AN ALGORITHM FOR USING A SLIDE-SCREW TUNER AS A COMPUTER-CONTROLLED IMPEDANCE

George Sloan
Sandia National Laboratories

SAND--90-1209C

DE90 010657

Introduction

In the testing of active microwave components, a common test procedure is to evaluate a device's performance when subjected to an all-phase, constant-standing-wave-ratio (APCS) load pull. Such a test specification is useful in verifying a device's stability and mismatch performance. Typically, APCS pulls are tediously performed by hand, with manually operated tuners. However, with the advent of mechanical, computer-controlled tuners, it is now possible to automate this procedure.

Two types of mechanical, automated tuners are currently available: double-slug and slide-screw. The two types have complimentary advantages and disadvantages. For example, a slide-screw tuner offers the advantage of having a removable probe, which enables the tuner to easily revert to a low-loss, 50 Ω line. A double-slug tuner, on the other hand, is always lossy due to its permanent slugs. However, the double-slug tuning mechanism does offer the advantage of being easily amenable to mathematical modeling, while the slide-screw mechanism is nonlinear and difficult to model. In fact, because of the slide-screw modeling difficulty, only the double-slug manufacturer supports the capability to operate a tuner as a true computer-controlled impedance.

The primary manufacturer-supported application for both tuners is to provide load pulls for the determination of power and noise contours. For this application, an impedance-finding capability is not necessary. For example, in power contouring, the test device's output power is measured and recorded at previously characterized tuner positions (which represent impedances randomly scattered over the Smith chart). Surface-fitting and contouring algorithms are then applied to the resulting data to calculate the constant power contours. Hence, the contouring software correlates power to impedance, not impedance to tuner positions, as would be needed for an impedance-locating capability. Unfortunately, it is an impedance-locating capability that is specifically needed to perform an APCS pull.

At Sandia, our goal was to integrate an APCS pull capability into a multi-test, single-connection tester. (The single-connection concept implies that many tests, such as network analysis, spectral analysis, and noise figure measurements can be made from a one-time, device-to-tester connection.) Consequently, the slide-screw tuner was the obvious choice due to its removable

probe capability. Hence, it became necessary to develop a custom algorithm capable of utilizing the tuner in an impedance-finding mode. The general concept used in implementing this capability was to empirically characterize the tuner over an acceptable range of tuner positions, and then use this characterization to intelligently predict the tuner positions needed to present the desired impedance.

Tuner Characterization/Calibration

The slide-screw tuner used by Sandia consists of a "trough" transmission line, a motor-controlled probe, and a motor-controlled carriage. The carriage is designed to carry the probe as it traverses the transmission line. The tuner's response (reflection coefficient) is controlled by the probe's insertion into the trough (which primarily varies magnitude) and by the carriage's position along the line (which primarily varies phase). The nonlinear nature of the tuner's response can be broken down into three main areas: 1) magnitude sensitivity to probe insertion, 2) phase variation with probe insertion, and 3) magnitude variation with carriage position. Hence, any attempt to characterize the tuner must account for these response deviations.

The Sandia tuner characterization procedure operates by creating a "calibration grid" of the tuner's response. The grid is created by measuring the tuner's reflection coefficient at equal, even steps of motor positions. The motor movement ranges are determined by the magnitude and phase of the grid points which which form the cater-cornered grid vertices. In other words, the calibration grid is specified by translating a magnitude and phase range into corresponding motor movement ranges. The general steps used in the calibration procedure are as follows:

1) Tuner is "zeroed" (reference motor positions defined).
2) Operator enters the calibration frequency, grid density (number of carriage and probe motor positions), and calibration window size (magnitude and phase ranges).
3) Tuner software intelligently searches for the calibration window corners that define the motor movement limits and records the corresponding motor positions.
4) Tuner software computes the motor steps corresponding to the grid intervals and moves the tuner (in an efficient manner) to each grid point, where it measures and records the tuner's response.

Step 1 of the algorithm is implemented by moving the carriage and probe motors to their negative movement limits; this procedure fully inserts the probe and moves the carriage to the reference end of the trough. Each tuner motor detects its motion limits via an optical switch. However, because the switch's trigger position can vary slightly with motor speed, a repeatable "zero position" reference is obtained by backing out the motor in single-step increments until the optical switch de-triggers.

In searching for the calibration window vertices (step 3), the tuner needs to know each motor's sensitivity and the carriage's phase orientation (whether positive carriage movement results in a positive or negative phase change). The carriage's sensitivity can be dynamically determined by simply moving the carriage a fixed number of steps and then measuring the magnitude and polarity of the resulting phase change. The probe's sensitivity can also be measured, but the resulting value would only be accurate for a given insertion range. This is because the probe's sensitivity is a function of its insertion. The algorithm resolves this problem by using a fixed, average sensitivity value over the entire insertion range.

Once the tuner's orientation and sensitivities are known, the algorithm can search for the calibration window vertices using the following iterative search procedure:

(New position) = (old position) + (sensitivity)(response error)

where the response error is simply the difference between the target magnitude/phase and the current magnitude/phase. The iteration continues, alternating between motors, until the magnitude and phase response errors are acceptably small. Fig. 1 illustrates the general tuner path used in locating the calibration window vertices. Starting at the zero position, the carriage is moved until the first phase target (θ_1) is found; next, the probe is inserted until the first magnitude target (m_1) is found. At this point, the carriage and probe are alternately adjusted until the magnitude/phase errors are acceptably small. The procedure is then repeated for the second calibration window vertex. Note that if the carriage has a positive orientation (Fig. 1(a)), θ_1 will be closest to the zero position, but if the orientation is negative (Fig. 1(b)), θ_2 will be closest to the zero position.

Fig. 2 is a plot of a 7x7 calibration grid made at 15 Ghz for a -20 to 380 degree phase range and a 0.7 to 0.4 magnitude range. Note that the tuner's nonlinearities are easily seen in the plot. The ripples in the horizontal curves represent magnitude variation with carriage movement; the ripples in the vertical curves represent phase variation with probe insertion, and the unequal spacings between the horizontal curves represent variations in the probe's magnitude sensitivity.

Three additional points should also be made relative to the tuner's calibration. First, the calibration window should be specified to be slightly bigger than the desired window; in other words, the window margin should be big enough to compensate for the tuner's nonlinearities (a ±10° phase and a ± 0.05 magnitude margin is usually sufficient). Second, the software must convert the 0 to ±180 degree phase measurement of a network analyzer to a strictly increasing phase relative to θ_1. Third, note that if the tuner's electrical length is not greater than twice the desired phase range, then it might be necessary to convert the phase range to one that is compatible with the tuner's achievable phase range, relative to θ_0. For example, if a 0-360 degree phase range is desired, but the realizable tuner range is 10-400, then the desired range should be changed from 0-360 to 10-370. This change can be made invisible to the operator, both during calibration and during impedance prediction.

Impedance Prediction

The basic concept used by the algorithm to predict tuner impedances is to locate the square of the calibration grid that bounds the desired impedance and then two-dimensionally interpolate within that square to determine the desired motor positions. The two-dimensional interpolation algorithm used is a custom technique that the author has named pseudo-2D interpolation. The general steps used in the procedure are as follows:

1) Search calibration grid for the grid square that bounds the desired impedance on two intersecting sides.
2) Interpolate/extrapolate along the vertical segments to find where the desired magnitude lies along the segments.
3) Interpolate/extrapolate along the horizontal segments to find where the desired phase lies along the segments.
4) Derive an equation for the line that connects the interpolation-derived magnitudes of step 2.
5) Derive an equation for the line that connects the interpolation-derived phases of step 3.
6) Solve for the desired motor positions by finding the intersection of the lines derived in steps 4 and 5.

Fig. 3 is an illustration of the above procedure: the x's represent carriage position; the y's represent probe position; the m's represent magnitudes; the θ's represent phase; the subscripts 1,2,3 and 4 represent the vertex numbers of a grid square, and the subscript "t" represents "target" or desired values.

Step 1 is implemented by systematically searching the calibration grid until a square is found where

$$\theta_3 \le \theta_t \le \theta_4 \quad \text{or} \quad \theta_1 \le \theta_t \le \theta_2$$

and

$$m_3 \le m_t \le m_1 \quad \text{or} \quad m_4 \le m_t \le m_2$$

Step 2 is next implemented by interpolating between m_1 and m_3 to find the position of m_{13} (=m_t), and by interpolating between m_2 and m_4 to find m_{24} (=m_t). For instance, the m_{13} interpolation is

$$y_{13} = (m_t-m_3)(y_{12}-y_{34})/(m_1-m_3)$$

Procedurally, step 3 is identical to step 2; the only difference is that phase values and horizontal segments are used in the interpolations. Note that step 1 forces at least one of each of the m_{ij} and θ_{ij} determinations to be true interpolations: The other two m_{ij},θ_{ij} determinations could possibly be extrapolations. The likelihood for extrapolations increases as the grid density increases, and as the desired impedance nears an actual grid point.

Step 4 consists of finding the equation of the line that connects m_{13} to m_{24}. The resulting equation is

$$y = (y_{24}-y_{13})(x-x_{13})/(x_{24}-x_{13}) + y_{13}$$

The corresponding equation for step 5 (line that connects θ_{12} to θ_{34}) is

$$y = (y_{12}-y_{34})(x-x_{34})/(x_{12}-x_{34}) + y_{34}$$

Lastly, the desired motor coordinates (x_t,y_t) are found by finding the common solution or intersection of the above equations. This point can be found by simple substitution or by matrix techniques, whichever is more convenient/efficient for the programmer to implement.

Fig. 4(a) is a plot of an APCS pull performed via the above algorithm. (The pull is performed in 5° steps for $|\Gamma| = 0.6$ and uses the calibration grid illustrated in Fig. 2.) The plot shows that the magnitude was held to within 0.01 units of the desired pull value. Fig. 4(b) is a plot of the magnitude and phase errors at each pull position. This plot shows that the phase error ranged from -0.5 to 1.2 degrees. Results for any pull within the calibration window produced similar accuracies (absolute magnitude error < .01, absolute phase error < 1.2°).

If the accuracy obtained using a given calibration grid is not sufficient, the grid density can simply be increased. Selection of the appropriate grid density depends on the accuracy desired, the size and magnitude range of the calibration window, and on the extent of the tuner's nonlinearity. The time required for calibration is only a secondary consideration in selecting grid density. This is because the automated calibration procedure is relatively quick; the 7x7 grid of Fig. 2 took less than 1.5 minutes to create.

The most accurate tuner response is obtained for reflection coefficient magnitudes that are less than 0.75 (the tuner is capable of magnitudes up to 0.9). Pull tests for magnitudes greater than 0.75 were found to have errors as much as 50% greater than those for magnitudes less than 0.75 (for similar calibration grid densities). This is primarily because the nonlinearity of the tuner's response increases significantly as the probe nears full insertion. Additionally, the tuner's linearity is a strong function of its opposite-port load. If the tuner's load is a poor match, then the tuner's nonlinearity will be significantly increased; hence, a denser calibration grid will be needed to obtain the same accuracy that would have been possible with a more ideal load. The effect of a poor load on tuner response can be predicted via the s-parameter equation for the reflection coefficient of a two-port network (in this case the tuner) connected to an arbitrary load (Γ_L):

$$\Gamma_{tuner} = S11 + (S_{21}S_{12}\Gamma_L)/(1-S_{22}\Gamma_L)$$

If Γ_L is ideal (Γ_L=0), then the tuner's response is its intrinsic S_{11}. However, if Γ_L is non-ideal, then the tuner's response is a function of the load and all the tuner's s-parameters, which in turn are functions of the tuner's position. It is easy to see how the nonlinear nature of the magnitude and phase solutions of the above equation could increase the nonlinearity of the tuner's magnitude/phase response.

Conclusion

Perhaps the most significant characteristic of the APCS algorithm is its simplicity. Before developing the algorithm, several other techniques were attempted. For example, one procedure the author attempted was based on performing least-squares paraboloid surface fits of the the grid area that bounded the desired impedance. This technique yielded errors that were two to three times greater than those resulting from the APCS algorithm. While some of the other techniques produced acceptable results, none were as simple or computationally efficient as the APCS algorithm.

Currently, the APCS algorithm is being used in an integrated, single-connection tester to perform an all-phase, constant-SWR load pull. While the APCS algorithm was designed specifically for this task, it is versatile enough to be adaptable to many other applications. The key to its versatility is that the algorithm essentially allows the tuner to function as a computer-controlled impedance. In fact, one application for the algorithm would be to use it in conjunction with the manufacturer's power/noise contouring software. The manufacturer's software could be used to predict the optimum noise/power impedance, and the APCS algorithm could be used to position to that impedance for a real-world verification test. Hence, the APCS algorithm appears to be useful both to enhance existing software and to implement unique, custom applications.

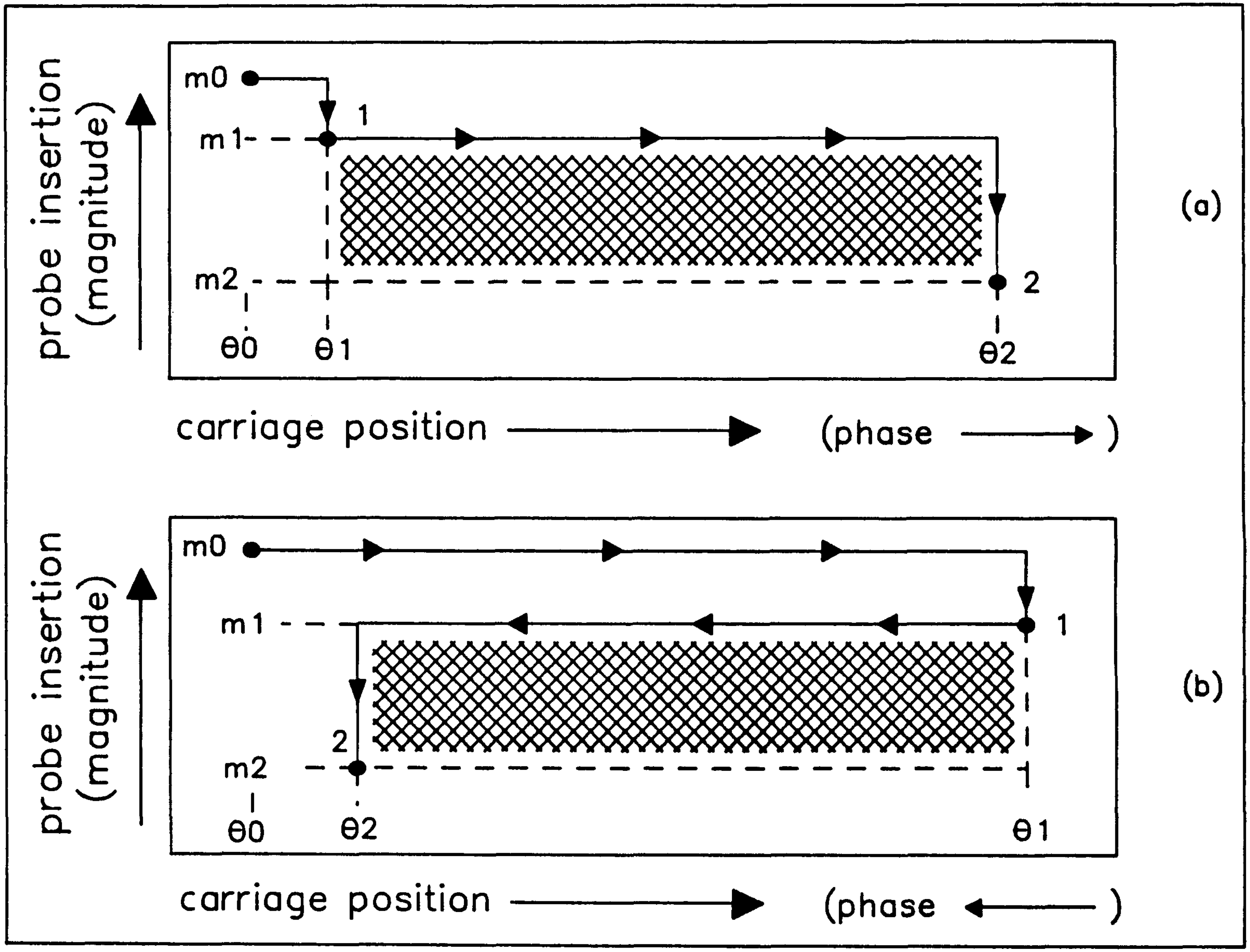

Figure 1. Tuner paths for locating calibration window vertices. (a) Positive phase orientation. (b) Negative phase orientation.

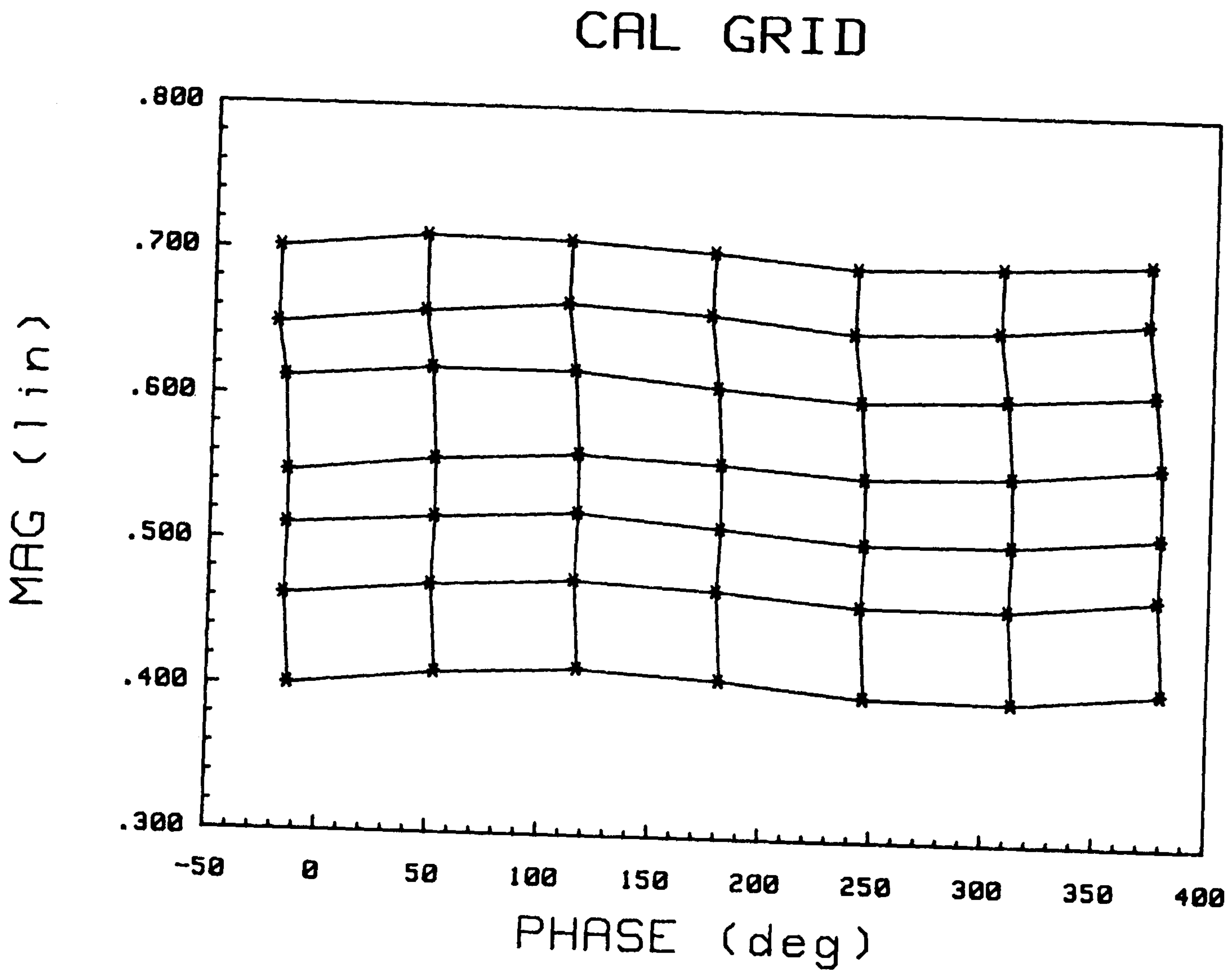

Figure 2. Calibration grid (7x7).

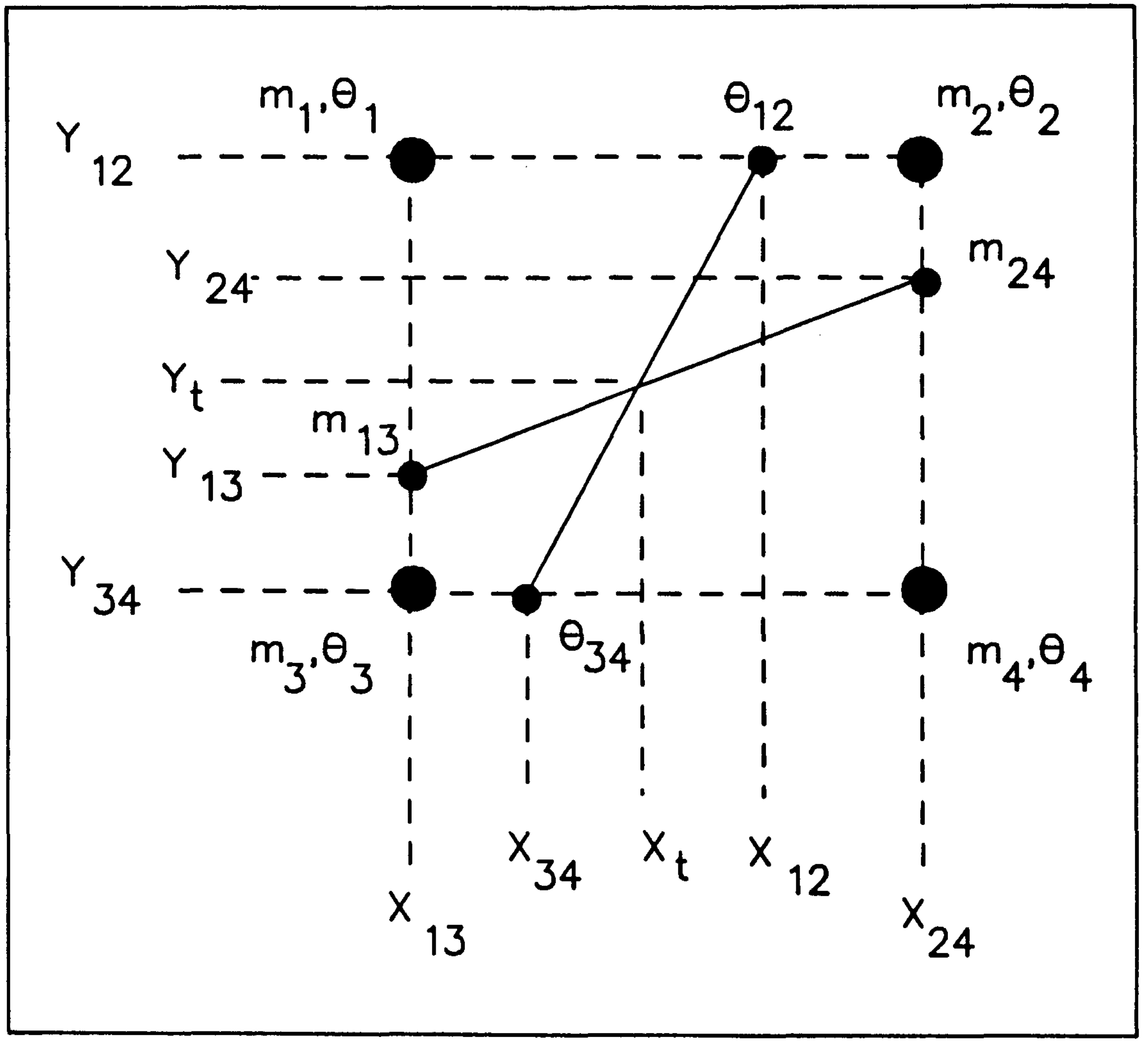

Figure 3. Pseudo–2D interpolation of grid square.

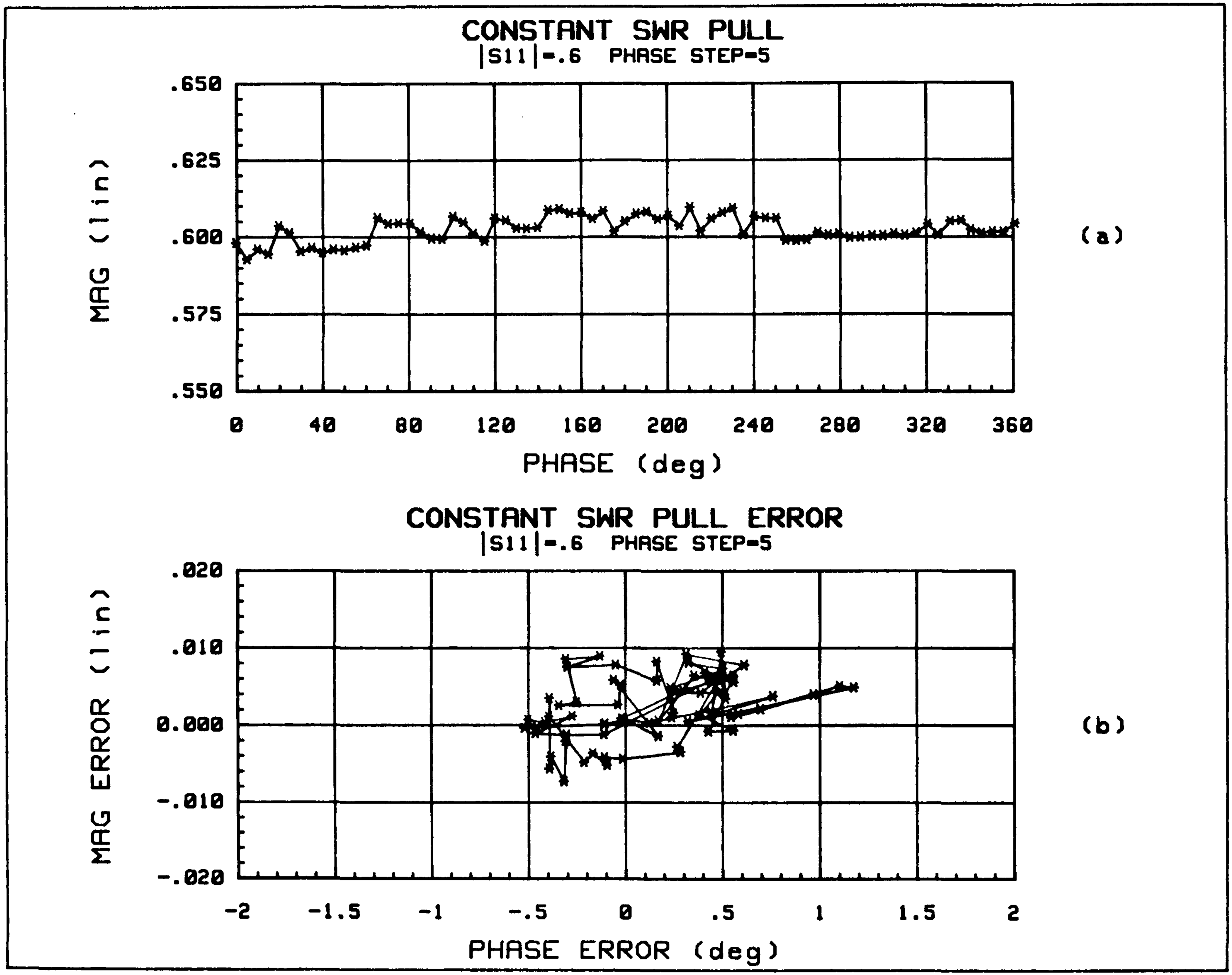

Figure 4. Tuner implementation of APCS pull. (A)Tuner response. (b)Pull error.