

MMT Encoder Output Modeling

January 28, 2010
D. Clark

Here are the results of some quick modeling of the expected output error from the Inductosyn encoder and the Heidenhain RCN829 encoder. Both pointing error for a random population of ~100 positions and tracking velocity estimator outputs are included.

Key Model Assumptions

The models are relatively unsophisticated since a lot of specific data on the RCN829 encoder specifications are as yet unknown. The model's design was influenced by some sample graphs of a different encoder's error curve, informed by the datasheet for the RCN829 that describes the unit's overall guaranteed error and reversal error.

Both models are run with a fixed-step solver at intervals of 1mS, which is believed to be reasonable due to the fact the system hardware runs at that rate so it captures the sampled-time behavior of the encoder system.

For the RCN829:

1. The absolute error curve is a "double peaked" sinusoid with an overall envelope of 1". The actual unit probably has a different shape and inflection points through zero error. The error curve modeled is a simple approximation of the catalog curve for a different encoder unit.
2. The reversal error, defined by Heidenhain as the error measured at 10 points around the total shaft rotation when moving to a specific position and then back to the starting point, is noted to be 0.4" maximum. In the model, this is modeled very simplistically as a 0.4" backlash, so there are discontinuous outputs when reversing the direction of shaft rotation; this is probably not the case but designing a smooth hysteretic model was judged too complicated for the moment. Heidenhain does note that the reversal error is dependent on the shaft coupling, so this is a critical point for proper mechanical design of the encoder mount.
3. The encoder quantization steps for the 29-bit encoder values are assumed monotonic and perfectly linear. This is certainly a simplification, but there is no data to suggest just what this value might be; we can for the purposes of simulation assume that this error is captured by the encoder error curvature.
4. It is not for the moment chosen to model the incremental-counting lines on the unit as that requires a significant amount of work to build a correct simulation model.

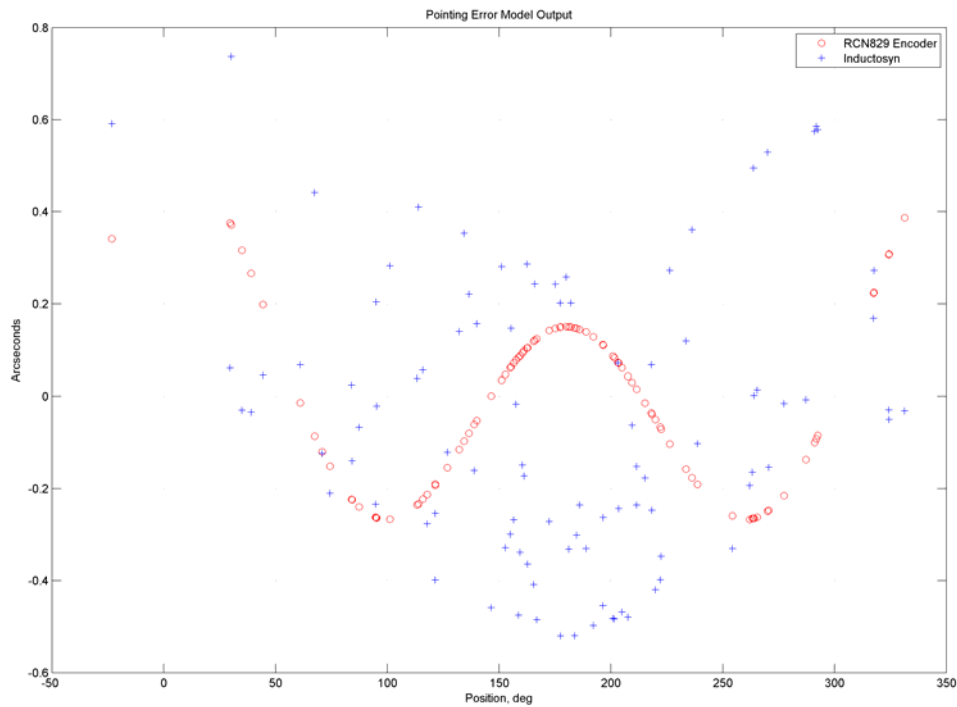
For the Inductosyn encoder:

1. The absolute error output curve is approximated at 0.25" peak for both the overall offset error, and the cyclical 512- and 1024-cycle/rev errors. No particular effort to scale or phase these errors was done, again in the interest of time. This error value could probably stand to be updated via a pointing run to check that it is correct, but this is the number to the best of my recollection.

2. The reversal error for the RCN829 is changed to reflect a 1-lsb bit width since the Inductosyn conversion electronics use a tracking loop to continuously calculate the shaft angle data; the minimum tracking reversal is then by definition 1-lsb wide. Inputs are chosen to remain well within the tracking loop's update rate.
3. The encoder output is quantized at 25-bit intervals, again assuming perfect monotonic values and linearity.

Model Outputs

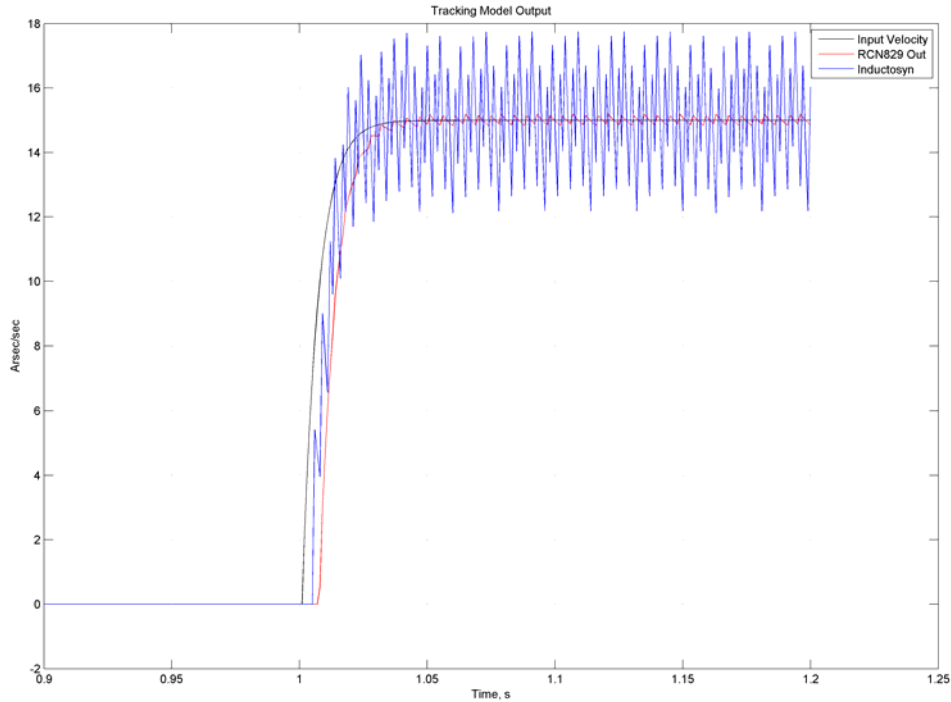
First, a random population of ~100 positions between 0 and 360 were created, then applied as constants to the simulation model. The resultant difference between the input shaft position and encoder output positions are plotted below:



The “double peak” sinusoid error approximation for the RCN829 is clearly visible; the Inductosyn looks more random, but is in fact cyclical on a 512- and 1024-cycles/rev basis.

The standard deviation of the error for each is **0.3”** for the Inductosyn and **0.18”** for the RCN829. For elevation, depending on the mounting angle for the unit chosen and its particular error curve the RCN829 pointing error could be significantly less. The above reflects more accurately the case when installed in the azimuth axis.

The next part of the simulation looks at velocity estimation for a constant tracking rate. The model inputs a position ramp starting at 1s with a $15''/s$ slope. The simulation model uses the discrete differentiator blocks from the actual elevation controller to calculate the velocity.



The input angle, since it has not undergone quantization is smooth and reflects the low-pass noise inherent in the velocity estimation filter. The larger quantization steps in the Inductosyn are clearly visible, and have quite a bit of variation. This is consistent with what we learned last week in the course of getting the elevation controller feedback from the absolute encoder. The much finer resolution of the RCN829 (about 2mas) results in a lot smoother velocity estimates.

Extending the RCN829 model to simulate $1/T$ counting might at this point be worthwhile to see how much improvement is in fact possible. However, the data here show that the velocity feedback with just the RCN829 positions and the existing estimator are quite acceptable; simulation at lower velocities ($\leq 2''/s$) would help flesh this out.