PFI configuration simulator

Peter Mao, Caltech

February 12, 2013

Contents

Abstract

A PFI simulator is being developed at Caltech in order to model the configuration performance of the PFI Cobra motors. The major result out of this simulator is that, under the current error budget allocations, PFI would need approximately 30 iterations to put 95% of its fibers on-target. These results are presented in PDR backup slide, not in this document.

This document is a description of the major elements of the simulation model: the geometry, the Cobra movement model, and the incorporation of errors and uncertainties. In laying out the underpinnings of this simulator, I hope to convince the reader that the results

from this simulator represent a best-case scenario for the current implementation of PFI, albeit one that looks far worse than has been previously considered.

1 Introduction

The PFI/MCS simulator under development at Caltech predicts the number of iterations to configure the locations of the science fibers on PFI. Prior to this work, JPL tests of Cobra motions demonstrated 95% convergence of fibers to within 5 μm of targets in 8 iterations. Since measurement errors in the JPL setup were well below the 5 μm *allowable fiber placement error*, this data represents a best-case scenario. This simulation extrapolates the Cobra motion behavior determined in the JPL tests to conditions with more realistic (larger) measurement errors. We can use the results from this simulation to tell us both how well the system works under a given error allocation and the accuracies required to meet a specific performance goal.

The simulator consists of four major components: the geometry of the PFI bench (Section 2), the Cobra movement model (Section 3), the incorporation of error into the feedback loop and the specfics of the convergence criteria (Section 4).

2 PFI/Cobra geometry and (θ, ϕ) space

2.1 geometry

The PFI bench consists of 2394 Cobra motors on 42 rails arranged in a hexagonal pattern on a nominally 8mm pitch. In addition, there are 153 fixed fiducial fibers around the perimeter of the hexagon and along three radial spines. The nominal positions of these motors and fiducial fibers are defined in [DOCUMENT]. In this simulation, the motor and fiber mechanical tolerances are included by displacing each motor or fiducial fiber from its nominal position. These displacements are uniformly distributed within a 100 μm radius of the nominal positons.

Each Cobra motor has two arms — the lower is referred to as the θ arm and the upper is the ϕ arm. The θ arm has full range of motion, while the ϕ arm is restricted to a half-circle (less than that in the real system). The arms are nominally 2.375 mm long; in simulation their lengths vary, in a uniform distribution, from nominal by ± 250 μm for the θ arm and ± 100 μm for the ϕ arm. As the real system is built up, true positions and arm-lengths will be updated to reflect the real PFI. In the meantime, the distributions and magnitudes of the deviations from nominal are set by random number generators.

As in the case of the real system, because the Cobra arm lengths are unequal, each motor has an inaccessible region about its θ origin with a radius equal to the difference in arm lengths. Although unequal arm lengths are included in this simulation, I am not presently simulating any error in our knowledge of the arm lengths. That error is important in include at some future time, as I expect it will have non-negligible impact on configuration.

Presently, the geometrical setup of the simulation does not have any tilt or defocus components. The θ and ϕ axes of all of the motors are parallel to the system z-axis. Bench tilt and warp are NOT presently modelled.

2.2 target distribution $\rightarrow (\theta, \phi)$

One target is assigned to each science fiber, with the target locations (nominally) uniformly distributed in the patrol region of each Cobra. Note that this way of distributing the targets doubles the probability of having targets in two-way overlapping patrol areas and triples that probability in three-way overlapping areas.

Within a patrol region, the distribution is not exactly uniform mainly because target locations are specified by their local (θ, ϕ) coordinates rather than in bench coordinates. θ is uniformly distributed on the open interval $(0, 2\pi)$ (this really should be a half-open interval to allow the possibility of 0 or 2π). For equal arm lengths, the ϕ distribution would be:

$$\phi = \arccos(1 - 2U(0,1)), \tag{1}$$

where $U(0,1)$ is a uniformly distributed random variable on the open interval $(0,1)$, and at $\phi = 0$, the fiber is on the θ motor axis. For speed and readability, I use this distribution for the radial (ϕ angle) component of the target locations, even though it is not exactly correct for unequal arm lengths.

3 Cobra movement model

For the simulator to be meaningful, we need to have an accurate representation of how the Cobras move in terms of the angular error that remains after a move of some given angular request is executed. A simple model for the angular error distribution may consist of a sum of a distribution with a constant width with one whose width is proportional to the move request angle. The constant-width component represents the noise floor of the system, while the request-angle-proportional component incorporates the notion that smaller moves should have better accuracy.

Presently, there is not enough data on Cobra movements to support such a model. Until enough data are collected, I am modelling the Cobra movements as a Markov process, with the Markov tranisition matrix generated from existing Cobra movement data. In this section, after presenting the JPL data set, I will describe the Markov process model for Cobra movement.

3.1 JPL test data

In mid-2012, Charles Fisher (JPL) and Joel Kaluzny (JPL) ran 300 Cobra convergence sequences. Their experiment consisted of starting the Cobras from a (fixed|random) location, choosing a location in the patrol area from a uniform random distribution, and moving the Cobras to the target locations. The Cobra θ and ϕ angles were measured before and after each step and the sequence was completed when the fiber was measured to within 5μm of the target location. For each iteration, the angle to the target before the move is referred to as the *request angle* and the angle after the move is the *angular error*. Cobras can be operated in fast (2π/s) or slow ($\pi/2$/s) modes — this data set consists of 1312 individual Cobra moves, of which, 42 are fast-mode moves.

The entire data set is summarized in Figure 1. Qualitatively, one can see that the fast θ moves have a very tight error distribution, while the fast ϕ does not. Also, note that as the fiber approaches the target and the request angle becomes small, there is a significant probability for the angular error to be *larger* than the angular request after a move!

Given the limited statistics on the fast moves, I model the fast θ angular error with a Gaussian distribution with $\sigma = 50$ mRad. The fast ϕ move has a more complicated distribution, which is puzzling, given that the ϕ motor has a smaller inertial load than the θ motor. The ϕ angle error distribution is a request-angle dependent uniform distribution:

$$\Delta\phi_{\mathrm{err}} = U(0.036|\Delta\phi_{\mathrm{req}}| - 0.080, \quad 0.190|\Delta\phi_{\mathrm{req}}| - 0.060)\left(\frac{\Delta\phi_{\mathrm{req}}}{|\Delta\phi_{\mathrm{req}}|}\right), \quad (2)$$

Where $U(a, b)$ is the uniform distribution over the open interval (a, b), and the second term is the sign of the move request angle, needed to convert the angular error relative over/under shoot coordinates to true angular coordinates. The fast-move distributions are only used if either of the request angles are greater than 1 radian.[1] Typically this is only true for the first move, and occasionally on the second. The slow-move distributions, which comprise the bulk of the JPL data is discussed in the next section.

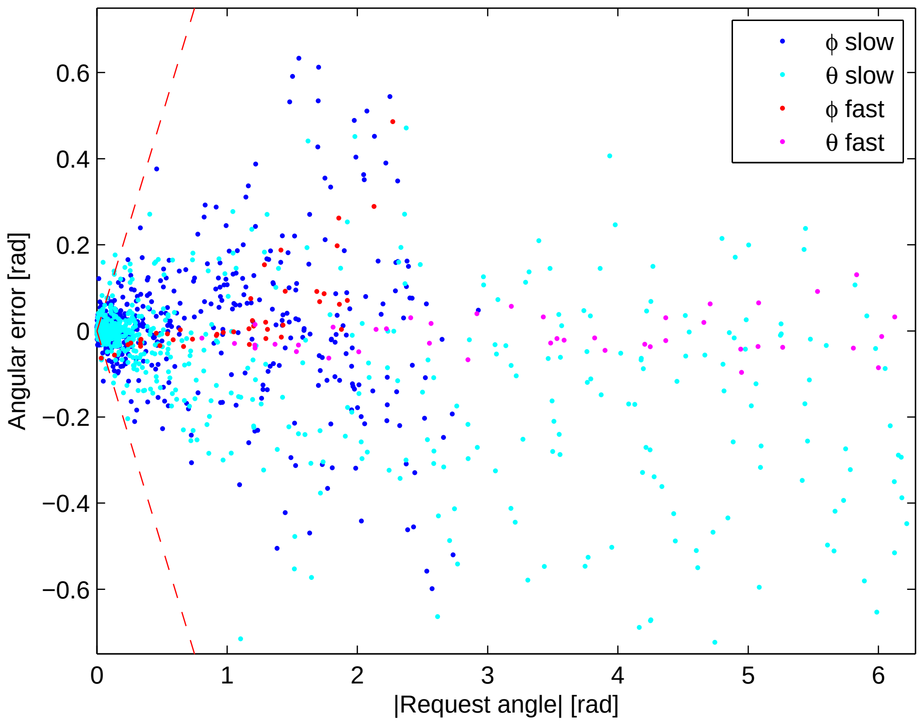

Figure 1: JPL Cobra movement data (angular error vs. request angle). In this figure, whether the request angle is positive or negative (absolute values are plotted), a postive angular error represents an overshoot of the target position, while a negative value represents an undershoot. The θ motor is far more accurate with fast mode than slow mode for large angular requests, while the ϕ motor is not significantly better. The dashed red lines are slope 1 and -1 references — points falling above the slope 1 line or below the slope -1 line represent moves where the angular error is larger than the angular request.

[1]This is programming laziness on my part – the real system should adjust speeds based on individual request angles.

3.2 Markov process model for Cobra movements

For any specific starting coordinate, there is not a lot of data with which to determine the outcome distribution. Until more data becomes available, I am using the existing slow-move data to generate a Markov transistion matrix which is used to propagate Cobras to the target locations. The θ or ϕ state of a Cobra is defined by the difference between its present angles and the angles needed to put it on-target ($\Delta\theta$ and $\Delta\phi$. The state of the bench is a histogram of these angular differences, which I write as a column-vector $\mathbf{v}$, where each element of $\mathbf{v}$ corresponds to one bin of the histogram. Presently, I use mostly[2] logarithmically-spaced bins for the histogram: 0–1, 1–2, 2–4, 4–8, ... mRad.

In a Markov process, the transition matrix $\mathbf{M}$ is used to calculate the probability distribution of a future state vector, given the current state vector. Each matrix element of the transition matrix, $M_{i,j}$, gives the probability that a Cobra in the j^{th} angular bin of the state vector will move into the i^{th} angular bin:

$$M_{i,j} = P(i|j). \tag{3}$$

Our initial probability distribution $\mathbf{p}_0$ is the norm of the angular distribution ($\mathbf{v}_0/|\mathbf{v}_0|$). The probability distribution after one Cobra move iteration is:

$$\mathbf{p}_1 = \mathbf{M}\mathbf{p}_0, \tag{4}$$

and the probability distribution after the n^{th} iteration is

$$\mathbf{p}_n = \mathbf{M}^n\mathbf{p}_0. \tag{5}$$

Note that $\mathbf{p}_n$ is the probability distribution for a single fiber. If we have N motors and want to know the probability that at least K of them are in some angular range, the expression is

$$P_{\geq K} = \sum_{j=K}^{N} \binom{N}{j} P_1^j (1 - P_1)^{N-j}, \tag{6}$$

where P_1 is the probability for single independent fiber to be in the specified angular range and $\binom{N}{j}$ are binomial coefficients.

In practice, the elements of the transition matrix are determined by collecting all Cobra moves with request angles in the j^{th} angular bin and using the corresponding angular error distribution to calculate the $P(i|j)$'s, one

[2] All but the first bin.

column at a time. For example, if we have 100 samples that start in the 8–16 mRad range and 25 of them move into the 2–4 mRad range, then the matrix element $P(2-4\text{mRad}|4-8\text{mRad}) = 0.25$. For the existing data set, the number of raw data points per transition matrix column range from 49 to 148 with a mean of 87 for θ and 110 for ϕ.

The first column of the transition matrix is treated differently because it defines the converged Cobra state. It is defined as

$$\mathbf{M}_{i,1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \tag{7}$$

which is the equivalent of saying that when we are within 1 mrad of the target postion for a given angular degree of freedom, we will not try to move that motor any more.

As an example, consider the θ angle state for a 6-Cobra bench. Suppose that the angles needed to put the fibers on-target are $\Delta\theta = (0.75, 1.25, 1.75, 2.25, 2.75, 3.25)$ mrad. Using the angular ranges defined above, the initial state and probability distribution of the bench are

$$\mathbf{v}_0 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ \vdots \end{bmatrix} \quad \text{and} \quad \mathbf{p}_0 = \begin{bmatrix} 1/6 \\ 1/3 \\ 1/2 \\ 0 \\ \vdots \end{bmatrix}, \tag{8}$$

while the desired end-state of the bench is

$$\mathbf{v}_f = \begin{bmatrix} 6 \\ 0 \\ 0 \\ \vdots \end{bmatrix}. \tag{9}$$

Suppose that in the JPL data set, when the request angle is between 1 and 2 mRad (ie, in the second bin), there is equal probability for the angular error to end up in any of the first three bins, and when the request angle is between 2 and 4 mRad, there is equal probability for the angular error to end up in either of the first two bins. The transition matrix would then be

$$\mathbf{M} = \begin{bmatrix} 1 & 1/3 & 1/2 \\ 0 & 1/3 & 1/2 \\ 0 & 1/3 & 0 \end{bmatrix}. \tag{10}$$

Using Equation 5, we can calculate the (single independent Cobra) probability distribution for an arbitrary number of Cobra moves. The evolution of this probability distribution is plotted in Figure 2. The probability at least 5 cobras in the 0–1 mRad bin is, according to Equation 6:

$$P_{\geq 5} = 6P_1^5(1 - P_1) + P_1^6. \tag{11}$$

Using the Markov tranisiton matrix, we can run Monte Carlo simulations on a set of Cobra motors. The Monte Carlo simulations bear out the simplest sanity check of this method in that the predicted number of iterations to place 95% of the fibers on-target (8) matches the JPL results. These results represent a best-case scenario for PFI. Measurement errors in excess of the errors in the JPL setup will result in a greater number of iterations to convergence. In the next section, I will discuss how measurement errors are incorporated into the configuration simulation.

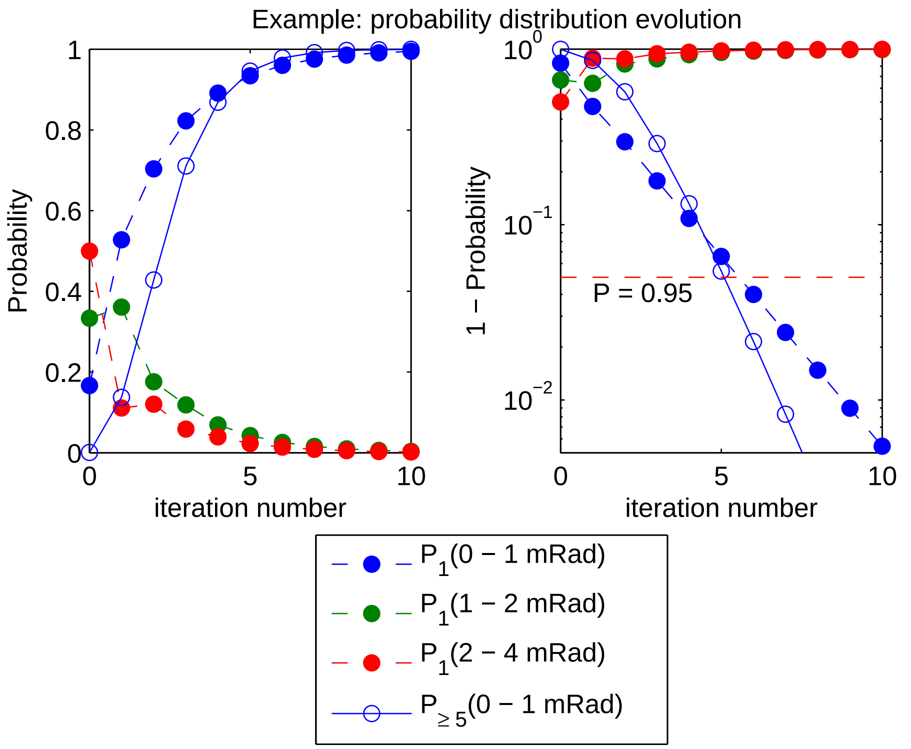

Figure 2: The left and right panels both show the $\Delta\theta$ probability distribution evolution of the hypothetical 6-Cobra bench. Details in the early evolution are easier to see in the linear P vs. n plot, while details near convergence are easier to see in the semilog $1-P$ vs. n plot. In the legend, the subscript on P indicates the number of independent Cobras for which the probability applies. After the 6th iteration, there is a 96% probability that the randomly chosen fiber that we are tracking is in the 0–1 mRad range. The probability that at least 5 fibers are in the 0–1 mRad range is shown as open blue markers; this probability almost reaches 95% at the 5^{th} iteration and crosses 99% on the 7^{th} iteration.

4 Movement simluation and convergence

The last two parts of the configuration simulator — the incorporation of position determination error into the configuration simulator and consideration of convergence criteria — are treated together in one section because they both require a clear distinction between the true position of a science fiber and its "determined" position. I use "determined" rather than "measured" here to emphasize that this is the PFI coordinate of the fiber, which depends on the MCS measurement error, but also includes other terms. When we set up a move request or when we decide whether or not a fiber is on-target, we only have the determined position to work with. On the other hand, in simulation, we have to keep track of and move the true position of the fiber in accordance with the movement model.

The fully installed PFI system is expected to have position determination errors greater than the $\sim$1 μm-level errors seen in the JPL test stand. Position determination error is a function of:

Calibration error : The error in the calibrated positions of reference fibers on the PFI bench.

Hard-stop repeatability error : The positional repeatability of the Cobra motors at their home positions.

WFC distortion error : The difference between our model of the back-lit WFC distortion and the true distortion.

Dome seeing : Error imparted on the postion of fiber images on MCS due to turbulence in the dome.

Centroiding error : Error in the centroid estimation of the fiber image on MCS.

Discussion of the calculation of the position determination error is covered in a separate white paper, unsurprisingly titled, "Calculation of the position determination error." Here, I will only point out that, with the exception of the hard-stop repeatability error, all components of the position determination error are more natually represented in Cartesian or cylindrical coordinates rather than (θ, ϕ) coordinates. For now, I treat position determination error as a 2-D Gaussian distributed error in Cartesian coordinates, even though it has one component that clearly does not follow that distribution. Future versions of the simulator will likely treat this error properly.

4.1 Movement simulation

Incorporating position determination error into the Cobra movement model requires going back and forth between Cartesian coordinates, where position determination error and allowable fiber placement error are well characterized, and (θ, ϕ) coordinates, where Cobra motions and angular errors are best understood.

To simulate a configuration step for a single fiber, we must keep track of its true position, while working with its determined position. The algorithm for moving a single fiber is shown schematically in Figure 3 and enumerated here:

1. Start with the current true position of the Cobra and the desired target position.

2. In Cartesian coordinates, randomly assign a determined position for the Cobra using the position determination error for the 1-σ width of the Gaussian distribution about the true position (see white paper on position determination error).

3. Calculate $\Delta\theta$, $\Delta\phi$, and the Cartesian distance between the determined fiber position and the target position. Continue only if the fiber is not on-target.

4. Roll the dice and consult the Markov transition matrix to determine where the *determined* position will move. In (θ, ϕ) coordinates, the *true* position will be transported parallel to the measured position.

5. The new true position is the starting point for the next iteration.

4.2 Convergence criteria

Presently, the simulator uses the loosest possible definition for convergence. I only require that the determined position of a fiber be within the allowable fiber placement error radius to consider the fiber on-target. Once it is on-target, it is never moved again. Depending on the position determination error at the current iteration, the fiber could, in fact, be well outside the allowable radius.

There are other possibilities for convergence criteria. Since many fibers are on-target early in the configuration, we could use the repeated measurements on these fibers to further refine their determined position. If, upon further inspection, we determine that the fiber is, in fact, just outside the

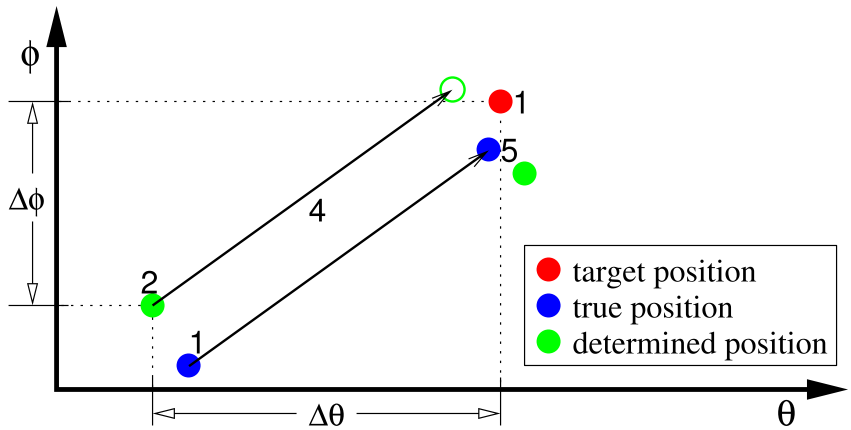

Figure 3: Schematic of a Cobra move iteration in PFI configuration. (1) The initial state of the system is defined by the initial true position (blue) and target position (red). (2) The initial determined position (green) is drawn from a 2D Gaussian distribution about the initial true position in Cartesian coordinates. (3) $\Delta\theta$ and $\Delta\phi$ are calculated between the initial determined position and the target postion. The Cartesian distance between the determined and target positions is also calculated at this time to evaluate whether or not the fiber is on-target. (4) Using the Markov transition matrix, calculate the actual move angles given the requested moves $(\Delta\theta, \Delta\phi)$. The open green circle represents the "virtual" new position of the determined position. (5) The new true position (blue) is defined by transporting the initial true position parallel to the determined position (initial to "virtual" new). The solid green dot hear the new true position represents the new determined position, evaluated in the next iteration.

allowable radius, we could re-activate the motor for that fiber. This would result in more accurate fiber placment, but would increase the number of iterations to convergence by a few steps.

Yet another possibility is changing the on-target metric from a simple distance-from-target criterion to a predicted fiber throughput. This would give us a better handle on performance vs. setup time tradeoffs.

5 Conclusion

The PFI configuration simulator is still a work-in-progress. I have laid out the methods used to simluate Cobra movements in the presence of the major error contributions to position determination: the Markov-process model for generating Cobra moves, and the algorithm for updating the true cobra position using the measurement and calibration determined fiber position.

This configuration simulator, despite not including all error terms, gives us a more realistic best-case-scenario for the configuration performance of PFI than a straight reading of the JPL test results. Given the current error budget allocations, instrument design, and operational plan, we configure PFI in no fewer steps than predicted by this simluator. Clearly, we need to modify PFI's fiber configuration plan. As we do so, this simulation will be useful in evaluating our choices in the design and implementation of PFI.