Sample Air Data Calibration Problem

by Juan 'Silv' Jurado

August 19, 2020

1 Introduction

Pitot-static systems are useful for inferring an aircraft's state since they measure static and dynamic pressures around the airframe, which are intrinsically linked to its performance and flying qualities. At the same time, the mechanisms that allow an aircraft to fly and produce lift are the same mechanisms that drive errors in the Pitot-static system's ability to produce accurate measurements. Specifically, the act of flying interferes with the static port's ability to measure true ambient pressure, P_a, and we inevitably end up measuring a corrupted version of it, which we call static pressure, P_s.

Our job as flight testers in the context of Pitot-statics is to calibrate the aircraft's Air Data System (ADS). Namely, we want to produce a correction curve for P_s so that the Air Data Computer (ADC) can accurately correct it back to P_a, and produce accurate measurements of airspeed, altitude, and Mach number to the aircrew. Note that this calibration is indeed a *curve*, because the difference between P_a and P_s is a function of Mach number, angle of attack, and load factor, to name a few. Essentially, it is a function of how much lift the aircraft is producing in order to achieve a particular flight condition. This so-called "static source position error curve" is defined as

$$\frac{\Delta P_p}{P_s} = \frac{P_s - P_a}{P_s}. \tag{1}$$

where the quantity ΔP_p is itself a function of the previously mentioned conditions (Mach number, angle of attack, etc.). Note that P_s also shows up in the denominator. It turns out this static position error can be modeled as a scale factor or percentage. This is also useful for an ADC needing to correct P_s into P_a because given a measured P_s and the calibration curve from (1) we can correct as follows

$$\Delta P_p = \frac{\Delta P_p}{P_s} P_s \tag{2}$$

$$P_a = P_s - \Delta P_p \tag{3}$$

So how do we come up with a $\Delta P_p / P_s$ curve? There are various ways of doing so. As shown in Figure 1, both altimeter and airspeed indicator errors emanate from ΔP_p, which is a pressure error. Therefore, we can use altitude methods (such as a the Tower Fly-by) to characterize altimeter error, or airspeed methods (such as a cloverleaf, or level turn, etc.) to characterize airspeed error, either way, they both lead to pressure error, which is exactly ΔP_p.

In this quick review, we will focus on solving a sample Tower Fly-by (TFB) problem. As just discussed, the TFB is an *altitude method* for characterizing ΔP_p. Once we have ΔP_p from any method (altitude, airspeed, survey), we can use it to produce corrections to any instrument (altimeter, airspeed, Mach number). We will use the sample data from Erbman's Air Data Course homework Problem 13.

2 Problem Setup

These are the setup parameters from the problem itself. We are given the following in-tower data:

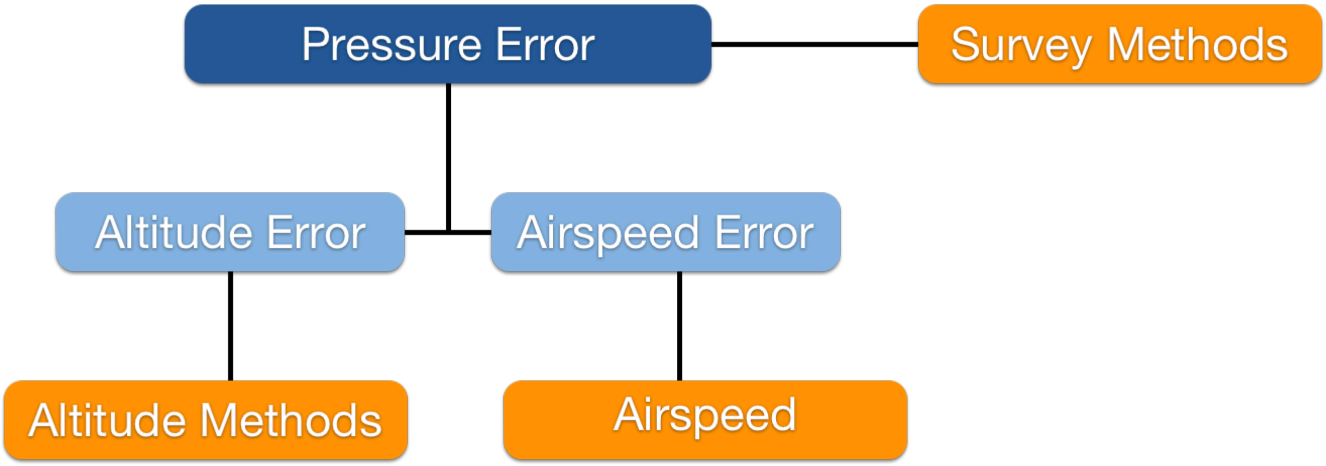

Figure 1: ADS Calibration Methods

Tower pressure altitude reading:	2010 ft
Tower ambient temperature (T_a) reading:	29.4° C
Tower grid reading:	2.5
Tower grid constant:	31.4 ft/dif

Additionally, we have the following information from the test aircraft:

Altimeter instrument correction, ΔH_{ic}	-22 ft
Airspeed indicator instrument correction, ΔV_{ic}	3.0 kts
Indicated altitude, H_i	2017 ft
Indicated airspeed, V_i	302 kts

Recall, this is an altitude-based method so we will first compute an altimeter error, then translate it to a pressure error. Additionally, this specific test point was completed at an airspeed, altitude, and weight (which affects angle of attack) of interest, which means we are only characterizing a single *point* on the ΔP_P curve for this aircraft. This is why you would normally do *many* TFB test points at varying Mach numbers and weight bands to characterize the entire curve (or family of curves).

3 Problem Solution

We are asked to compute the following parameters: H_{ic}, V_{ic}, H_c, ΔH_{pc}, and $\Delta P_p/P_s$. Let's begin by computing H_{ic} and V_{ic} which are the instrument-corrected altitude and airspeed, respectively. Recall instrument corrections are simply due to internal mechanical, rounding or other errors the instrument itself creates when displaying the information to the aircrew. Since we were directly given the corrections in the setup, these two are straightforward:

$$H_{ic} = H_i + \Delta H_{ic}$$
$$H_{ic} = 2017 - 22$$

$$\boxed{H_{ic} = 1995 \text{ [ft]}} \tag{4}$$

$$V_{ic} = V_i + \Delta V_{ic}$$
$$V_{ic} = 302 + 3.0$$

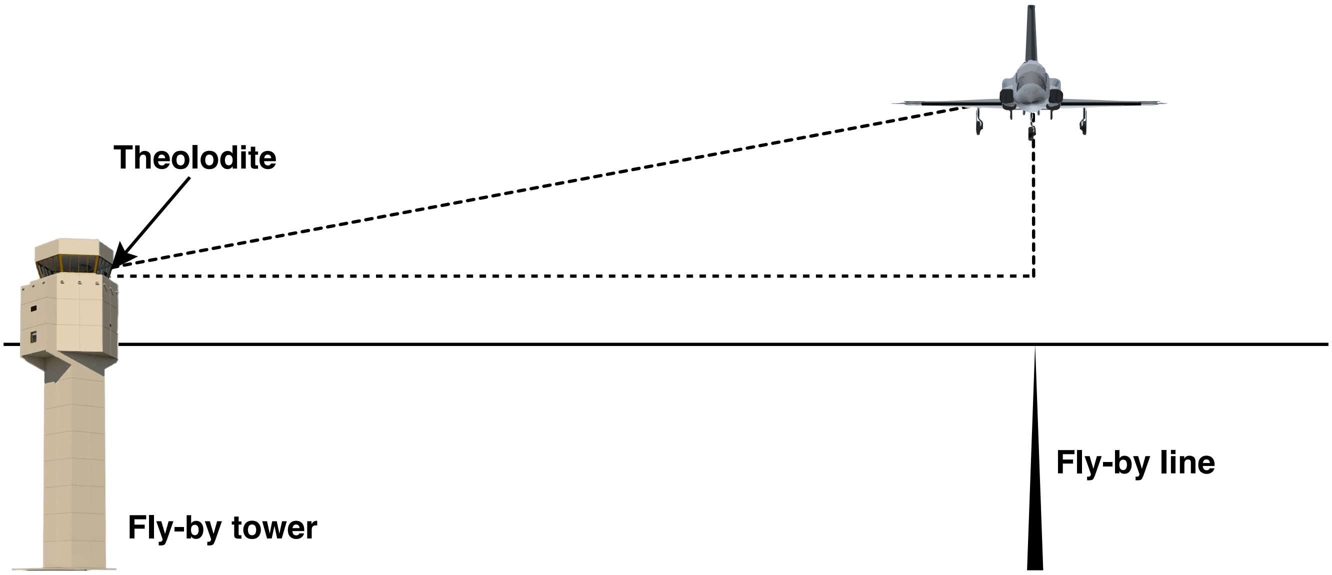

Figure 2: TFB Diagram

$$\boxed{V_{ic} = 305 \text{ [kts]}} \tag{5}$$

Next, we are asked to find H_c, or the position-corrected altitude. Recall

$$H_c = H_{ic} + \Delta H_{pc} \tag{6}$$

which means we need to first find ΔH_{pc} since we already have H_{ic}. This is where the TFB technique comes in handy. As illustrated in Figure 2, we are going to leverage the fact that we have an altimeter and a thermometer in the tower, at zero airspeed, reading truth data very close to the test aircraft. The bulk of the computations for this technique are related to extrapolating our in-tower truth P_a up to the actual test altitude flown by the aircraft. Let's begin by writing out some of the basic equations we will need.

$$\delta(H) = (1 - 6.875\,59 \times 10^{-6} H)^{5.2559} \tag{7}$$
$$\theta(H) = 1 - 6.875\,59 \times 10^{-6} H \tag{8}$$
$$M(Q_c, Pa) = \sqrt{5((Q_c/P_a + 1)^{2/7} - 1)} \tag{9}$$
$$T_{SL} = 15 + 273.15 = 288.15 \text{ [K]} \tag{10}$$
$$P_{SL} = 14.696 \text{ [psi]} \tag{11}$$
$$a_{SL} = 661.48 \text{ [kts]} \tag{12}$$

Next, we will use the geometry of the tower theodolite to find out how much higher than the tower the aircraft flew. Then we will convert that geometric difference into a pressure altitude difference using (8) and (10)

$$T_{test} = T_a + 273.15 = 29.4 + 273.15 = 302.55 \text{ [K]}$$
$$T_{std} = T_{SL}\theta(H_{tower}) = T_{SL}\theta(2010) = 288.15(0.9862) = 284.17 \text{ [K]}$$
$$\Delta H = 2.5(31.4) = 78.5 \text{ [ft]}$$
$$\Delta H_c = \frac{T_{std}}{T_{test}} \Delta H = \frac{284.17}{302.55} 78.5 = 73.73 \text{ [ft]}$$

Now we can add this pressure altitude difference to the tower's pressure altitude reading to obtain the true pressure altitude at the aircraft's fly-by altitude.

$$H_c = H_{tower} + \Delta H_c = 2010 + 73.73$$

$$\boxed{H_c = 2083.73 \text{ [ft]}} \tag{13}$$

At this point, we have the necessary information to use (6) and compute the altitude position error correction

$$\Delta H_{pc} = H_c - H_{ic}$$
$$\Delta H_{pc} = 2083.73 - 1995$$

$$\boxed{\Delta H_{pc} = 88.73 \text{ [ft]}} \tag{14}$$

Now that we have an altitude error, ΔH_{pc}, we can infer the underlying pressure error, ΔP_p, which will then be used to compute airspeed and Mach number errors caused by the same underlying phenomenon. Let's start by using (7) with H_c and H_{ic} to compute P_a and P_s, respectively:

$$\begin{aligned} P_a &= P_{SL}\delta(H_c) = 14.696(0.9270) = 13.6226 \text{ [psi]} \\ P_s &= P_{SL}\delta(H_{ic}) = 14.696(0.9300) = 13.6669 \text{ [psi]} \end{aligned} \tag{15}$$

We then combine these using (1) and compute:

$$\boxed{\Delta P_p/P_s = 0.0032} \tag{16}$$

Finally, we have computed a pressure error in the form $\Delta P_p/P_s$. Now we can use it to infer airspeed and Mach number errors. Remember that this error is a function of Mach number, angle of attack, load factor, etc. Therefore, this single $\Delta P_p/P_s$ we have computed is only a point on a curve such as the example shown in Figure 3, and only valid for a particular weight band and load factor. Let's start with finding the airspeed position error using V_{ic} and the result from (16):

$$q_{c_{ic}} = P_{SL}\left(\left(1 + 0.2\left(\frac{V_{ic}}{a_{SL}}\right)^2\right)^{7/2} - 1\right)$$
$$q_{c_{ic}} = 14.696\left(\left(1 + 0.2\left(\frac{305}{661.48}\right)^2\right)^{7/2} - 1\right) = 2.3058 \text{ [psi]}$$

Now, to correct $q_{c_{ic}}$ we will need ΔP_p, which can be found by multiplying (16) by P_s. This is where the fact we were told a reference altitude to use matters. In this specific homework problem, we were asked to use the "test altitude," which means we already have P_s from (15). If we were asked to use a different reference altitude, we would have to find a new P_s for that altitude using (15) with a different H_{ic}.

$$\begin{aligned}
\underbrace{q_c}_{P_T-P_a} &= \underbrace{q_{c_{ic}}}_{P_T-P_s} + \underbrace{\Delta P_p}_{P_s-P_a} \\
q_c &= q_{c_{ic}} + (\Delta P_p/P_s)P_s \\
q_c &= 2.3058 + 0.0032(13.6669) = 2.3502 \text{ [psi]} \\
V_c &= a_{SL}\sqrt{5\left(\left(\frac{q_c}{P_{SL}}+1\right)^{2/7}-1\right)} \\
V_c &= 661.48\sqrt{5\left(\left(\frac{2.3502}{14.696}+1\right)^{2/7}-1\right)} = 307.77 \text{ [kts]} \\
\Delta V_{pc} &= V_c - V_{ic} \\
\Delta V_{pc} &= 307.77 - 305
\end{aligned}$$

$$\boxed{\Delta V_{pc} = 2.77 \text{ [kts]}} \tag{17}$$

All that's left is to compute M_{ic} and M_c in order to arrive at ΔM_{pc}. We'll use (9) and our results from above to compute both.

$$M_{ic} = M(q_{c_{ic}}/P_s) = M(2.3058/13.6669)$$

$$\boxed{M_{ic} = 0.4772} \tag{18}$$

$$\begin{aligned}
M_c &= M(q_c/P_a) = M(2.3502/13.6226) = 0.4823 \\
\Delta M_{pc} &= M_c - M_{ic} = 0.4823 - 0.4772
\end{aligned}$$

$$\boxed{\Delta M_{pc} = 0.0051} \tag{19}$$

Finally, we compute the last parameter using

$$\begin{aligned}
\Delta P_p/q_{c_{ic}} &= (\Delta P_p/P_s)P_s/q_{c_{ic}} \\
\Delta P_p/q_{c_{ic}} &= (0.0032)(13.6669)/2.3058
\end{aligned}$$

$$\boxed{\Delta P_p/q_{c_{ic}} = 0.019} \tag{20}$$

4 Summary

In this quick review we have walked through a notional ADS calibration problem where we used an altitude method (the TFB) to arrive at an altitude error for a particular flight condition. Next, we used that altitude error to compute the underlying pressure error. Finally, we used pressure error and a reference altitude to infer other instrument errors such as airspeed and Mach number. It is important to remember two key takeaways: (i) the TFB is simply one method of obtaining this information and happens to be altitude-based, and (ii) classical methods such as the TFB, cloverleaf, and level turn all require multiple test points at stabilized flight conditions, while newer methods have focused on inferring the entire curve from a single test point. Choosing a method for your program should include considerations for resources available, supersonic requirements, and level of accuracy required.

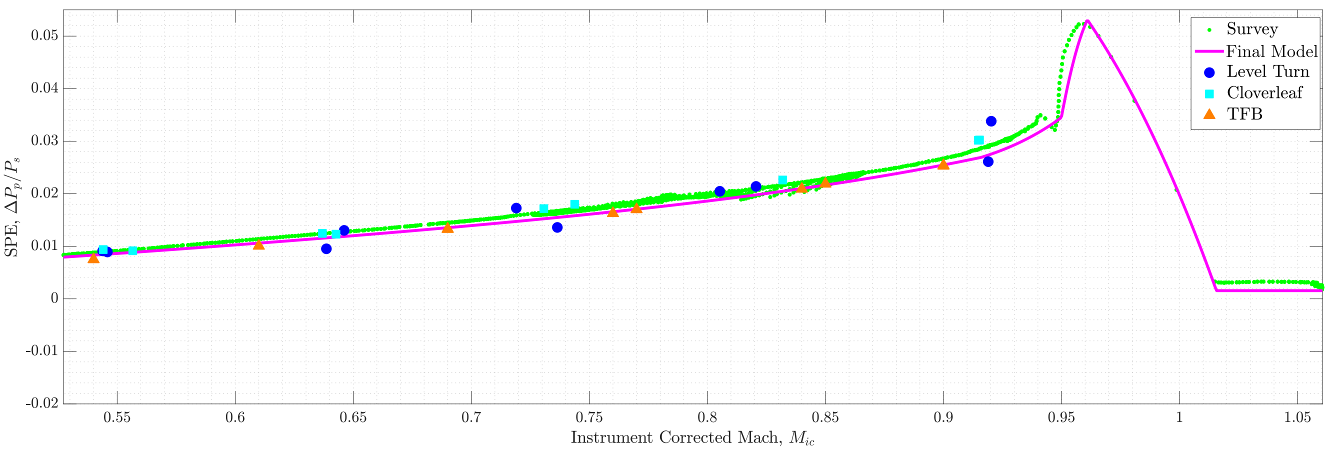

Figure 3: T-38 Static Source Position Error Curve

5 Sample MATLAB Code

```matlab
% ED Quick Review: Pitot-statics
% Sample MATLAB Code
% Written by Juan Silv Jurado, Fall 2019

clear; clc; close all;

% Problem Setup
towerHc = 2010; % Pressure altitude in tower, ft
gridReading = 2.5; % Tower theolodite reading, unitless
towerTa = 29.4; % Tower ambient temperature, deg C
gridConstant = 31.4; % Tower theolodite constant, ft/dif

% Test Point Data
deltaHic = -22; % Altitude instrument correction, ft
deltaVic = 3.0; % Airspeed instrument correction, kts
Hi = 2017; % Indicated altitude, ft
Vi = 302; % Indicated airspeed, kts

% Problem: find Hic, Vic, Hc, deltaHpc, deltaPp_Ps
% Step 1: Instrument corrections
Hic = Hi + deltaHic;
Vic = Vi + deltaVic;
fprintf('Hic: %0.3f ft\n',Hic);
fprintf('Vic: %0.3f kts\n',Vic);

% Step 2: Altitude position-error correction
% Set up basic equations/functions (alts below 36K feet and subsonic)
pressureRatio_H = @(H)  (1 - 6.87559e-6*H).^5.2559; % delta(H)
tempRatio_H = @(H) 1 - 6.87559e-6*H;
Mach_QcPa = @(Qc,Pa) sqrt(5*((Qc/Pa+1)^(2/7)-1));
seaLevelTemp = 15 + 273.15; % deg K
seaLevelPressure = 14.695951733258404; % psi
seaLevelSoS = 661.48; % kts

% Geometric height of aircraft above tower
geometricDelta = gridReading*gridConstant;

% Convert geometric height difference to pressure altitude difference
Ttest = towerTa+273.15; % Test-day temp at tower, K
Tstd = seaLevelTemp*tempRatio_H(towerHc); % Standard temp at tower alt, K
deltaHc = Tstd/Ttest*geometricDelta; % feet PA

% Compute true pressure altitude at aircraft's fly-by altitude
Hc = towerHc+deltaHc; % True aircraft pressure alt, ft
fprintf('Hc: %0.3f ft\n',Hc);

% Compute altitude position error correction
deltaHpc = Hc-Hic;
```

```
fprintf('deltaHpc: %0.3f ft\n',deltaHpc);

% Now convert this altimeter error to pressure error
Pa = seaLevelPressure*pressureRatio_H(Hc);
Ps = seaLevelPressure*pressureRatio_H(Hic);
deltaPp_Ps = (Ps-Pa)/Ps;
fprintf('deltaPp_Ps: %0.4f\n',deltaPp_Ps);

% Step 3: Airspeed and Mach number corrections
% Compute Qcic from Vic and the aircraft altitude
Qcic = seaLevelPressure*((1+0.2*(Vic/seaLevelSoS)^2)^(7/2)-1);

% Correct Qcic into Qc using deltaP_p/Ps from the TFB
Qc = Qcic+(deltaPp_Ps*Ps);

% Use Qc to get Vc, or position corrected airspeed
Vc = seaLevelSoS*sqrt(5*((Qc/seaLevelPressure+1)^(2/7)-1));

% Obtain deltaVpc using Vic and Vc
deltaVpc = Vc-Vic;
fprintf('deltaVpc: %0.4f kts\n',deltaVpc);

% Next, compute Mic and Mpc using Qcic/Ps and Qc/Pa respectively
Mic = Mach_QcPa(Qcic,Ps);
fprintf('Mic: %0.4f\n',Mic);
Mc = Mach_QcPa(Qc,Pa);
deltaMpc = Mc-Mic;
fprintf('deltaMpc: %0.4f\n',deltaMpc);

% Finally, compute deltaPp/Qcic
deltaPp_Qcic = (deltaPp_Ps*Ps)/Qcic;
fprintf('deltaPp_Qcic: %0.4f\n',deltaPp_Qcic);
```

```
Hic: 1995.000 ft
Vic: 305.000 kts
Hc: 2083.731 ft
deltaHpc: 88.731 ft
deltaPp_Ps: 0.0032
deltaVpc: 2.7730 kts
Mic: 0.4772
deltaMpc: 0.0051
deltaPp_Qcic: 0.0192
```