Using Gradient and Least Squares Methods to Estimate Unknown Parameters of an Iced Aircraft

Paul Pawola
ECE 417
Fall, 2000

Abstract

This project involved using the Gradient and Least Squares Methods to identify the parameters in the Input/Output transfer function of an iced aircraft. By identifying the values of the parameters, the level of icing can be found. Initially, the state space form of lateral dynamics was converted into a transfer function, and from the transfer function was reparameterized into an identification form. This form was then simulated in Matlab for both parameter identification methods, initially without noise and secondly in the presence of noise with the transfer function parameters α_0 and β_3 as the unknowns. The results of these simulations yielded that using the gradient method to identify α_0 gave the best indication of icing.

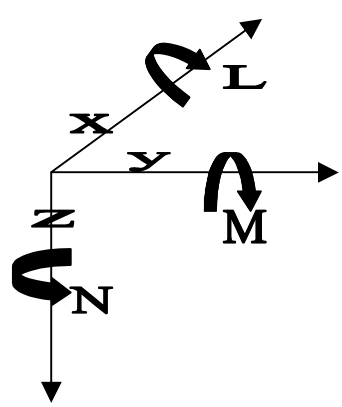
Introduction

Work currently being done by the *Smart Icing Systems* (SIS) group at the University Of Illinois Urbana-Champaign involves using control theory to detect the change to aircraft flight dynamics, more specifically changes to the stability and control derivatives, during the accretion of ice on the wings and the tail. The goals of this project were to extend the results of this research and to apply the parameter estimation techniques presented in ECE 417 to view how these estimation techniques' results compare to previous results and to gain better insight into the actual implementation and results of these techniques when applied to a real system.

When ice accretes on an aircraft, the aircraft flight dynamics are changed due to changes in the stability and control derivatives [1]. The stability and control derivatives can be thought of as the coefficients to the states in the linear state equations when the aircraft's nonlinear dynamics are linearized around an operating point. When these change the plane can become unstable when the pilot applies an maneuver, and the result may be disastrous. The severity of icing is characterized by η, which is defined as 0 for no icing and 1 for severe icing. When ice forms on the aircraft, the stability and control derivatives are changed, and thus each stability and control derivative is a function of η, as well as other factors [2]. The values for the various stability and control derivatives are well known when the aircraft is completely free of ice and also when the aircraft is fully iced [3]. The purpose, then, of the parameter estimation in this scenario is to identify the values of the stability and control derivatives, and to use these values to evaluate the severity of the icing.

The aircraft model used for this research takes into account the six possible degrees of movement available to the aircraft: translational displacement in the x,y,z directions, and rotational displacement around the x,y,z directions.

Figure 1: Aerodynamics Aircraft Model Reference directions [1]



Here {X,Y,Z} represent the x,y,z directions, and {L,M,N} represent the corresponding rotations. When characterizing the aircraft, it is necessary to describe not only the orientation in each of these six directions, but also velocities as well. Table 1 gives the symbols used to describe the aircraft angle and angular velocity in each of the rotational directions.

Table 1: Symbol Used To Describe Aircraft Angles and Angular Velocities [3]

	L Rotational Direction "Roll"	**M Rotational Direction "Pitch"**	**N Rotational Direction "Yaw"**
Angle	ϕ	θ	ψ
Angular Velocity	p	q	r

To simplify the calculations, the dynamics of the aircraft are separated into two categories, lateral motion and longitudinal motion. The lateral motion includes displacements and velocities in the Y, L, N directions, while the longitudinal motion includes the corresponding values in the X, Z, M directions. The parameter identification for the longitudinal motion is covered by Melody et. al in [1]. For the lateral motion, roll rate, yaw rate, roll angle and yaw angle are considered. Additionally, the sideways velocity, v, is added to state. The resulting the state vector used in this project was $\mathbf{x} = [\mathbf{p}, \mathbf{r}, \mathbf{v}, \boldsymbol{\phi}, \boldsymbol{\psi}]^T$ [3].

The inputs to the system correspond to the pilot of the aircraft changing the roll, pitch, and yaw angles, and are designated as in Table 2. For the lateral motion, the input is $\mathbf{u} = [\delta_r, \delta_a]$. Generally, the inputs used have been doublets, or sinusoids.

Table 2: Names and Symbols Used for Aircraft Input [3]

Input Name	**Symbol Used**	**Angle Changed**
Elevator	δ_e	Pitch
Rudder	δ_r	Yaw
Aileron	δ_a	Roll

When the system is put into state space form, since there are five elements of the state vector and two inputs, $\mathbf{A} \in R^{5x5}$ and $\mathbf{B} \in R^{5x2}$. Since each of the elements of the state are physical values and assumed measurable, the **C** matrix for the output of the system is arbitrary, depending on what output is desired. Each element of the **A** and **B** matrices is a linear combination of the stability and control derivatives. In this project, the parameter estimation methods estimate the parameters of the Input/Output model transfer function. When the state space representation is converted into an transfer function, each parameter of the transfer function corresponds to a complex equation of stability and control derivatives. Therefore, for simplicity, all parameters discussed will be terms of the transfer function parameters.

System Description

Setting the state vector **x = [p, r, v, ϕ, ψ]^T** and the input vector **u = [δa, δr]**, the state space representation of the lateral system, with all stability and control derivatives known and η=0.4, is

$$\dot{x} = \begin{vmatrix} -6.3115 & 0.7238 & -8.0287 & 0 & 0 \\ 09.5593 & -1.0454 & 4.3137 & 0 & 0 \\ -0.008 & -0.984 & -0.1792 & 0.119 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{vmatrix} x + \begin{vmatrix} -15.6207 & 1.1942 \\ -0.5266 & -5.7881 \\ 0 & 0.0478 \\ 0 & 0 \\ 0 & 0 \end{vmatrix} u$$

For this project, in order to identify the parameters using the given methods, it was necessary to convert the system from state-space form into a transfer function, and from the transfer function to reparameterize the system into an identification form in terms of the known and unknown parameters. Since each of the quantities of the state vector **x** are assumed measurable, any **C** matrix can be used. For this project, the matrix **C = [0 0 0 1 0]** was used to set the output of the system equal to ϕ. For simplicity, the input δ_a was set to zero, and the resulting transfer function, simplified to one decimal place, was found to be:

$$Y(s) = \frac{1.2s^2 - 3.1s - 41.4}{s^4 + 7.5s^3 + 12.5s^2 + 33.4s + 0.6} U(s)$$

or

$$Y(s) = \frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{s^4 + \beta_3 s^3 + \beta_2 s^2 + \beta_1 s + \beta_0} U(s)$$

The transfer function was then expanded and all signals were filtered through a stable filter Δ(s). The resulting reparameterization is

$$\left(s^4\Delta\right)Y = \left[\alpha_2, \alpha_1, \alpha_0, \beta_3, \beta_2, \beta_1, \beta_0\right]\begin{bmatrix} (s^2\Delta)U \\ (s\Delta)U \\ (\Delta)U \\ -(s^3\Delta)Y \\ -(s^2\Delta)Y \\ -(s\Delta)Y \\ -(\Delta)Y \end{bmatrix}$$

$$or \quad Z = \theta^{*T}\phi$$

$$with$$

$$\Delta = \frac{10000}{s^5 + 20s^4 + 155s^3 + 580s^2 + 1044s + 720}$$

Implementation

Gradient Method

For the parameter estimation, the transfer function parameters α_0 and β_3 were chosen. Since there are only two unknown parameters, the input need only to have one frequency to be persistently exciting. The input chosen was u(t) = 2sin(t). A sample of the input is shown in Figure (2).

The first estimation method simulated was the gradient method. In this method, the adaptive law is $\dot{\theta} = \Gamma e\phi$ [4], where the normalized estimation error $e = \frac{z - \hat{z}}{m^2}, m^2 = 1 + \left(\sqrt{\phi^T\phi}\right)^2$ [4]. In this case, the adaptive gain Γ is set to a constant value of 6, so that each element of $\dot{\theta}$ is scaled equally. To allow for ease in changing the location and number of unknowns, the adaptive law was implemented such that the elements of $\dot{\theta}$ that correspond to known parameters were set to zero while keeping the dimension the system set to seven.

Initially, the known system transfer function was simulated in Matlab using the input u(t) to give the signal y(t). Next, the signals u(t) and y(t) were simulated through the filter $\Delta(s)$ to give the filtered versions of the input and output that make up $\phi(t)$ and z(t). It is key to note that each of these initial simulations are done at discrete time instances within a time range, in this case the initial time being 0 sec and the final time being 250 sec.

To simulate the parameter estimation, the Matlab command *ode23* was used. This command allows the user to create a Matlab file that contains a system of differential equations, and when given the file name, the initial conditions, and the time range, Matlab will numerically solve the specified system for times in the time range. However, due to the way Matlab solves the equations, it finds the numerical solution at nonuniform time steps. Because of this, at each time step during the simulation of the parameter estimation, the signal z(t) and the elements of $\phi(t)$ needed to be interpolated. This resulted in slowing the simulation significantly so that it ran much slower than real time, where the ratio of simulation time to real time exceeded 1:5.

When running the simulation with noise, the noise was modeled as zero-mean and uniformly distributed on [-1, 1]. Noise was added to each signal. Additionally, the simulation running time was increased from 250 sec to 300 sec. A sample of the noise is shown in Figure (3).

Least Squares Method

The set up of the implementation of the least squares method was the same as that for the gradient method, with one minor exception: in this case, the elements of θ^* that were known were subtract out from z:

$$Intitially: z = \theta^{*T}\phi$$

$$Here: Z_{LS} = z - \theta^{*T}_{known}\phi_{known} = \theta^{*T}_{unknown}\phi_{unknown}$$

This change has no bearing on the outcome of the simulation. This was done to reduced the order of the system from seven down to two, and thus to reduce the complexity of the system of differential equations.

The adaptive law for the least squares method is $\begin{array}{l}\dot{\theta} = pe\phi \\ \dot{p} = -p\dfrac{\phi^T\phi}{m^2}p\end{array}$ [4], where m and e are the same as in the gradient method. To simulate the adaptive law, the same Matlab command *ode23* was used with the equations for the least squares method. Like in the gradient method, zero-mean noise uniformly distributed on [-1, 1] was added. For all least squares simulations, the simulation running time was set to 300 sec.

Results

The simulations of the gradient method, without noise, showed that the unknown parameters converged to their correct values. As shown in Figure (4), the value of the parameter α_0 very quickly approached the known correct value of –41.4. At this time, although the input to the system did not change, the rate of change of α_0 drastically slowed, and then α_0 approached the known value asymptotically.

For estimation of β_3, like the estimation of α_0, the value of the parameter estimate very quickly approached the known correct value of 7.5, then drastically slowed to approach the value asymptotically. However, unlike α_0, the estimate of β_3 had overshoot, where the value actually exceeded the correct value then slowly decayed down to the correct value. Additionally, the early values of the estimate dipped very negative, but this was quickly corrected. This is shown in Figure (5).

In the presence of noise, both parameters again approached the known correct value, but there was error in each of the final estimates. Figure (6) shows the results of the estimation of α_0 in the presence of noise, the combined result of ten simulations, each each with different noise realizations. Figure (7) shows the results for β_3. Clearly from the figure it is seen that there is a spread of final values of the estimates. Table 3 shows this spread statistically:

Table 3: Statistics of Gradient Method Parameter Estimation With Noise (10 Simulations)

Parameter	Minimum	Maximum	Mean	Standard Deviation
α_0	-41.5	-40.8	-41.1	0.209
β_3	6.9	8.2	7.7	0.444

As shown in table 3, on average, neither parameter converged to the correct value, although the error was rather small for the error between the Mean final value and the correct final value. For α_0, the error from the Mean final value to the correct final value is less than 1%, and the maximum error is 1.45%. The results for β_3 show the error from the Mean final value to the correct final is 2.67%, and the maximum error is 9.33%.

The simulations of the least squares method, as in the gradient method, had both parameters converged to the correct value without the presence of noise. For α_0, the rate of change closely followed the shape of a decaying exponential, without the abrupt changes witnessed in the gradient method. The estimation of β_3 did not dip as far negative as it did in the gradient method, but the overshoot in estimation was much more pronounced. However, despite these differences, the general shape of the response of the estimates for both α_0 and β_3 were similar to that in the gradient method. Figures (8) and (9) show these results.

In the presence of noise, the least squares method results did not perform was well as the gradient method in terms of average error, but performed much better in terms of maximum error and final value standard deviation. For both parameters, the final value after 300 seconds was, on average, further away from the correct value than in the gradient method. For β_3, the overshoot in the presence of noise was further increased. Figures (10) and (11) show the combined results for twenty simulations. Table 4 gives the statistics of the results:

Table 4: Statistics of Least Squares Method Parameter Estimation With Noise (20 Simulations)

Parameter	Minimum	Maximum	Mean	Standard Deviation
α_0	-40.8	-40.6	-40.7	0.07
β_3	7.69	7.85	7.75	0.05

As Table 4 presents, on average, both parameters did not converge to, or even cross, the correct value. For α_0, as in the gradient case, the error was small: the error from the Mean final value to the correct final value is 1.67%, and the maximum error is 1.92%. For β_3 the error from the mean final value to the correct final is 3.33%, and the maximum error is 4.67%.

If, during a maneuver, the level of icing is determined by the parameter identification to be severe, the pilot should be alerted so that measures can be taken to ensure that the aircraft does not become unstable. Therefore, it paramount that these parameters converge quickly so that the pilot can be alerted in a timely manner. Define, for this discussion, the term *strong setting time* to be the time when the parameter estimate crosses, for the last time, to within ±5% of the correct value, and the term *weak setting time* to be the time when the parameter estimate crosses, for the last time, to within ±10% of the correct value. Table (5) shows the strong and weak settling times for the parameters with no noise added.

Table 5: Strong and Weak Settling Time For α_0 and β_3 With No Noise.

	Gradient Method		Least Squares Method	
	Strong Settling Time (sec)	**Weak Settling Time (sec)**	**Strong Settling Time (sec)**	**Weak Setting Time (sec)**
α_0	5.37	3.36	35.75	14.64
β_3	28.67	9.44	90.03	33.37

When noise is added to the system, the results get much worse as the settling times increase and become, in some instances, quite variable, as shown in Tables (6a) and (6b).

Table 6a: Statistics of Strong and Weak Settling Times For α_0 and β_3 in the Presence of Noise for Gradient Method (10 simulations)

	α_0		β_3	
	Strong Settling Time (sec)	**Weak Setting Time (sec)**	**Strong Setting Time (sec)**	**Weak Setting Time (sec)**
Minimum	3.78	3.21	N/A	247.80
Maximum	5.97	3.35	N/A	299.80
Mean	5.05	3.29	N/A	285.05
Standard Deviation	0.93	0.05	N/A	18.22

Table 6b: Statistics of Strong and Weak Settling Times For α_0 and β_3 in the Presence of Noise for Least Squares Method (20 simulations)

	α_0		β_3	
	Strong Settling Time (sec)	**Weak Setting Time (sec)**	**Strong Setting Time (sec)**	**Weak Setting Time (sec)**
Minimum	30.16	13.61	48.66	20.22
Maximum	52.40	18.30	230.52	83.10
Mean	41.24	16.14	124.28	40.08
Standard Deviation	5.50	1.38	44.88	13.58

The N/A for the strong settling time for β_3 in the gradient method stands for Non Available. This is due to the fact that, for six of the ten simulations, the parameter failed to cross to within ±5% of the correct value during the simulation time.

Conclusion

After simulating the parameter estimation methods and analyzing the results, the issue that needs to be addressed is how useful are these techniques in warning the pilot of the aircraft that ice has accumulated. To determine this, both the parameter estimation method and the parameter itself must be analyzed.

Tables (3) and (4) show that, in the presence of noise, the gradient method was more successful than the least squares method in achieving an accurate parameter identification. The fact that the standard deviation was higher for the gradient simulations is not important since the average error for the gradient method is low. In both methods, the results for α_0 had less error than the results for β_3. These results suggest that using the gradient method to identify the parameter α_0 will give the most accurate indication of icing.

Another important issue is the speed of the parameter identification. Tables (5), (6a), and (6b) show the weak and strong settling times of both parameters in both identification methods. From these tables, using the gradient method to identify β_3 is clearly unacceptable, since the weak settling times are, on the average, over four minutes, and the parameter never met the criteria for the strong settling time during the simulations. Over both parameters and both identification methods, identifying α_0 using the gradient method is, by far, the quickest of the combinations and therefore can give the quickest indication of icing.

From the results presented in this project, it was found that using the gradient method to identify α_0 gives both the most accurate and the quickest indication of icing: within 2% accuracy with a strong settling time of about 5 seconds and a weak settling time of about 3.3 seconds. These are the best results obtained in this project; these results are comparable to those obtained by using the methods used by the SIS group, which are achieved by using the H^∞ algorithm to specifically identify the stability and control derivatives. The results using the H^∞ method identify the actual values for certain stability and control derivatives on the order of 2 to 3 seconds. The reader is referred to [3] for these results.

References

[1] Melody, J.W., Basar, T., Perkins, W. R., Voulgaris, P.G. (2000). Parameter Identification for Inflight Detection and Characterization of Aircraft Icing. Smart Icing Systems Project, Decision and Controls Group, Coordinated Sciences Lab at the University of Illinois Urbana-Champaign. <http://black1.csl.uiuc.edu/~jmelody/icing/publications.html>

[2] Hillbrand, Thomas. (1999). Identification of Time-Varying Flight Dynamics Parameters for Aircraft Icing Detection and Classification Using H^{∞} Methods. University of Illinois at Urbana-Champaign, UILU-ENG-99-2223, DC-190. Smart Icing Systems Project, Decision and Controls Group, Coordinated Sciences Lab at the University of Illinois Urbana-Champaign. <http://black1.csl.uiuc.edu/~jmelody/icing/publications.html>

[3]. Melody, J. (2000) 2000 Nasa Review. Smart Icing Systems Project, Decision and Controls Group, Coordinated Sciences Lab at the University of Illinois at Urbana-Champaign. <http://black1.csl.uiuc.edu/~jmelody/icing/publications.html>

[4]. Liberzon, Daniel. (2000) ECE 417 Class Notes, Fall 2000.

List of Figures