1 System identification of soft and squishy plants with delay, offset, drift and noise

Reference plants that contain offset, drift, delay and noise are proposed. In-system measured Step Response data is used to derive plant models. The modeled plants are simulated and compared to the reference plants. The plant model Impulse Response is compared to the reference Impulse Response.

The motivation for system identification is to create a mathematical plant model that represents the behavior of the plant. After determining a usable plant model, we can create controllers that have desirable properties, such as eliminating overshoot and oscillation, accelerating the performance of sluggish plants, shaping plant input, and shaping plant output. Other properties, such as delay, offset, drift, and noise are also managed. In-system measurements are used to assure that we have captured all system effects, known or unknown, understood or not understood. A plant model is also very useful for simulation. Determining the plant model, then, is the foundation for everything that follows.

First, a few definitions:

The “plant” is simply the thing we are attempting to control.

“Delay” means that the plant does not immediately respond to control effort.

“Softness” refers to how smooth and gentle the plant responds to a step input. Plants that that have sharp and crisp response to a step input are considered to be “hard,” whereas plants that have a slower, smoother, and gentle response to a step input are considered to be “soft.” We can see this effect directly in the sharpness and crispness of the leading edge of the plant step response.

“Squishy” plants can have elastic effects such as a lung or soft tubing that stretch and relax as pressures change.

“Offset” refers to the plant output being shifted up or down.

“Drift” refers to a shifting of the plant output in a sloped fashion, as would be seen as a drifting up or down as soft plants continue to stretch or relax in response to changes in control effort.

“Impulse Response” is terminology used in the context of continuous-time systems, whereas “Unit Response” is terminology used in the context of discrete-time systems. In every situation in which “Impulse Response” is used in the context of discussing discrete-time systems, it is intended to convey the meaning of “Unit Response.”

Noise is intentionally injected into the plant output to simulate real-world plants in which noise may be inherently part of the system, or part of the measurement of the system.

Various combinations of conditions are presented as reference plants which are modeled and compared to verify the quality of the model.

In the charts that follow, “Pout ref” is the “Plant output reference,” which is basically the reference plant we are attempting to model. “Pin” refers to “Plant input,” which is the control effort applied to the plant.

We start by defining a plant that has no delay, no drift, and no noise. After we have a look at this plant, we will add delay, drift, and noise.

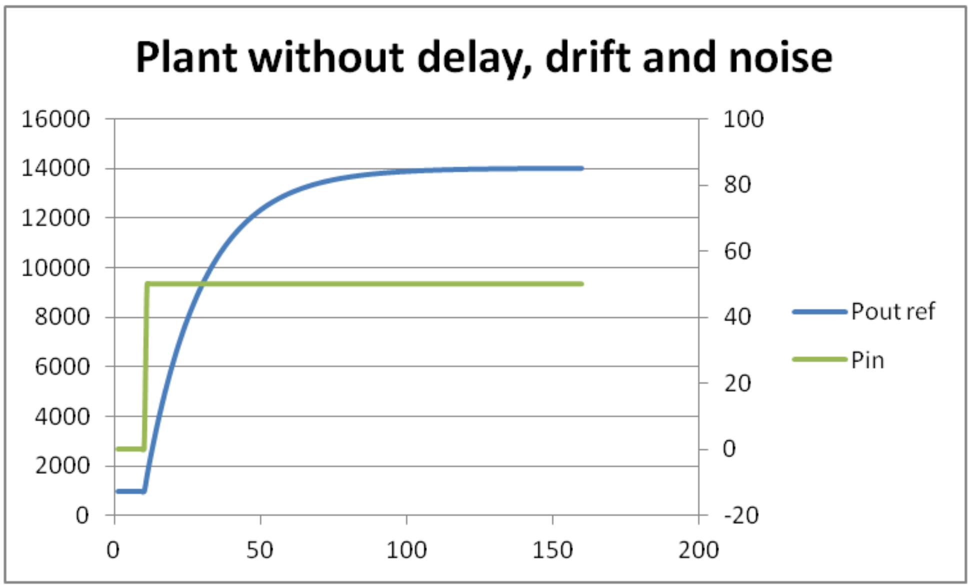

Figure 1

Figure 1 shows that there is no delay from control effort to the plant response. We can also see the sharp leading edge of the exponential rise of the plant response Pout.

Let's add delay and change the leading edge of the response to be softer.

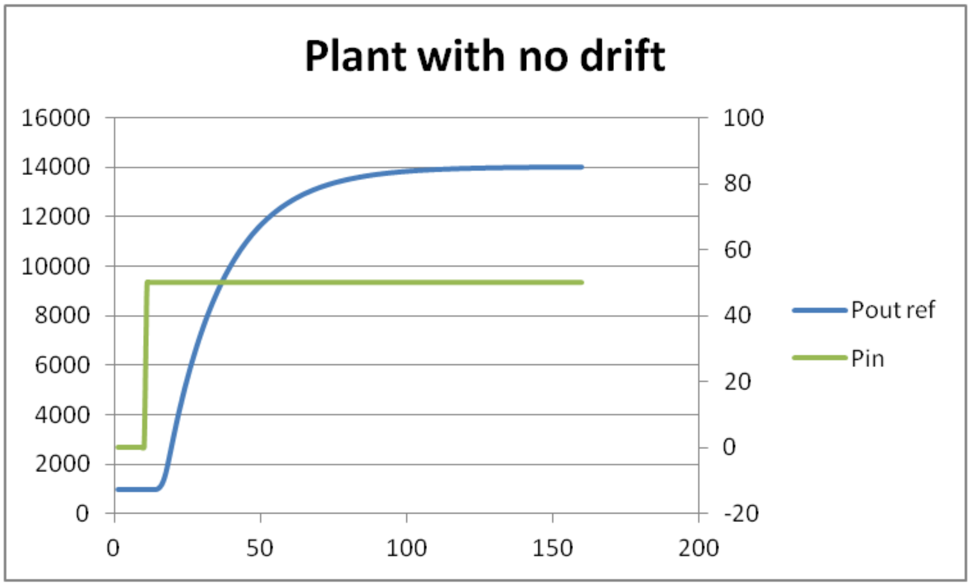

Figure 2

Figure 2 shows the delay from control effort to plant response. We can also see the soft corner of the leading edge of the plant response.

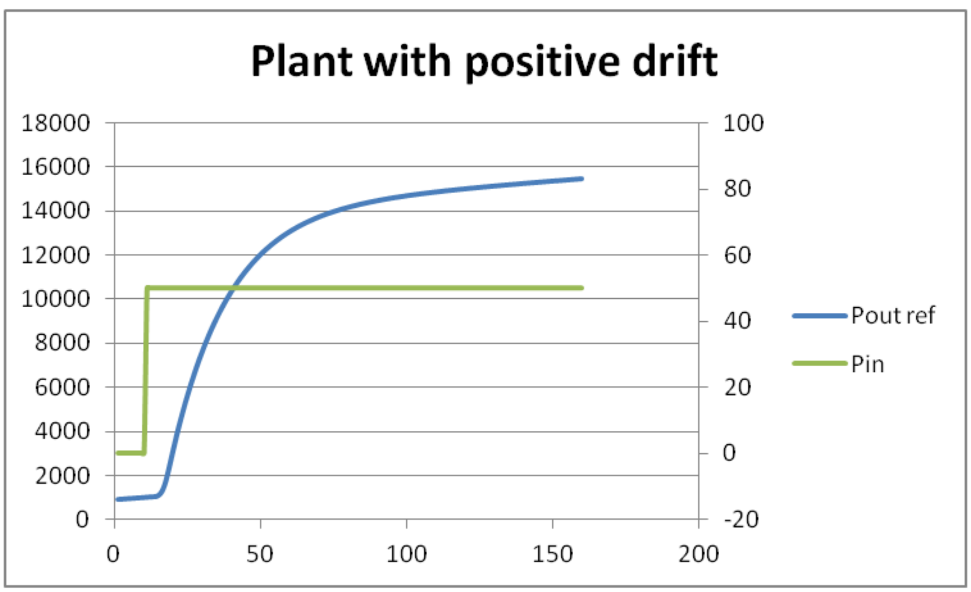

Figure 3

Figure 3 shows that the plant is drifting up.

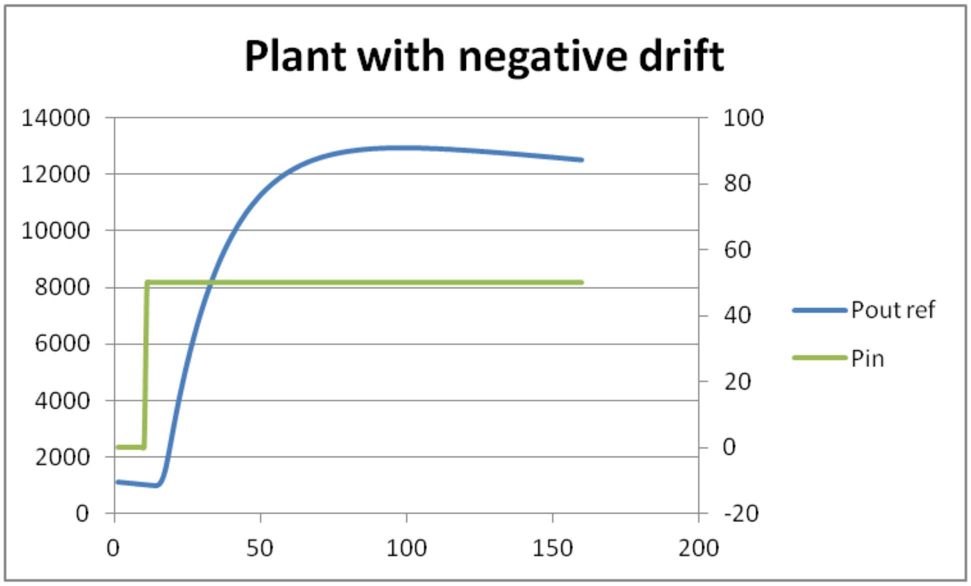

Figure 4

Figure 4 shows that the plant is drifting down. Let's add noise to the plant.

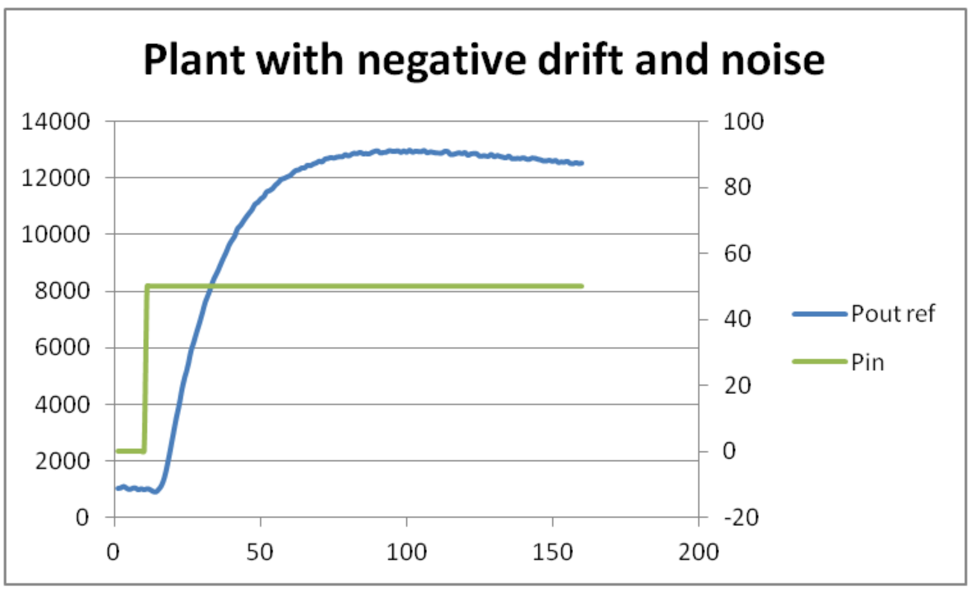

Figure 5

The chart in Figure 5 has offset, delay, drift and noise, all conditions that may be encountered when identifying real plants.

Let's use a matrix solver to get a least-squares best fit for the model parameters. The chart in figure 6 shows the plant with the model overlaid. There really are two lines here, but they overlay almost perfectly, even in the magnified views of Figures 7 and 8.

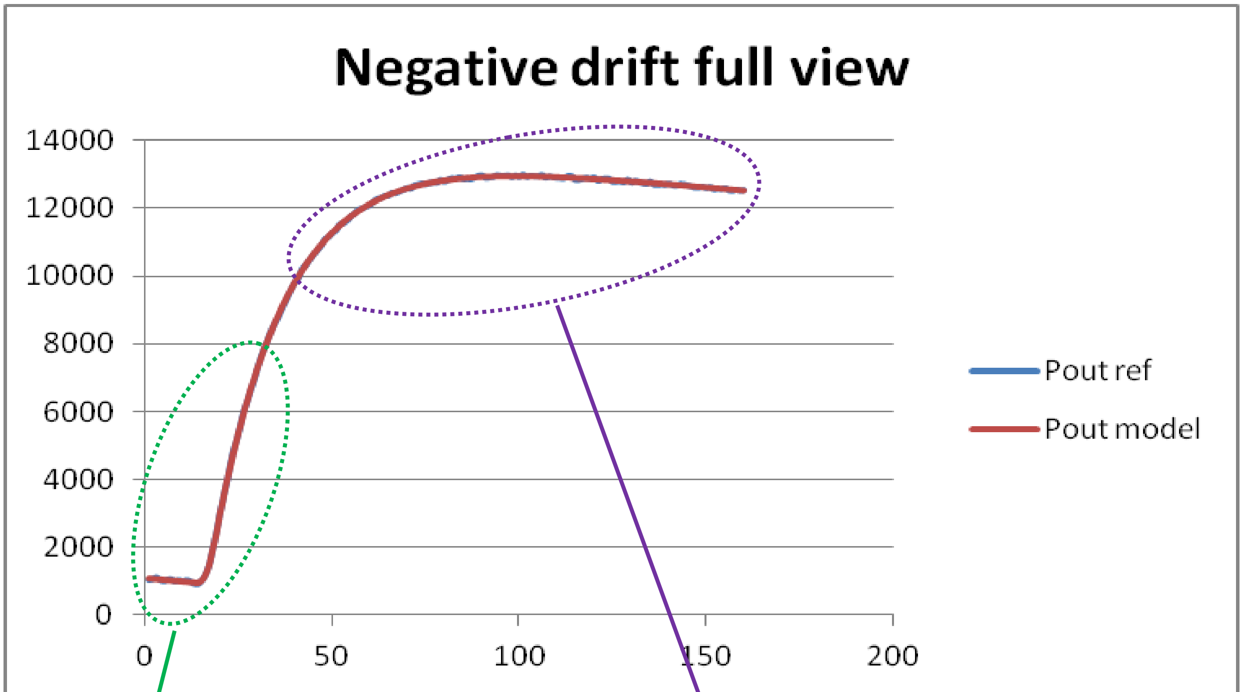

Figure 6

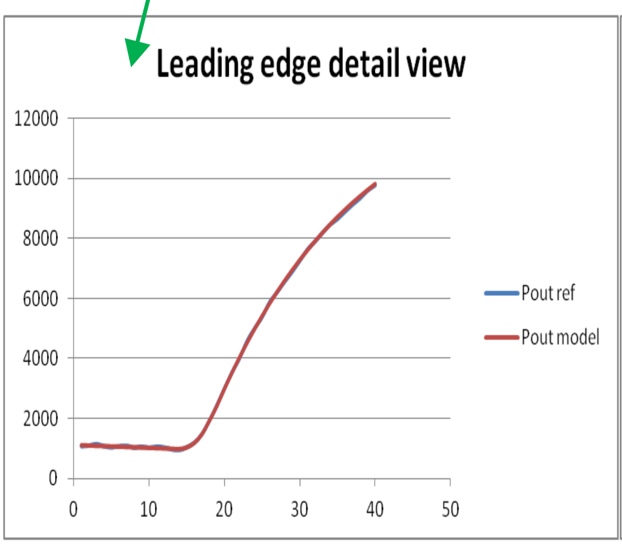

Figure 7

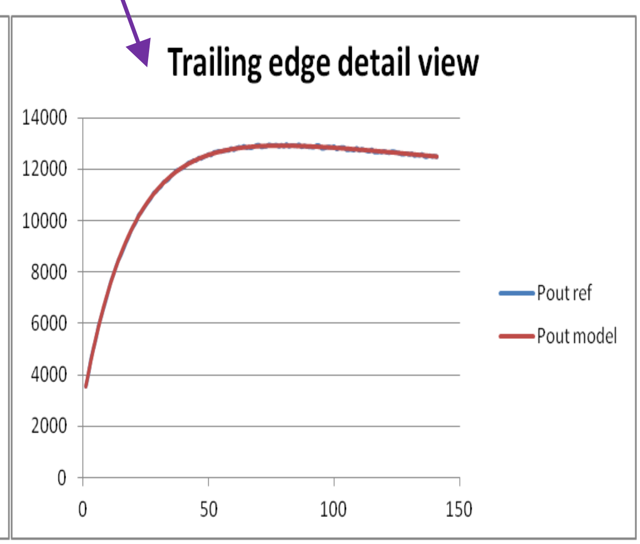

Figure 8

For completeness, we also construct models for the positive drift (Figure 9) and no drift (Figure 10) cases. As before, the lines overlay almost perfectly.

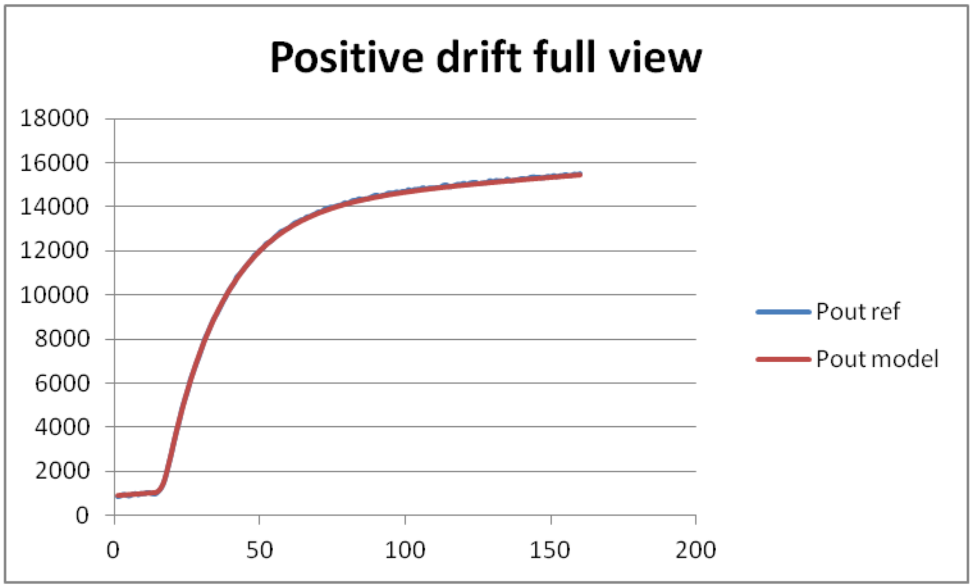

Figure 9

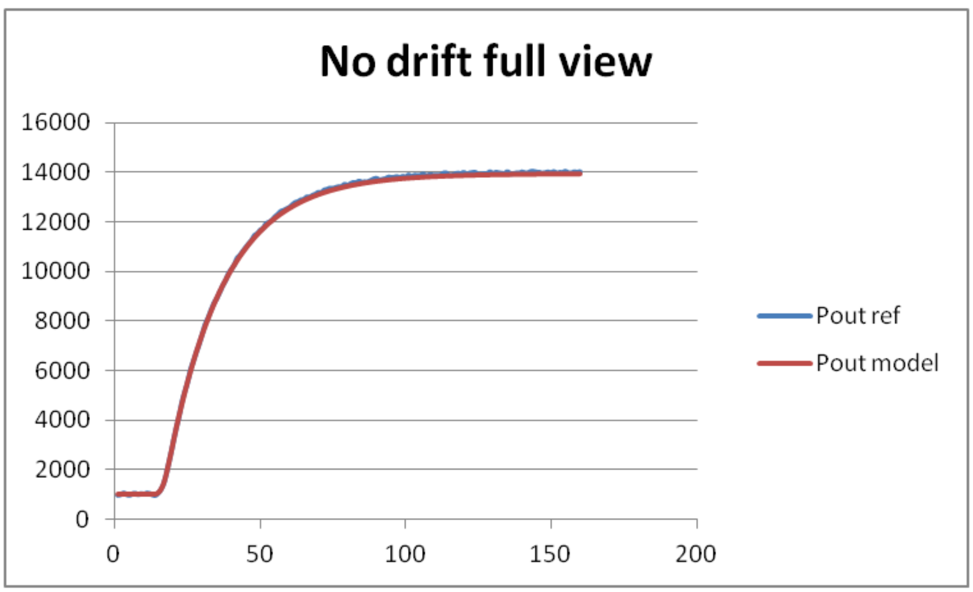

Figure 10

And finally, in Figure 11, the Impulse Response for the plant model is compared to the reference Impulse Response for the no-drift case.

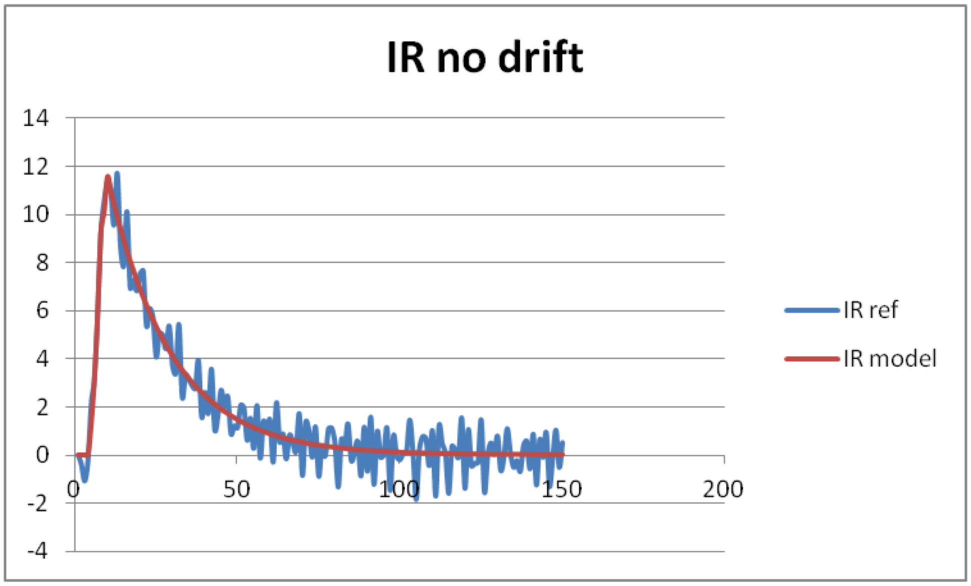

Figure 11

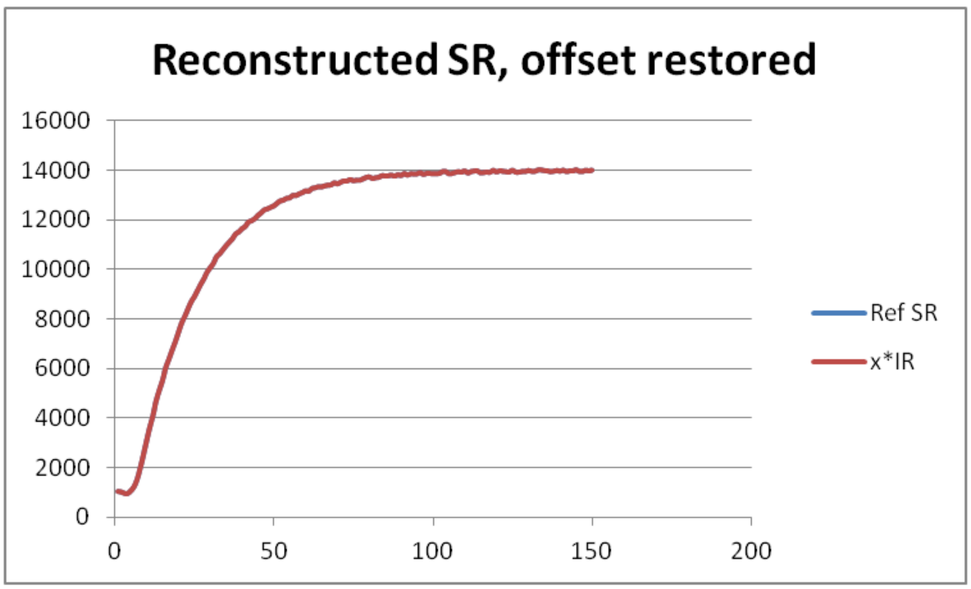

Figure 12

The chart in Figure 12 shows that the reference Step Response can be reconstructed from the noisy reference Impulse Response (with offset restored) to get the original reference (in-system measured) Step Response, warts and all. The chart shows that the reconstructed Step Response *exactly* overlays the original in-system measured Step Response data. This is not a particularly useful result, except that it demonstrates that the inverse operations get us back to *exactly* what we started with, including noise and offset.

Summary:

The computed model reasonably represents the reference plants that we are identifying and shows that plants with these characteristics can be identified reasonably well in the presence of delay, offset, drift, and noise.

Models derived using least-squares techniques reduce noise significantly. This is especially pronounced in the derivation of the reference and model Impulse Responses, in which we see a dramatic improvement.

Accurate plant models enable us to do a better job of creating controllers that match the real physical system. This will be explored further in future papers.

Accurate plant models are very useful for simulation.

The Impulse Response for the no-drift plant model is shown and compared to the reference Impulse Response. Dramatic noise-reduction is achieved.

It is important to point out that the computed model is not derived from analytical input, but rather from data collected during in-system operation. This has the advantage of identifying a system that may contain effects that are not known or not understood. We proceed by collecting the data during in-system operation, which includes all effects, known or unknown, understood or not understood, and then computing a model that best represents that system.