parameter-searching method is the Newton-Raphson iteration, which are described in detail as follows.

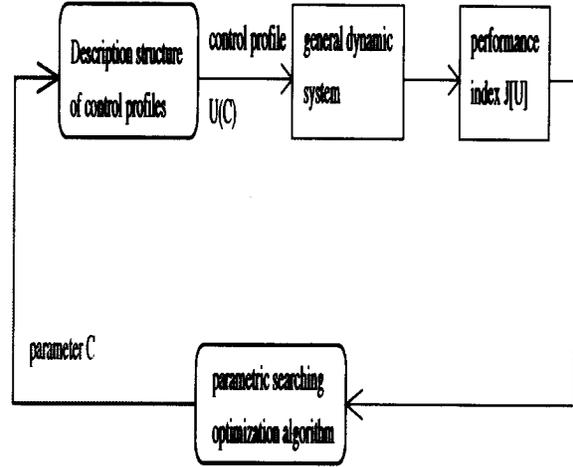


Fig. 2. Optimal dynamic fitting method

#### 3.1 Parametrization of $C_{df}(M)$

Cubic splines with deficiency number 2 are used to parametrize the control profile  $C_{df}(M)$ . The idea of Hermite interpolation is employed. Suppose that the  $C_{df}(M)$ , where  $M \in [M_0, M_f]$ , and  $[M_0, M_f]$  is divided into  $n$  segments with  $n + 1$  knots  $M_0 = M_1 < M_2 < \dots < M_n < M_{n+1} = M_f$ .  $[M_0, M_f]$  can be estimated to cover the practical Mach range. Denote  $f_i, d_i$  as the function value and the first-order derivative of  $C_{df}(M)$  at the  $i$ -th knot  $M_i$  respectively. Consider the interval  $[M_i, M_{i+1}]$  and suppose  $M \in [M_i, M_{i+1}]$ . Then a Hermite-type polynomial  $C_{df_i}(M)$  can be uniquely determined by  $f_i, d_i, f_{i+1}, d_{i+1}$ , i.e.,

$$\begin{aligned} C_{df_i}(M) &= [\gamma_1(\tau_i), \gamma_2(\tau_i), \gamma_3(\tau_i), \gamma_4(\tau_i)] \\ &\quad [f_i, d_i, f_{i+1}, d_{i+1}]^T \\ &= [\alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}] [1, M, M^2, M^3]^T \\ &= [\beta_{1i}, \beta_{2i}, \beta_{3i}, \beta_{4i}] [1, \tau_i, \tau_i^2, \tau_i^3]^T \quad (10) \end{aligned}$$

where  $\tau_i = (M - M_i) / h_i$ ,  $h_i = M_{i+1} - M_i$ .

Obviously  $\tau_i \in [0, 1]$ . One can also easily get

$$A_1 [\beta_{1i}, \beta_{2i}, \beta_{3i}, \beta_{4i}]^T = [f_i, d_i, f_{i+1}, d_{i+1}]^T \quad (11)$$

where

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & h_i^{-1} & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & h_i^{-1} & 2h_i^{-1} & 3h_i^{-1} \end{bmatrix}, \\ A_1^{-1} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & h_i & 0 & 0 \\ -3 & -2h_i & 3 & -h_i \\ 2 & h_i & -2 & h_i \end{bmatrix}. \end{aligned}$$

Then,

$$\begin{aligned} \begin{bmatrix} \gamma_1(\tau_i) \\ \gamma_2(\tau_i) \\ \gamma_3(\tau_i) \\ \gamma_4(\tau_i) \end{bmatrix}^T &= [1, \tau_i, \tau_i^2, \tau_i^3] A_1^{-1} \\ &= \begin{bmatrix} 1 - 3\tau_i^2 + 2\tau_i^3 \\ h_i(\tau_i - 2\tau_i^2 + \tau_i^3) \\ 3\tau_i^2 - 2\tau_i^3 \\ h_i(-\tau_i^2 + \tau_i^3) \end{bmatrix}^T. \end{aligned} \quad (12)$$

Clearly,

$$[\beta_{1i}, \beta_{2i}, \beta_{3i}, \beta_{4i}]^T = A_2[\alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}]^T \quad (13)$$

where

$$A_2 = \begin{bmatrix} 1 & M_i & M_i^2 & M_i^3 \\ 0 & h_i & 2h_i M_i & 3h_i M_i^2 \\ 0 & 0 & h_i^2 & 3h_i^2 M_i \\ 0 & 0 & 0 & h_i^3 \end{bmatrix},$$

$$A_2^{-1} = \begin{bmatrix} 1 & -h_i^{-1} M_i & h_i^{-2} M_i^2 & -h_i^{-3} M_i^3 \\ 0 & h_i^{-1} & -2h_i^{-2} M_i & 3h_i^{-3} M_i^3 \\ 0 & 0 & -h_i^{-2} & -3h_i^{-3} M_i \\ 0 & 0 & 0 & h_i^{-3} \end{bmatrix}.$$

So, the curve  $C_{df}(M) \in C^1[M_0, M_f]$ , i.e.,  $C_{df}(M)$  is a smooth function. As the number of segments  $n$  and the knots can be determined by the user, sufficient flexibility of the parametric description of  $C_{df}(M)$  can be guaranteed. On the other hand, a spline description with low-order smoothness has fewer parameters to be searched, and is an effective description method in most applications. Moreover, explicit expressions in terms of either  $M$  or  $\tau_i$  can be readily obtained from (10)-(13). For example, when  $M$  is used, for  $M \in [M_i, M_{i+1}]$ , we have

$$C_{df_i}(M) = [\alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}][1, M, M^2, M^3]^T.$$

where, from (10)-(13),

$$[\alpha_{1i}, \alpha_{2i}, \alpha_{3i}, \alpha_{4i}] = [f_i, d_i, f_{i+1}, d_{i+1}](A_1^{-1})^T (A_2^{-1})^T.$$

The detailed form can be easily obtained as follows:

$$\begin{cases} \alpha_{1i} = \beta_{1i} - h_i^{-1} M_i \beta_{2i} + h_i^{-2} M_i^2 \beta_{3i} - h_i^{-3} M_i^3 \beta_{4i} \\ \alpha_{2i} = h_i^{-1} \beta_{2i} - 2h_i^{-2} M_i \beta_{3i} + 3h_i^{-3} M_i^3 \beta_{4i} \\ \alpha_{3i} = -h_i^{-2} \beta_{3i} - 3h_i^{-3} M_i \beta_{4i} \\ \alpha_{4i} = h_i^{-3} \beta_{4i} \end{cases}$$

$$\begin{cases} \beta_{1i} = f_i \\ \beta_{2i} = h_i d_i \\ \beta_{3i} = -3f_i - 2h_i d_i + 3f_{i+1} - h_i d_{i+1} \\ \beta_{4i} = 2f_i + h_i d_i - 2f_{i+1} + h_i d_{i+1} \end{cases}$$

Now denote

$$C^T = [c_1, c_2, \dots, c_{2n+2}] \quad (14)$$

where  $c_{2i-1} = f_i$ ,  $c_{2i} = d_i$ , ( $i = 1, 2, \dots, n + 1$ ). The functional minimization problem

$$\min_{C_{df}} J[C_{df}]$$

can then be converted into a multivariable parametric optimization problem, i.e.

$$\min_C J[C_{df}].$$

A class of direct search methods can be used to solve the above problem. However, this will be less effective when  $n$  is large. In most applications, an initial guess at the control profile  $C_{df}^{(0)}(M)$  can be obtained within a feasible region. Thus, the Newton-Raphson iteration is preferred and  $\partial J/\partial c_i$  and  $\partial^2 J/\partial c_i \partial c_j$  ( $i, j = 1, 2, \dots, 2n + 2$ ) are to be determined. In what follows,  $\partial J/\partial c_i$  is denoted by  $\varphi_i$  for brevity.

### 3.2 Newton-Raphson Iteration

From the optimal control theory, the optimality condition of the OTCP is

$$\frac{\partial H}{\partial C_{df}} = \frac{\partial L}{\partial C_{df}} + \frac{\partial \lambda^T f}{\partial C_{df}} = 0. \quad (15)$$

From (15),

$$\frac{\partial L}{\partial C_{df}} = -\frac{\partial \lambda^T f}{\partial C_{df}} \quad (16)$$

i.e.,

$$\frac{\partial L}{\partial c_i} = -\frac{\partial \lambda^T f}{\partial C_{df}} \frac{\partial C_{df}}{\partial c_i}. \quad (17)$$

So, integrating (17) from 0 to  $T$  w.r.t. time  $t$  yields

$$\begin{aligned} \varphi_i &= -\int_0^T \frac{\partial \lambda^T f}{\partial C_{df}} \frac{\partial C_{df}}{\partial c_i} dt \\ &= -\int_0^T \frac{\rho s V}{2m} [\lambda_1(u_x - w_x) + \lambda_2 u_y + \lambda_3(u_x - w_x)] \frac{\partial C_{df}}{\partial c_i} dt. \end{aligned} \quad (18)$$

Noticing that

$$\frac{\partial V}{\partial c_i} = \frac{u_x - w_x}{V} \frac{\partial u_x}{\partial c_i} + \frac{u_y}{V} \frac{\partial u_y}{\partial c_i} + \frac{u_z - w_z}{V} \frac{\partial u_z}{\partial c_i}$$

and ignoring the effect of  $\partial \rho/\partial y$ ,

$$\frac{\partial \varphi_i}{\partial c_i} = \frac{\partial^2 J}{\partial c_i \partial c_j} = -\int_0^T \frac{\rho s V}{2m} \left[ \left( \frac{\partial V}{\partial c_j} \lambda_1 + V \frac{\partial \lambda_1}{\partial c_j} \right) \right]$$

$$\begin{aligned}
& (u_x - w_x) + \left( \frac{\partial V}{\partial c_j} \lambda_2 + V \frac{\partial \lambda_2}{\partial c_j} \right) u_y \\
& + \left( \frac{\partial V}{\partial c_j} \lambda_3 + V \frac{\partial \lambda_3}{\partial c_j} \right) (u_z - w_z) + \lambda_1 \frac{\partial u_x}{\partial c_j} \\
& + \lambda_2 \frac{\partial u_y}{\partial c_j} + \lambda_3 \frac{\partial u_z}{\partial c_j} \left] \frac{\partial C_{df}}{\partial c_i} dt \quad (19)
\end{aligned}$$

where  $i, j = 1, 2, \dots, 2n + 2$ . Partial differentiation of (1) w.r.t.  $c_j$  ( $j = 1, 2, \dots, 2n + 2$ ) gives

$$\frac{d}{dt} \left( \frac{\partial X}{\partial c_j} \right) = \frac{\partial f}{\partial c_j}, \quad \left. \frac{\partial X}{\partial c_j} \right|_{t=t_0} = 0. \quad (20)$$

Similarly, by partially differentiating (9) w.r.t.  $c_j$  ( $j = 1, 2, \dots, 2n + 2$ ), the d.e.'s of  $\partial \lambda / \partial c_j$  can be obtained,

$$\frac{d}{dt} \left( \frac{\partial \lambda}{\partial c_j} \right) = \frac{\partial H}{\partial c_j}, \quad \left. \frac{\partial \lambda}{\partial c_j} \right|_{t=t_f} = 0. \quad (21)$$

By integrating (1) and (20) simultaneously from 0 to  $T$  (with the initial conditions given in (1) and (20) respectively), then integrating (9) and (21) from  $T$  to 0 (with the final conditions given in (9) and (21) respectively),  $\partial \varphi_i / \partial c_i$  and  $\varphi_i$  in (18) and (19) can be computed easily by a proper quadrature algorithm. Therefore, the Newton-Raphson iteration can be applied, i.e.,

$$\begin{cases} C^{(k+1)} = C^{(k)} + \delta C^{(k)} \\ A^{(k)} \delta C^{(k)} = B^{(k)} \end{cases} \quad (22)$$

where  $k$  is the iteration number;  $\delta C^{(k)}$  is the optimal increment of  $C^{(k)}$  at the  $k$ -th iteration. Let  $a_{i,j}$  and  $b_i$  the elements of matrices  $A^{(k)}$  and  $B^{(k)}$  respectively; it is observed that

$$\begin{cases} a_{i,j} = \frac{\partial \varphi_i}{\partial c_j}, \\ b_i = -\varphi_i, \quad i, j = 1, 2, \dots, 2n + 2. \end{cases} \quad (23)$$

Attention must be paid to the calculation of  $\partial C_{df} / \partial c_i$ . From (10) and (12),  $\partial C_{df} / \partial c_i$  does not equal 0 in  $[M_k, M_{k+1}]$  only, where  $k = [(i + 1)/2]$ , and  $[\cdot]$  denotes an integer truncating operation, i.e.,

$$\frac{\partial C_{df}}{\partial c_i} = \begin{cases} 0, & M \notin [M_k, M_{k+1}], \forall i; \\ \gamma_1(\tau_k), & M \in [M_k, M_{k+1}], i \text{ is odd}; \\ \gamma_2(\tau_k), & M \in [M_k, M_{k+1}], i \text{ is even}. \end{cases}$$

Furthermore, it is known that the  $M(t)$  curve is usually a  $U$ -type, where the minimal Mach number point corresponds to the trajectory apogee. Thus, when  $C_{df}(M)$  is parametrized w.r.t.  $M$ , one must consider the ascending and descending parts of a ballistic trajectory separately in the process of forward and backward integration. Suppose  $M \in [M_k, M_{k+1}]$ . If the current Mach number  $M$  is in the descending part of the trajectory, it

is known from the causality relations that in the forward integration of (1) and (20)

$$\frac{d}{dt} \left( \frac{\partial X}{\partial c_i} \right) = 0, \quad i = 2k + 3, 2k + 4, \dots, 2n + 2 \quad (24)$$

while in the backward integration of (9) and (21)

$$\frac{d}{dt} \left( \frac{\partial \lambda}{\partial c_i} \right) = 0, \quad i = 1, 2, \dots, 2k - 2. \quad (25)$$

Similarly, if the current Mach number  $M$  belongs to the ascending trajectory, it is necessary to set

$$\frac{d}{dt} \left( \frac{\partial X}{\partial c_i} \right) = 0, \quad i = 1, 2, \dots, 2k - 2 \quad (26)$$

in the forward integration of (1) and (20), and

$$\frac{d}{dt} \left( \frac{\partial \lambda}{\partial c_i} \right) = 0, \quad i = 2k + 3, 2k + 4, \dots, 2n + 2 \quad (27)$$

in the backward integration of (9) and (21).

#### 4. SOME PRACTICAL SIMPLIFICATIONS

The Newton-Raphson iteration (22) will converge rapidly in a second-order sense if a proper initial point  $C^{(0)}$  can be chosen. A rough but reasonable  $C_{df}^{(0)}(M)$  can be obtained from (1) by direct numerical differentiation (DND), but larger errors will be introduced. So, to get this  $C_{df}^{(0)}$  with smaller errors, data-fitting-based DND techniques can be used with care. The method and software of (Chen and Dou, 1991) are helpful.

From the last section, it is noted that  $12(n + 1) + 6$  d.e.'s are to be integrated in the forward or the backward direction, i.e., for Newton-Raphson iteration (22), there is a total of  $24(n + 1) + 12$  d.e.'s to be integrated. With a larger  $n$ , the burdensome computation in forward or backward integration is obvious. Based on the fact that, as discussed before, a proper  $C^{(0)}$  can be obtained by the method described in this paper, some simplifications can be made to reduce the computational cost. One idea is to avoid the backward integration pass. Another way is to use an approximated Hamiltonian function.

##### 4.1 Quasi-Newton-Raphson Iteration

To avoid the backward pass, by taking a partial derivative w.r.t.  $c_i$  directly in (7),

$$\varphi_i = \frac{\partial J}{\partial c_i} = 2 \int_0^T (v_d(t) - v_r(t)) \frac{\partial v_d(t)}{\partial c_i} dt \quad (28)$$

where  $\partial v_d(t)/\partial c_i$  can be determined from (6) and from  $\partial X(t)/\partial c_i$ . From (28), let

$$\frac{\partial \varphi_i}{\partial c_j} \approx 2 \int_0^T 2 \frac{\partial v_d(t)}{\partial c_i} \frac{\partial v_d(t)}{\partial c_j} dt. \quad (29)$$

Replacing equations (18) and (19) with (28) and (29) respectively, forms the quasi-Newton-Raphson iteration which was called *differential correction* in (Chapman and Kirk, 1970). In (29), the terms related to the second-order partial derivatives, i.e.,  $2(v_d(t) - v_r(t))\partial^2 v_d/\partial c_i \partial c_j$ , are ignored. Since  $C^{(0)}$  can be properly chosen to make the iteration converge, the effect of these ignored terms will be smaller and smaller, and the convergence approaches the second-order sense. Although this simplification needs a few more iterations than that of iteration (22), the computing cost can be cut down by half because the backward integration is not required. Thus, the above quasi-Newton-Raphson iteration is applicable and attractive.

The explicit assumption used in this scheme is that a proper  $C^{(0)}$  could be chosen to make the iteration convergent. In the specific problem of this paper, this assumption is feasible in practice as discussed in Section 6.

#### 4.2 Using an approximated Hamiltonian function

The main starting point for utilizing an approximated Hamiltonian function to simplify the computation is based on the simple fact that velocity is mainly affected by aerodynamic drag. By introducing the *dominant state*  $u$ , one can re-write the Hamiltonian function (8) in an approximated form as

$$\tilde{H} = \tilde{L} + \tilde{\lambda} \left\{ -\frac{\rho s}{2m} V^2 C_{df} - g \sin(\theta_v) \right\} \quad (30)$$

where  $\tilde{L} \triangleq (u_t - u)^2$ ,  $\tilde{\lambda}$  is a co-state variable,  $\theta_v = \tan^{-1}(u_y/u_x)$ , and  $u_t$  is the tangential velocity along the ballistic trajectory which is obtained by a transformation from the actual radar measured velocity,  $v_r$ , which is in a radial sense ( $\vec{RP}$ ). The cost function is defined here in terms of the tangential velocity of the ballistic trajectory. This is because in this case such a cost function results in simpler mathematical derivations. However, the result would be the same if the cost function were re-defined in the radial sense, as in (7).

From (6),

$$u_t = v_r \left( \frac{r_x}{r} \frac{u_x}{u} + \frac{r_y}{r} \frac{u_y}{u} + \frac{r_z}{r} \frac{u_z}{u} \right) = v_r \frac{v_d}{u}, \quad (31)$$

which can be regarded as the desired output. With  $\partial \theta_v / \partial C_{df} \approx 0$ ,  $\partial V / \partial u \approx 1$ ,

$$\begin{aligned} \dot{\tilde{\lambda}} &= -\partial \tilde{H} / \partial u \\ &= -2(u - u_t) + \tilde{\lambda} \rho s V C_{df} / m, \quad \tilde{\lambda}(T) = 0, \end{aligned} \quad (32)$$

$$\dot{u} = \partial \tilde{H} / \partial \tilde{\lambda} = -\frac{\rho s}{2m} V^2 C_{df} - g \sin(\theta_v), \quad (33)$$

$$\varphi_i = \int_0^T \frac{\rho s}{2m} V^2 \tilde{\lambda} \frac{\partial C_{df}}{\partial c_i} dt, \quad (34)$$

$$\frac{\partial \varphi_i}{\partial c_j} = \int_0^T \frac{\rho s}{2m} \frac{\partial C_{df}}{\partial c_i} \left[ V^2 \frac{\partial \tilde{\lambda}}{\partial c_j} + 2\tilde{\lambda} V \frac{\partial u}{\partial c_j} \right] dt. \quad (35)$$

From (32) and (33), the d.e.'s for  $\partial \tilde{\lambda} / \partial c_j$  and  $\partial u / \partial c_j$  can be written as

$$\frac{d}{dt} \frac{\partial u}{\partial c_i} = -\frac{\rho s}{2m} \left[ V^2 \frac{\partial C_{df}}{\partial c_i} + 2C_{df} V \frac{\partial u}{\partial c_i} \right] \quad (36)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial \tilde{\lambda}}{\partial c_i} &= -2 \frac{\partial u}{\partial c_i} + \frac{\rho s}{m} \left[ \tilde{\lambda} V \frac{\partial C_{df}}{\partial c_i} + \tilde{\lambda} C_{df} \frac{\partial u}{\partial c_i} \right. \\ &\quad \left. + V C_{df} \frac{\partial \tilde{\lambda}}{\partial c_i} \right]. \end{aligned} \quad (37)$$

The standard Newton-Raphson iteration can be applied similarly by the forward integration of (1) and (36) with  $\partial u / \partial c_i|_{t=0} = 0$ , and then by the backward integration of (32) and (37) with  $\partial \tilde{\lambda} / \partial c_i|_{t=T} = 0$ . The total number of the d.e.'s to be integrated in this scheme is  $4n + 11$ , which is less than the half of the quantity ( $12n + 18$ ) for the quasi-Newton-Raphson iteration. If the idea of quasi-Newton-Raphson iteration is combined with this approximated Hamiltonian function scheme, the total number of d.e.'s will be  $2n + 6$ . This is achieved by replacing (34) and (35) with

$$\varphi_i = \int_0^T 2(u - u_t) \frac{\partial u}{\partial c_i} dt \quad (38)$$

$$\frac{\partial \varphi_i}{\partial c_j} = \int_0^T 2 \frac{\partial u}{\partial c_i} \frac{\partial u}{\partial c_j} dt \quad (39)$$

where only forward integrations of (1) and (36) are required, as discussed in Section 4.1.

The explicit assumption used in this simplification is based on the simple intuition that the velocity is mainly affected by aerodynamic drag. If a quasi-Newton-Raphson iteration is used, a similar assumption on  $C^{(0)}$  should be observed as discussed in Section 4.1.

## 5. CORRECTION OF INACCURATE $X(0)$

In most optimal tracking control problems, the initial state  $X(0)$  is assumed to be known exactly. In many real applications, this is not the case. Because the accuracy of  $X(0)$  will directly affect the final optimal solution, it is reasonable to take the inaccurate initial states as the design parameters. To take the  $X(0)$  as design parameters can be easily carried out in the proposed optimal dynamic fitting method. In this paper, the approximated initial  $X(0)$  can readily be obtained from the measured data and the testing setup.

To apply the quasi-Newton-Raphson method, the procedure for optimizing  $X(0)$  is similar to that for  $C^T$ . Referring to (20), 36 d.e.'s related to  $X(0)$  can similarly be determined. The relevant initial integration conditions must be set to 0 except the following

$$\begin{aligned} \partial x/\partial x_0|_{t=t_0} &= 1, & \partial y/\partial y_0|_{t=t_0} &= 1, \\ \partial z/\partial z_0|_{t=t_0} &= 1, & \partial u_x/\partial u_{x0}|_{t=t_0} &= 1, \\ \partial u_y/\partial u_{y0}|_{t=t_0} &= 1, & \partial u_z/\partial u_{z0}|_{t=t_0} &= 1, \end{aligned} \quad (40)$$

where  $\partial C_{df}/\partial X(0)$  is ignored. For the simplified schemes introduced in the above section, the initial state corrections can be made accordingly.

It should be pointed out that if some of the initial states are not directly related to the final solution and not sensitive to the performance index, one must then determine carefully, from the concept of fitting flexibility, which initial states are to be taken as the design parameters. This is an important issue in practical applications.

## 6. RESULTS FROM ACTUAL FLIGHT TESTING DATA

The main purpose of the flight tests is to finally *measure* the aerodynamic drag coefficient curve of the artillery projectiles. Several tests were carried out under different firing conditions. The identified curves are similar if the angle of attack along the trajectory is small. One of the identified curves is presented here. A set of complete flight testing data is given in the APPENDIX.

To simplify the choice of  $C^{(0)}$ , a constant  $C_{df}^{(0)}(M)$  is assumed which means only one parameter to be determined. For a spin-stabilized projectile, choosing  $C_{df}^{(0)}(M)$  as 0.3 is reasonable. Thus, the initial parameters are set as  $c_{2i-1}^{(0)} = 0.30, c_{2i}^{(0)}$ , ( $i = 1, 2, 3$ ). It should be pointed out that, from the Mach range of this set of flight testing data, to

guarantee the "fitting flexibility", at least 2 segments (3 knots) should be used. In this paper, the number of segments  $n$  is 2 with 3 knots. However, a larger  $n$  can be applied but will not always be desirable because this will increase the computation cost and cause the "over-fitting" problem.

It has been verified that the four schemes, i.e., the standard Newton-Raphson iteration (22), the quasi-Newton-Raphson version, the standard Newton-Raphson iteration utilizing the approximated Hamiltonian and its quasi-Newton-Raphson form, all have the same converged  $C_{df}(M)$  which is plotted as a solid line in Fig. 3. The computational cost comparisons of the four schemes support the arguments regarding the computational simplifications in Section 4. In Fig. 3, the curve shown as a dash-dotted line is the result of the work described in (Chen and Dou, 1993b).

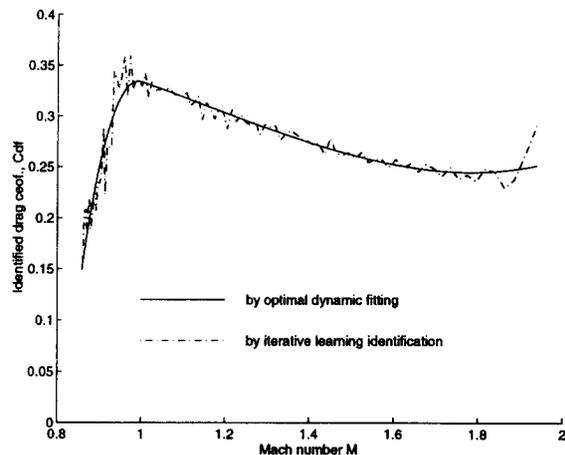


Fig. 3. Comparison of the results of optimal dynamic fitting and iterative learning identification (Chen and Dou, 1993b)

It can be clearly observed that the result of this paper is a fitting of (Chen and Dou, 1993b). The correctness of the identified result can be observed from Fig. 4, where the nominal zero-lift drag coefficient  $C_{d0}(M)$  and the angle of attack induced drag coefficient  $C_{d\alpha^2}(M)$  are given. Clearly, the identified  $C_{df}(M)$  comprehensively reflects the effects of the air on the motion of the center of the projectile's mass, and the motion around the center of mass. Approximately,

$$C_{df}(M) \approx C_{d0}(M) + C_{d\alpha^2}(M)\alpha^2,$$

where  $\alpha$  is the angle of attack. It can be estimated from Fig. 3 and Fig. 4 that the angle of attack at the maximum point in  $C_{df}(M)$  is around  $9^\circ$ . This observation also verified the O2 in Section 2, which implies that, for the problem of identifying the drag coefficient curve from radar measured velocity data, a point-mass 3-DOF model (1) is

sufficient.

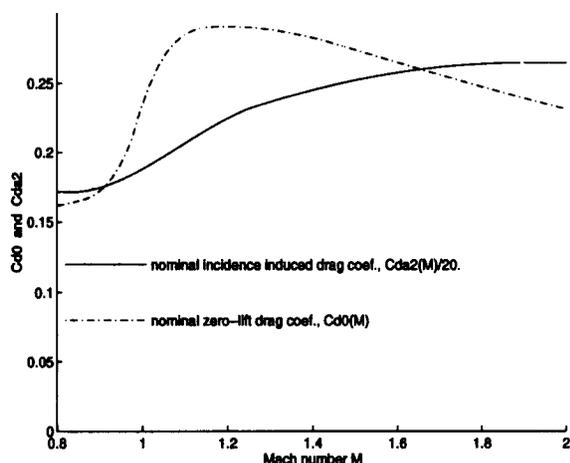


Fig. 4. Nominal drag coefficients: zero-lift drag coefficient  $C_{d0}(M)$  and incidence induced drag coefficient  $C_{d\alpha^2}(M)$

## 7. CONCLUSION

The proposed optimal dynamic fitting method has been successfully applied to an optimal tracking control problem, namely aerodynamic property curve identification from flight testing data. Cubic splines with deficiency 2 are employed as a parametric description structure for the control profile (Mach history), which guarantees the smoothness of the control profile. The problem is then converted to a multivariable parametric minimization one, and is solved by the standard Newton-Raphson iteration. To reduce the cost of the computation, a quasi-Newton-Raphson iteration is proposed, and the cost is effectively reduced by half, using this method. Further simplifications of the problem are introduced by using an approximated Hamiltonian of the dominant variable, which cuts the computing cost by half once more. Also, inaccurate initial states can easily be corrected. The optimal fitting drag coefficient curve obtained is apparently better than the commonly used fitting ballistic factor. Thus, a more accurate firing table could then be produced. Furthermore, the method of this paper supplies an effective way of solving a wider class of singular optimal control problems. Results of practical flight testing data reduction indicate the effectiveness of the methods proposed in this paper.

## ACKNOWLEDGEMENTS

The authors are grateful to the reviewers' encouragement, patience and technical insights.

## REFERENCES

- Anderson, L. C. and J. H. Vincent (1985). Application of system identification to aircraft flight test data. In 'Proceedings of the 24th IEEE Conference on Decision and Control'. Fort Lauderdale, FL, USA. pp. 1929–1931.
- Bartelson, N. (1975). A method for determination of aerodynamic drag by doppler data. In 'Proc. of the First Int. Ballistics Symp.'. Orlando, USA.
- Chapman, G. T. and D. B. Kirk (1970). 'A method for extracting aerodynamic coefficients from free flight data'. *AIAA Journal*.
- Chen, Y. (1990a). 'Researches on trajectory prediction models and software for spin-stabilized projectiles (I): Automatic model-switching trajectory prediction method'. *Projectile and Rocket Fascicule of Acta Armamentarii* 1990(3), 1–7 (in Chinese).
- Chen, Y. (1990b). 'Researches on trajectory prediction models and software for spin-stabilized projectiles (II): Studies on trajectory prediction model reduction'. *Projectile and Rocket Fascicule of Acta Armamentarii* 1990(4), 1–11 (in Chinese).
- Chen, Y. (1991). 'Researches on trajectory prediction models and software for spin-stabilized projectiles (III): A new rigid body 6-dof model for trajectory prediction of unguided spin missiles'. *Projectile and Rocket Fascicule of Acta Armamentarii* 1991(1), 1–10 (in Chinese).
- Chen, Y. and H. Dou (1991). 'Data fitting algorithm and general purpose software by constrained multi-staged polynomials with different orders'. *Aerodynamic Experiment and Measurement Control* 5(3), 78–86 (in Chinese).
- Chen, Y. and H. Dou (1993a). 'Researches on the optimal control solution of identifying fitting drag coefficient curve from radar measured velocity data'. *Aerodynamic Experiment and Measurement Control* 1993(2), 81–89 (in Chinese).
- Chen, Y. and H. Dou (1993b). Robust curve identification by iterative learning. In 'Proc. of the First Chinese World Congress on Intelligent Control and Intelligent Automation (CWC ICIA'93)'. Beijing, China. pp. 1973–1980.
- Chen, Y., D. Lu, H. Dou and Y. Qing (1992). Optimal dynamic fitting and identification of aerobomb's fitting drag coefficient curve. In 'Proc. of the First IEEE Conf. on Control Applications'. Dayton, Ohio, USA. pp. 853–858.
- Eulrich, B. J. and E. G. Rynaski (1975). 'Identification of nonlinear aerodynamic stability and control parameters at high angle of attack'. *AGARD Methods for Aircraft State and Parameter Identification* pp. 1–15.
- Fratter, C. and R. F. Stengel (1983). 'Identification of aerodynamic coefficients using flight testing data'. *AIAA Paper No. 83-2099*.

- Gupta, N. K. and K. W. Illif (1982). 'Identification of aerodynamic indicial functions using flight data'. *AIAA Paper 82-1375*.
- Illif, K. W. (1985). 'Extraction of aerodynamic parameters for aircraft at extreme flight conditions'. *NASA-TM-86730 H-1290 NAS 1.15:86730 AGARD-PAPER-24* pp. 1-24.
- Illif, K. W. (1989). 'Parameter estimation for flight vehicle'. *AIAA Journal of Guidance and Control* **12**(5), 609-622.
- Illif, K. W. and R. E. Maine (1986). 'A bibliography for aircraft parameter estimation'. *NASA TM-86804*.
- Jiang, Q. and Q. Chen (1986). Dynamic model for real-time estimation of aerodynamic characteristics. In 'Collection of Technical Papers - AIAA Atmospheric Flight Mechanics Conference'. Williamsburg, VA, USA. pp. 331-339.
- Larimore, W. E., W. M. Lebow and R. K. Mehra (1985). Identification of parameters and model structure for missile aerodynamics. In 'Proceedings of the American Control Conference'. Boston, MA, USA. pp. 18-26.
- Lieske, R. F. and A. M. Mackenzie (1972). Determination of aerodynamic drag from radar data. In 'Aberdeen Proving Ground Technical Report'. USA.
- Linse, D. J. and R. F. Stengel (1994). 'Identification of aerodynamic coefficients using computational neural networks'. *Journal of Guidance, Control, and Dynamics* **16**(6), 1018-1025.
- Messner, W. and R. Horowitz (1993). 'Identification of a nonlinear function in a dynamical system'. *Trans. of ASME: J. of Dynamic Systems, Measurement, and Control* **115**, 587-591.
- Peter, H. (1989). 'System identification collaboration - the role of AGARD'. *Vertica* **13**(3), 207-212.
- Ross, A. J. (1975). 'Determination of aerodynamic derivatives from transient responses in manoeuvring flight'. *AGARD Methods for Aircraft State and Parameter Identification* pp. 1-10.
- Tobak, M. and L. B. Schiff (1981). 'Aerodynamic mathematical modeling basic concepts'. *AGARD LS-114 paper 1*.

- Williamson, W. E. J. (1980). 'Instrument modeling for aerodynamic coefficient identification from flight test data'. *AIAA Journal of Guidance and Control* **3**(3), 225-279.

#### APPENDIX: FLIGHT TESTING DATA

The complete flight testing data used in this paper are given as follows:

- 1). Projectile's Physical Parameters:  
 $d=0.155$  m.,  $m=44.99$  kg;
- 2). Atmosphere:  
ICAO Standard,  $w_x = w_z = 0.0$  m/sec.;
- 3). Initial State:  
 $(X_0) = [480.03, 447.38, 0, 274.6, 410.7, 0]^T$ ;
- 4). Radar Position:  
 $[x_r, y_r, z_r] = [-67.4, 150.0, 0]$  m.;
- 5). Radar Measured Data:

$v_r$  series:

(Row by row in equal time intervals of  
 $h = 0.3$  sec. over time period [0.6, 30.6] sec. )

652.9757	644.2016	635.4464	627.0872	618.8109
610.3392	602.2492	594.5349	586.8960	579.4882
572.0804	565.0996	557.9661	551.2083	544.5161
538.0422	531.5992	525.3740	519.2748	513.2719
507.3897	501.6705	495.9301	490.4462	484.9349
479.6691	474.3199	469.1093	464.0607	459.0376
454.1041	449.2432	444.4515	439.6902	435.0269
430.4870	426.0405	421.5356	417.1813	412.8641
408.7190	404.4716	400.4320	396.3070	392.3452
388.3574	384.5102	380.7192	376.9328	373.1922
369.5753	365.9011	362.3417	358.8414	355.2994
351.8651	348.4608	345.0592	341.7309	338.4507
335.2063	331.9882	328.8130	325.6838	322.6044
319.5674	316.5649	313.5905	310.6943	307.7371
304.9160	302.0498	299.2961	296.4955	293.7404
291.0726	288.3348	285.7450	283.1665	280.6173
278.1360	275.9500	273.6631	271.5456	269.4207
267.2413	265.2183	263.2231	261.2731	259.3871
257.4787	255.6622	253.8853	252.1319	250.3103
248.6151	246.9184	245.2147	243.5340	241.9693
240.3281				