To: TEMRV Distribution

From: Lucas Chavez, Civ AFRL/RDLE

Subject: TEMRV 20110817-xxx: Optimized alignment of specular specimens for TMARS

Date: September 30, 2011

Keywords: Calibration, Active Tracking, Lasers, Modeling and Simulation, Scatterometry

S drive location: S:/DEP/TEMRV (Technical Engineering Memos)

---

1 Introduction

Many applications of directed energy involve identifying and tracking a desired target. The Air Force Research Lab (AFRL) at Kirtland is currently at the forefront in research and development of these types of tracking systems. Current methods rely on measurements returned from the reflection on the target. In order to interpret these measurements, the tracking system must model the reflection with a Bidirectional Reflectance Distribution Function (BRDF). This model is dependent on many variables:

- Optical properties of the material being lazed
- Temperature of the target
- Shape of the lazed surface
- Spatial relations between the laser, target, and detector

These complexities can make the problem of tracking very difficult. In order to facilitate ongoing research into the problem, AFRL has designed and developed a test bed which allows measurements to be taken in a controlled environment.

The Target Material Angle Resolved Scatterometer (TMARS) is a highly resolved 6 DOF laser scatterometer which allows control over temperature and positioning of a test specimen during lazing. The system is comprised of three main components:

- Source – Position is always fixed
- Specimen – Position is controlled with 3 linear actuators and 3 rotational actuators
- Detector – Position is controlled with a single rotational actuator

One test TMARS is routinely used to perform is named the Theta-2Theta scan. The test assumes the laser, detector, specimen surface, and center of rotation are all aligned at a single point. Figure 1 shows a top-down view of the test set up. For this particular test, 2 rotational actuators are used; 1 for the specimen and 1 for the detector. During data acquisition, the specimen is rotated through a range of angles while the sensor is rotated by twice that range. Ideally, this allows the sensor to “track” the specular reflection from the specimen.

Setup for any test, including the Theta2-theta scan, involves the specimen being mounted on TMARS by a human operator. The operator must manually move the actuators so that the axis of rotation lines up with the surface of the specimen. This manual alignment is subject to error as depicted in the Figure 2. Small errors in positioning are magnified at a distance and lead to large errors at the sensor. These errors become apparent during a Theta-2Theta scan.

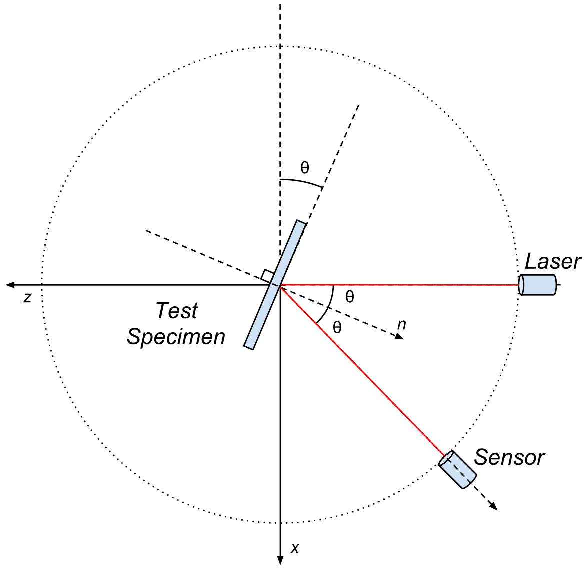


Figure 1: Test set-up

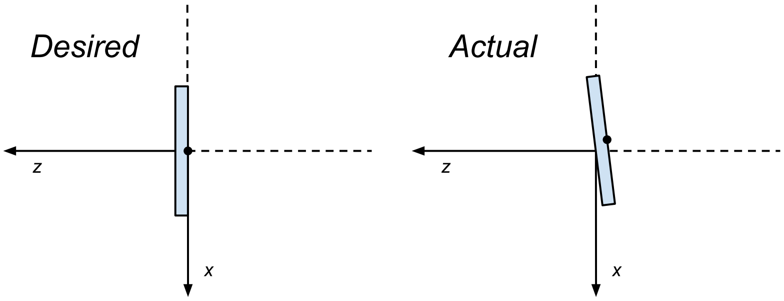
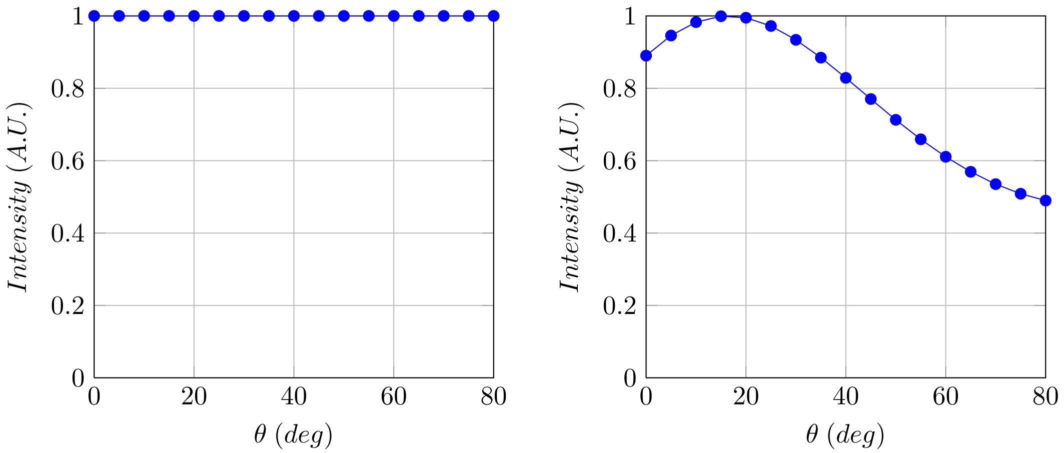


Figure 2: Positioning error

Figure 3 shows why the Theta-2theta scan is an ideal test to expose misalignments in the system. The plots are generated from simulations which take into account the geometry of the system and model the laser as a Gaussian beam. Plot (a) shows the result of a perfect alignment. As the specimen is rotated from 0° to 80°, the sensor is moved to double the current angle of the specimen and a measurement is taken. We can see that the specular peak of the reflection is seen by the detector at each increment, and the result is ideally a constant of 1 throughout the scan. Plot (b) shows what happens when there is a slight misalignment in the system. In this case approximately 2 mm in both translations and less then .1° in rotation. We can see that, due to the complex geometry of the problem, the scan can dramatically change due to small changes in alignment.



(a) Correctly aligned

(b) Misaligned

Figure 3: Plot of two Theta-2theta scans showing the difference between aligned and misaligned specimens

2 Objective

Our goal is to design an algorithmic method to optimize the final positioning of the specimen by leveraging the information contained within the Theta-2theta scan. The proposed method should require no additional hardware and should be automated in order to remove human error.

We will be using calibration and optimization procedures from the field of robotics. These techniques have been developed with the sole purpose of calibrating machinery while minimizing time and effort. The techniques are data driven and usually work on the principle of minimization of some error function. The error function is based on modeling and simulation of the system. This approach of modeling and optimization through some minimization technique has been proven to obtain reliable results for a variety of calibration problems. One such example is given in [1].

3 Approach

3.1 Modeling and Simulation of a Theta-2theta scan

The first step in designing an optimization procedure was to develop a kinematic model of the system. This model allows us to simulate and control the geometric relations between the source, specimen, and sensor. Each of these components has its own coordinate frame. Transformation matrices are used to relate one coordinate frame to another. Figure 4 describes the dependent relations between the coordinate frames and specifies the variables which are used to control the transition. The following list describes some key points of the model:

- Base frame – The base frame is defined to have the center of rotation at its origin and the laser along its z-axis.
- Sensor – The sensor rotates about the center of rotation at a radius of 125 cm. The variable θ_d describes the rotation from the base frame.
- Specimen mount – The mount can be rotated and translated from the base frame with actuators. The variables θ_c, X_c, and Z_c describe this movement.

- Specimen surface – There is a 3 DOF transition from the mount to the face of the specimen. The variables θ_s, Z_s, and X_s describe this translation.

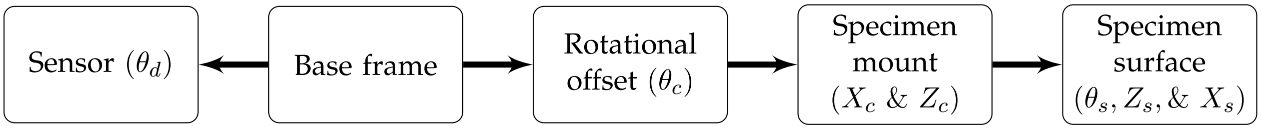

Sensor (θ_d)
Base frame
Rotational offset (θ_c)
Specimen mount (X_c & Z_c)
Specimen surface ($\theta_s, Z_s, \&\ X_s$)

Figure 4: Kinematic flowchart of the system

Fundamentally, the kinematic model is a function which outputs the intensity observed by the sensor. This function has two types of inputs: The joint parameters ($\theta_d, \theta_c, X_c, \&\ Z_c$) which we can control; and the specimen parameters ($\theta_s, X_s, \&\ Z_s$) which we assume are unknown. These inputs and the resulting transformation matrices are used to determine the distance from the reflected source beam to the sensor. The intensity at the sensor is then calculated with a Gaussian beam assumption.

In order to simulate a Theta-2theta scan, the kinematic model is evaluated in a for loop. Each iteration of the for loop changes the value of θ_c & θ_d with $\theta_d = 2 \times \theta_c$. The result is an array of intensity values corresponding to the array of angles chosen for θ_c. Figure 3 provides an example of the typical output resulting from a simulated Theta-2theta scan. The model was developed using [2, 3]

3.2 Simulation of real world data

In reality, the optimization procedure would be initiated after taking real word data with the actual system. For this study we will be investigating the feasibility of the algorithm by simulating the data which would of come from the real world system. In order to simulate the real world data we will be utilizing the kinematic model of the system. The steps displayed on the gold background of Figure 5 provide a flowchart describing the step by step procedure for simulating the real world data.

The first step corresponds to obtaining a new specimen with unknown specimen parameters. These are the parameters which will searched for by the optimization procedure. The second step represents the operator mounting the specimen and moving the actuators so that the specimen surface is nearly at the desired location. The third step requires generating N Gaussian perturbations about the initial state. Finally, the actuators are moved to each perturbation and a Theta-2theta scan is simulated. These scans and their associated joint parameters constitute the observed data.

3.3 Optimization

The last step in Figure 5 refers to the actual numerical optimization. The first step towards implementing the numerical optimization is the design of an error function (E). The desired error function has the following characteristics:

1. Must be able to quantify the difference between the observed data and the models prediction

2. Must contain a global maxima where there is no difference between the observed intensities and the intensities predicted by the model

3. Should be relatively smooth to help the optimization algorithm

4. Should have small values far from the optimal solution

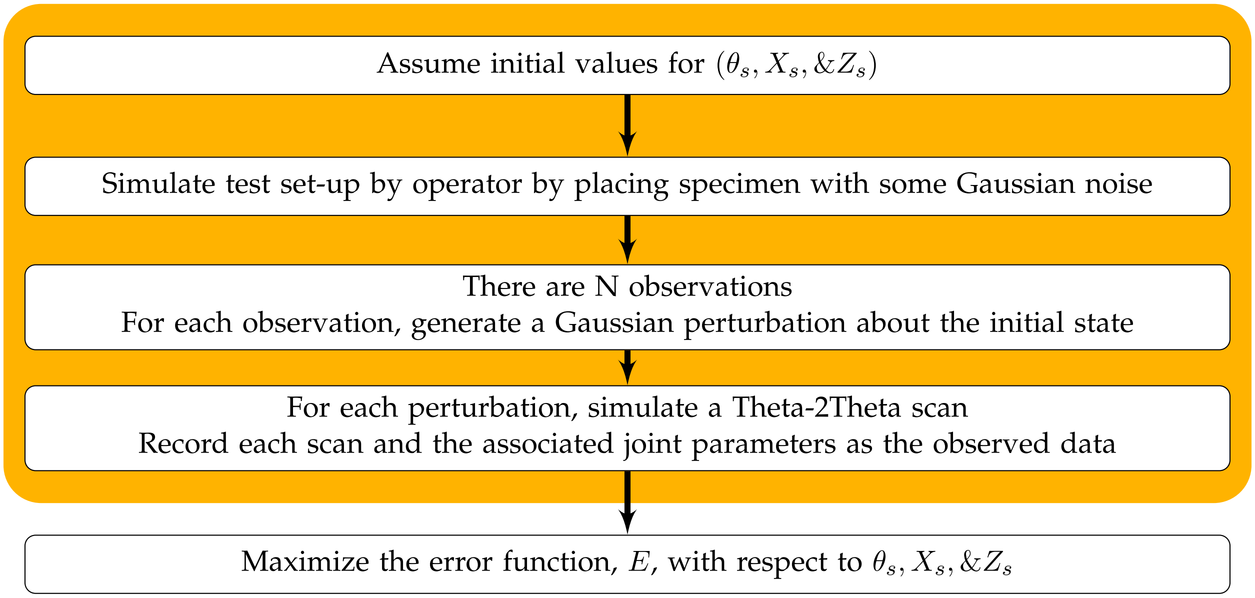

Assume initial values for $(\theta_s, X_s, \& Z_s)$
Simulate test set-up by operator by placing specimen with some Gaussian noise
There are N observations
For each observation, generate a Gaussian perturbation about the initial state
For each perturbation, simulate a Theta-2Theta scan
Record each scan and the associated joint parameters as the observed data
Maximize the error function, E, with respect to $\theta_s, X_s, \& Z_s$

Figure 5: Optimization procedure

We will be discussing two different error functions which were considered. Each error function had the same general form as Equation 1. Where D_i stands for an array of size N. Each element in the array comes from a calculation which tries to quantify the difference between the observed intensity $(I^{observed})$ and the corresponding intensity as predicted by the model (I^{model}). The two error functions are distinguished by how D_i is calculated.

$$E = \sum_{i=1}^{N} e^{-D_i} \tag{1}$$

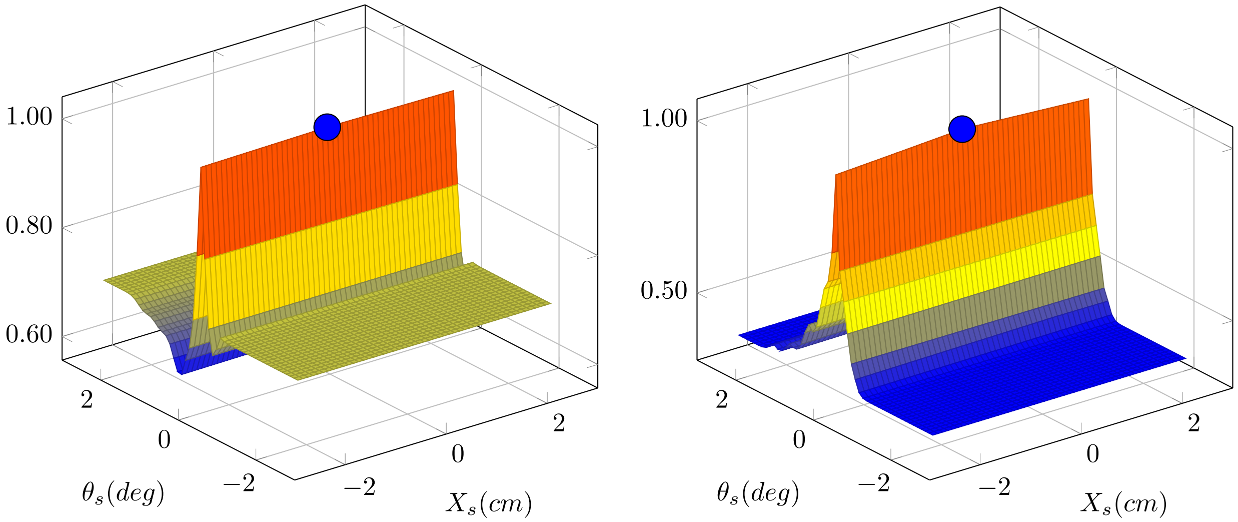
In the first error function which was considered, D simply relied on the average of the absolute difference between the scans. The calculation for D can be seen in Equation 2. Through testing it was found that this error function had poor performance for items 3 and 4 from the above list.

$$D_i = mean \left| I_i^{observed} - I_i^{model} \right| \tag{2}$$

Due to poor performance from the first error function, it was redesigned in order to take into account information regarding the shape of the scan. The idea was, if the observed scan and the model prediction have very similar shape but are offset by some amount they should be rewarded more. In order to quantitatively describe the similarity in shape between the scans, the cross-covariance function was chosen to be part of the error function [4, 5]. Equation 3 shows the redesigned engine of the error function. Let $c_{x,y}$ stand for the unbiased cross-covariance between signals x and y. Also, let o stand for "observed" and m stand for "model."

$$D_i = \frac{max\left[c_{o,m}\right]}{max\left[c_{o,o}\; c_{m,m}\right]} \tag{3}$$

Figure 6 gives a comparison of the two error functions. The blue dot corresponds to the optimal solution. Each error function surface was computed using the exact same data. We can see that the first error function has several valleys. This valleys can be very deceptive to optimization algorithms which rely on the gradient information. Also, in comparison to the final error function, the first error function has relatively large values far from the optimal solution. This will also have an adverse affect on the optimization algorithm. By redesigning the error function we were able to make a more tractable optimization problem.



(a) First error function

(b) Final error function

Figure 6: Comparison of the error functions

Once the final error function was chosen, we were able to decide on which optimization algorithm would be used. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) minimization algorithm was chosen to perform the computation. The BFGS was chosen because it is a very popular method for solving nonlinear optimization problems which lack constraints. Also, a very reliable Matlab implementation of the algorithm is available freely online [6].

4 Results

We will be going through a single case study to understand the results from the optimization procedure. First, the specimen parameters had to be arbitrarily chosen. For this case study, the specimen parameters were chosen to be:

$$[X_s\ Z_s\ \theta_s] = [.1(cm)\ .1(cm)\ .2(deg)]$$

Mounting the specimen by the operator was then simulated by taking the negative of the specimen parameters and adding noise. Which resulted in:

$$[X_c\ Z_c\ \theta_c] = [.9067(cm)\ -1.9216(cm)\ .1047(deg)]$$

Next, a value for N and an array of angles to rotate the specimen through had to be chosen:

$$N = 10 \quad \& \quad angles = 0:2:80$$

After all parameters were chosen, the observations could be simulated. First, each permutation was generated by adding gaussian noise to the control parameters. A standard deviation of 1.5

cm was chosen for X_c & Z_c and .5 deg for θ_c. For each permutation, a Theta-2theta scan was simulated. Figure 7 plots each of the simulated scans. These scans and their associated control values were recorded as the observed data.

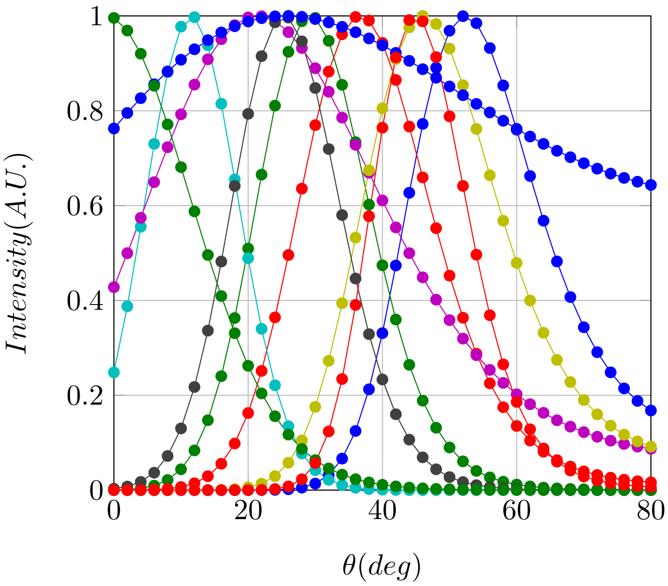


Figure 7: Plot of the simulated observations

Next, the numerical optimization was performed. The observed data was supplied to the BFGS algorithm. The initial starting point was set to be the initial position set by the operator. With these values set, the BFGS algorithm was run. The results can be seen below. The optimization took 42 seconds and made 303 calls to the error function. The error was driven from an initial value of .541 to a final value of 1.

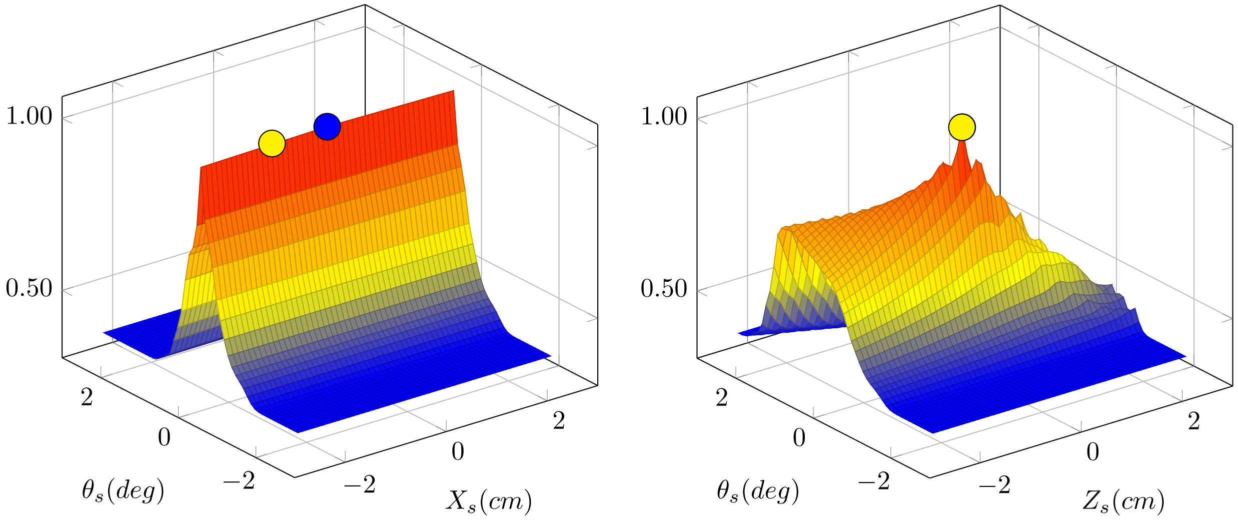
$$[X_s\ Z_s\ \theta_s] = [.9932(cm)\ .0962(cm)\ .2(deg)] \quad (optimized)$$

In order to visualize the results of the optimization, the surface of the error function was calculated and can be seen in Figure 8. There are two plots because we are trying to optimize 3 parameters and each surface plot can only show how the error function changes against 2 variables. It is worth noting, each surface plot took around 6 minutes to calculate and made 2,601 calls to the error function. One may think it would be possible to generate these plots and use them to pick the optimal solution instead of using a numerical optimization algorithm. We must remember, the plots could be generated because we had the “unknown” specimen parameters. Therefore, for each plot we set the variable which is not shown on the plot to the correct solution while the other two were varied. In an actual implementation this would not be possible.

Before discussing the results in Figure 8 let us discuss the ridge like feature in plot (a). This ridge corresponds to a line of solutions which satisfy the error function. There is a geometrical reason for this line of solutions and it comes from the fact that each point along the surface of the specimen is a valid solution. Meaning, if you refer back to Figure 1, when the specimen is aligned in Z and θ, X is free to move and each movement will result in an equally as valid Theta-2theta scan.

Now that we have some more understanding of the error function surface, we can use Figure 8 to visualize the result of the optimization. The yellow dot corresponds to the solution found by the BFGS algorithm. The blue dot corresponds to actual solution which was arbitrarily picked at the beginning of the simulation. In plot (b) these dots lie on top of each other while in plot (a) we can see the BFGS algorithm converged to a point on the ridge.

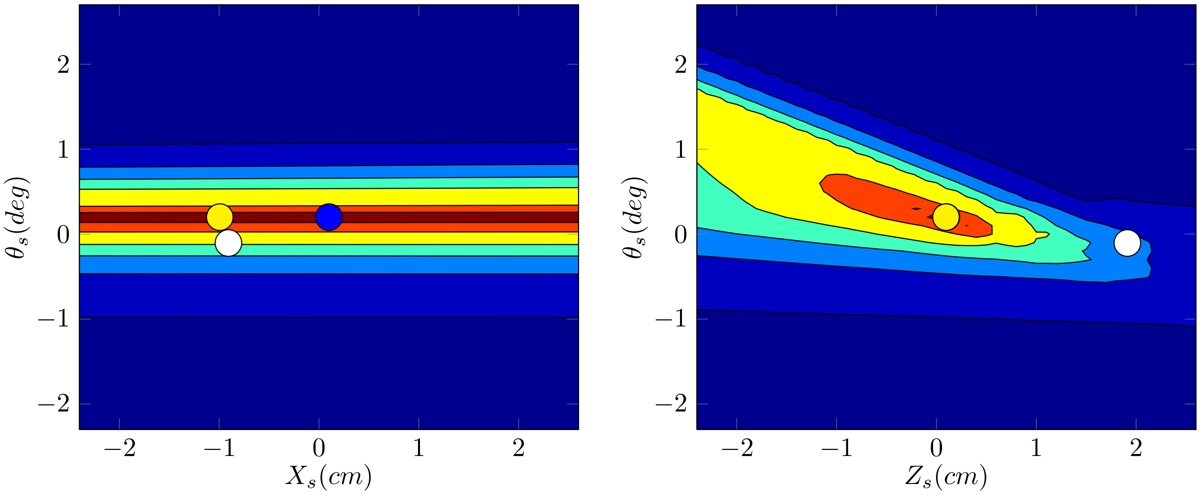
In order to better visualize the initial starting point of the optimization a contour plot of the error function was generated which can be seen in Figure 9. This plot was created with the same information as Figure 8. The yellow and blue dots still correspond to the optimized and correct solution respectively. However, in Figure 9 we can see an additional white dot which corresponds to the initial starting point of the optimization as set by the operator during mounting of the specimen.



(a) $X_s\ vs\ \theta_s$

(b) $Z_s\ vs\ \theta_s$

Figure 8: Results of the optimization on a surface plot of the error function



(a) $X_s\ vs\ \theta_s$

(b) $Z_s\ vs\ \theta_s$

Figure 9: Results of the optimization on a surface plot of the error function

5 Conclusion

It has been shown through initial testing and simulation that it is possible to calibrate the placement of specimens in TMARS with a completely automated method. Results have shown that the method is capable of improving the placement of the specimen with no additional input from the operator. A real world implementation of this system would be quite useful because it should improve the accuracy of results obtained from TMARS. However, we must remember that the method relies on several constraints which are summarized in the following list:

- Reflectance from the specimen must contain a specular peak. As the reflectance becomes more diffuse the method is adversely affected and, at a certain threshold, will not be able to converge.
- According to the error function, there is a line of acceptable solutions instead of a single point.
- The method calibrates only 3 of the 5 DOF of the specimen.
- The specimen is assumed to be flat in the local region around the point of the reflectance. Clearly, this may not be the case for many specimens.

Also, we must note there may be a numerical instability with the kinematic model. The ridge in plot (a) of Figure 8 has a slight downward slope. Meaning, that according to the error function, those points along the specimen surface are in slight disagreement with the observed data. In theory, these points should have no difference with the observed data, especially since this testing was done in a simulated environment. It is believed that the kinematic model may be unstable because it represents rotations between frames with matrices. It is known, when rotation matrices have the effect of canceling each other, the resulting rotation matrix can be non-orthogonal which adds error to subsequent calculations. I believe the kinematic model must be rewritten with quaternions used to represent rotations from frame to frame. Quaternions are mathematical constructs used mainly by the aeronautical society to describe orientation in space craft [7, 8]. They have many advantages over rotation matrices, including numeric stability. Initial work has been started towards rewriting the kinematic model and is included with the Matlab code.

References

[1] C.S. Gatla, R. Lumia, J. Wood, and G. Starr, "Calibrating pan-tilt cameras in robot hand-eye systems using a single point", in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3186–3191.

[2] T. Yoshizawa, *Handbook of optical metrology: Principles and applications*, vol. 10, CRC, 2009.

[3] B.E.A. Saleh and M.C. Teich, *Fundamentals of photonics*, John Wiley & Sons, 2007.

[4] L. Tan, *Digital signal processing: fundamentals and applications*, Academic Pr, 2008.

[5] H. Stark and J.W. Woods, *Probability and random processes with applications to signal processing*, Prentice Hall Upper Saddle River, NJ:, 2002.

[6] "FMINLBFGS: Fast Limited Memory Optimizer - File Exchange - MATLAB Central; http://www.mathworks.com/matlabcentral/fileexchange/23245-fminlbfgs-fast-limited-memory-optimizer".

[7] J.B. Kuipers, *Quaternions and rotation sequences*, Princeton university press Princeton, NJ, 1999.

[8] A.J. Hanson, "Visualizing quaternions", in *ACM SIGGRAPH 2005 Courses*. ACM, 2005, pp. 1–es.