PSIG 0911

State Estimation in Online Models

Jason P. Modisette, ATMOS International

This paper was prepared for presentation at the PSIG Annual Meeting held in Galveston, Texas, 12 May – 15 May 2009.

ABSTRACT

Online models of pipelines almost always suffer from a surfeit of data: while a hydraulic model only requires one pressure or flow measurement at each boundary, real pipelines have pressure and flow meters all over the place. One would ideally like to make the best possible use of all the measurements. This process is called "state estimation".

Over the last thirty years, several schemes have been developed for state estimation. Two of the more popular ones, a simple pressure-based scheme that discards many of the measurements and the Equal Error Fractions approach, will be examined here and compared with a new maximum-likelihood scheme and, for reference, with no state estimation at all.

Comparisons will be performed on simple simulated gas and liquid pipelines in the presence of meter noise, quantization errors, and hydraulic transients. The performance of the various approaches will be examined, as well as the complexity of implementation and run speed.

INTRODUCTION

A hydraulic model of a pipeline requires that one boundary condition, either pressure or flow, be specified at every place fluid enters or leaves the line. Pipelines tend to have much more instrumentation than this; for example, there are often both pressure and flow measurements available at supply and delivery locations. The extra data from these additional instruments can improve the estimate of the pressures and flows in the system (that is, the hydraulic state of the line). Techniques for doing this are called "state estimation".

There is no industry standard technique for state estimation. While hydraulic models themselves now largely belong to two schools - method-of-characteristics models and implicit finite difference models - there are almost as many types of state estimation around as there are modelers. The goal of this paper is to describe and compare some of the more popular state estimation techniques with an eye to providing recommendations for which schemes to use in which circumstances.

There are several desirable properties of a state estimator; it's not enough to say "it should produce the best possible estimate of the state", because a method that works well in steady state might do something spectacularly wrong during a transient, or a method that works for certain types of instrument errors might have trouble with other sorts. Also, it's nice if a state estimator produces some form of output that can drive hydraulic tuning.

Methods of State Estimation

We will examine three approaches to state estimation: two that are widely used in online models and a new approach developed by the author. In this section we discuss the qualitative advantages and disadvantages of the different methods, and then in the Results section below we will compare their performance – how accurate an estimate of the state they can produce in the presence of various sorts of errors.

Pressure-Pressure Method

Assuming one already has a functioning offline model, the most easily-implemented way to perform state estimation is to run that using some subset of the available measurements as boundary conditions, and then after each model step is complete, somehow integrate the remaining measurements into an estimate of the actual state. This has the benefit that the actual model code doesn't need to be modified at all from its offline form; the additional measurements are included after the fact.

If the model is run with pressure boundary conditions, then the output of the model will include flow rates at the two ends of the pipe. These can then be compared with the flow meter readings; if the flow meters are reading higher than the model, this suggests that the actual flow rate is higher than the modeled flow rate, and vice versa. Given estimates of the errors in the flow meter readings and the errors in the modeled flows, we can then come up with an overall best guess of the flow in the system. At its most straightforward (as implemented here) this can be done by solving for a "consensus flow" at every location i that minimizes the weighted sum

$$\sum_i \left(\frac{Q_i^{consensus} - Q_i^{\mathrm{mod}el}}{\sigma_i^{\mathrm{mod}el}}\right)^2 + \sum_i \left(\frac{Q_i^{consensus} - Q_i^{measured}}{\sigma_i^{measured}}\right)^2$$

Minimizing this sum with respect to the vector of $Q_i^{consensus}$ is a system of linear equations which can be solved exactly on each time step. This consensus flow can then be taken to be the "real" flow. In a network, we can apply some additional knowledge here: for instance, that at a split the consensus flows must add up to zero. This sort of thing can be included in the above optimization problem using the method of Lagrange Multipliers, which leaves the problem still exactly solvable. In this article we won't be examining any scenarios with flow splits, so it won't be necessary to worry about this.

Note that one could also implement this method using flow boundary conditions, but since a steady state model can't be solved purely on flows that would require some special treatment for steady state. In addition, if the flow meters don't add up to a net inflow of zero, the model will eventually either drain or blow up the pipe unless some preprocessing is applied to force them to sum to zero. Thus the author has not encountered this approach except using pressure boundary conditions.

The pressure-pressure method is simple but has some potential problems: if there's a small mid-line delivery ("small" compared to the mainline flow rate), the hydraulic model will be much more stable if a flow boundary condition is used there: errors in the flow will cause smaller errors in the pressure, but errors in the pressure in the event a pressure boundary is used will cause large errors in the flow. However, if the model is run with a flow boundary at some points, there's no direct way to use any pressure data that may be available there in this scheme.

In the case of a pipeline (or section) operating at a small fraction of capacity, the errors in the pressure meters can be significant compared to the total pressure drop between meters, meaning that the flows calculated by a pressure-pressure model may be complete nonsense.

Yet another problem with this method is that it requires the user to come up with a relative weighting of the accuracy of the flow meters as compared to the accuracy of the flows computed by the model driven by pressure meters; this latter term has to include both the errors arising from the pressure meters themselves and those arising from unknown or incorrectly known characteristics of the pipeline system. This can be quite difficult to estimate; it the author's experience it is often estimated by guesswork.

The "consensus flows" calculated by this method may exhibit some strange properties if the flow meters are out of calibration, that is, if for instance the flow meter at the origin is reading high but the terminus flow meter is accurate. In that case the consensus flows can show a continuous packing of the line that goes on forever. This can be a problem because in a complex pipe network that has been split up into many pressure-meter-bounded regions, the actual model solution flows are not going to be conserved at mid-line pressure meters, such as those at metered pump or compressor stations. If the model is being used to drive a batch tracker, it will generally be more appropriate to use the consensus flows for that since they can be generated in a manner that ensures they are conserved where they should be; however, in that case something will have to be done about this possible eternal unpacking. So even though the pressure-pressure method is initially easy to implement, it can lead to some additional complexities.

Finally, it is not aesthetically pleasing to the author (or, in the author's experience, to the users of the software) that the flow rates computed by this approach do not correspond to the computed pressure drops unless the system is perfectly tuned up.

Simple hydraulic tuning under the pressure-pressure method can be done by examining the differences between the consensus flows and the modeled flows in the pipes; if the consensus flows are higher than the modeled flows the pipe's resistance to flow can be reduced a bit, and vice versa. This would correct for a steady state error in the pipe's resistance if that error were constant across all flows; or instead of the resistance, the pipe roughness or diameter or some other property could be adjusted.

Equal Error Fractions (EEF) Method

The Equal Error Fractions (EEF) method has a long history with the PSIG, having first been described in [1] at the 1987 conference. Unlike the P-P method, it allows co-located pressure and flow measurements to be used on the same footing.

If a pressure and flow meter are located in the same spot, then there is only one possible set of values for the modeled pressure and flow rate such that the modeled pressure disagrees with the measured pressure exactly as much as the modeled flow rate disagrees with the measured flow rate. It's compelling from a physical standpoint to only treat meters in the same spot on the same footing (as opposed to treating meters all over the pipeline on the same footing, as is done in the maximum likelihood approach) because the meters at the two ends of a pipe really aren't interchangeable: if a very accurate pressure and flow meter are located at the upstream end and a very inaccurate pressure and flow meter are located at the downstream end, the model still requires a downstream boundary condition, so the less accurate meters must be used.

In the case of a single pipe with pressure and flow meters at either end, the EEF approach has the boundary conditions:

$$\frac{P_i^{\mathrm{mod}\,eled} - P_i^{measured}}{\sigma_i^P} = \frac{Q_i^{\mathrm{mod}\,eled} - Q_i^{measured}}{\sigma_i^Q}$$

at the upstream end and

$$\frac{P_i^{\mathrm{mod}eled} - P_i^{measured}}{\sigma_i^P} = -\frac{Q_i^{\mathrm{mod}eled} - Q_i^{measured}}{\sigma_i^Q}$$

at the downstream end. Here Ps represent pressures, Qs represent flow rates, and σs represent the magnitudes of errors in the meters (specifically, the standard deviation in the case that the errors are due to normally distributed noise).

Qualitatively, EEF is somewhat more complicated to implement than the PP method but really it just requires the replacing of pressure or flow boundary conditions in the model with the special EEF boundary condition above.

However, for more complicated pipelines and configurations of meters, EEF requires that some different boundary conditions be derived in a customized manner: a mid-line delivery, a mid-line compressor or pump station with suction and discharge pressure metering and flow metering, etc., are all treated differently.

Hydraulic tuning can be performed under the EEF method by defining regions of pipes bounded by pressure and flow meters and evenly distributing a field of pressures and flows in some manner that agrees as well as possible with the measurements; the differences in each pipe between these pressures and flows and the model solution can then be used to drive tuning.

Maximum Likelihood Method

The maximum-likelihood method is a way of treating all of the pressure and flow measurements in the system on the same footing, while still enforcing the physics of the model. This method has the nice property that one set of equations will work for any pipeline configuration – there are no custom cases required for special configurations of flow or pressure meters like there are in EEF. The maximum-likelihood method produces the most likely estimate of the state of the pipeline under the assumptions that (1) the errors in the measurements are all normally distributed, (2) the underlying pipe model captures the physics correctly, and (3) the errors in the measurements are not correlated between time steps.

In this approach, the normal model equations other than the boundary conditions are treated as constraints on the minimization of the quantity

$$\sum_i \left(\frac{P_i^{measured} - P_i^{modeled}}{\sigma_i^P}\right)^2 + \sum_i \left(\frac{Q_i^{measured} - Q_i^{modeled}}{\sigma_i^Q}\right)^2$$

using the method of Lagrange multipliers. So if there are N model equations which can be written as $f_i(\{P,Q\}) = 0$, then the full minimized function becomes

$$\sum_i \left(\frac{P_i^{measured} - P_i^{modeled}}{\sigma_i^P}\right)^2 + \sum_i \left(\frac{Q_i^{measured} - Q_i^{modeled}}{\sigma_i^Q}\right)^2 + \sum_i \lambda_i f_i(\{P,Q\})$$

which is minimized with respect to the Ps, Qs, and λs. Note that if the original model equations are linear in the Ps and Qs (or, as is the case with our implicit model, linearized in the Ps and Qs) then this function is quadratic in all the variables, and so can be minimized exactly by solving a linear system.

Unlike in the pressure-pressure method, having flow meter drift does not present any problems for the maximum-likelihood approach. Although initially the model may pack or unpack the line a bit trying to honor the miscorrelated meters, it will soon reach a point where any further unpacking (for example) will cause the pressures to drift too far from the measured pressures, and so it will end up accepting an error in the flow meters in order to keep the linepack at a reasonable value. On the other hand, this method does not gain any benefit in its state estimation from the known fact that over a long period the net flow in or out of the line has to be, on average, zero.

Maximum-likelihood results can be used to drive tuning in exactly the same manner as the EEF results can, as was described in the previous section.

All of the methods presented here exhibit a common weakness: they treat each model solution time step in isolation from all the other steps. This is a problem because it throws away some information: how the errors the model is seeing are correlated from step to step. This information might be useful in tuning, specifically in distinguishing one type of error (say, a pressure meter that has drifted) from another (say, a gas pipe which has some liquid holdup and thus higher resistance to flow than expected.)

TEST CASES

The data used in testing these state estimators is all synthetic data generated for made-up pipelines. This is done because only on a made-up line can we know what the "right answer" is, so that we can definitively say which methods of state estimation do a better job of finding it.

Thus the approach is to use a pipeline model to generate pressure and flow data for the scenarios of interest, then add normally-distributed and other errors to all measurements. Since all of the state estimation methods described above can be used with implicit finite difference models, we have used such a model formulation for both the data generation and the state estimation.

The Test Pipelines

We use two pipelines: a slingshot gas line with 1-minute sample intervals and a slingshot liquid like with 10-second sample intervals. Each pipeline has a pressure meter at either end and a flow meter at either end and no other instrumentation, as seen in Figure 1; temperature and composition are assumed to be constant everywhere (so the liquid line has no batches). The pipes are level (no elevation changes) and have no diameter or other property changes along the line. The pipes are both 18" inner diameter and 40 miles long, and both start with upstream and downstream pressures of 1000 psia and 500 psia. The pipe roughness is 0.0018" and the fluid temperature is 60 F. The gas line is assumed to be moving 70% methane / 30% ethane and the liquid line is moving diesel with a specific gravity of 0.8. For simplicity of modeling the friction factors are calculated at upstream conditions and then assumed to be constant.

We examine the behavior of each state estimation technique both during a single scenario containing both steady state and violently transient periods. The transient changes are done by changing the upstream pressure setpoint while holding the downstream pressure fixed; the schedule of setpoint changes is shown in Figure 2. In a real system these would represent changes in the discharge setpoint on an upstream compressor or pump station. The downstream pressure is held constant - none of the state estimation techniques being examined here have any "preference" for the upstream or downstream ends, so it is not necessary to vary both the upstream and downstream behavior. Figure 2 shows the profile of pressure vs. time at the upstream end of the pipeline used for the transient tests.

The Model

All of the tests were performed on a simple implicit finite difference model which assumed a constant friction factor, isothermality, and a simple equation of state: ideal gas in the gas case and constant-isothermal-bulk-modulus in the liquid case.

The only equations in the model other than the boundary conditions are conservation of mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0$$

and conservation of momentum (in the absence of slope):

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho v^2)}{\partial x} + \frac{\partial P}{\partial x} + \frac{f\rho v|v|}{2D} = 0$$

where

ρ = density, a function of pressure and temperature (which is 60 F)
P = pressure
v = velocity
f = friction factor (treated here as a constant evaluated at typical Re and relative roughness for the system)
D = inner diameter

These equations are written assuming that these quantities are in a consistent set of units, such as SI. These are coupled to an equation of state, which is the ideal gas law for the gas line:

$$\rho(P,T) = \frac{PM}{RT}$$

where

R = gas constant
M = molecular weight of the gas
T = the temperature of the gas, 60 F

and an equation assuming constant isothermal bulk modulus for the liquid line

$$\rho(P,T) = \rho_0 \exp(\frac{P - P_o}{K})$$

with

ρ_o = density at 14.7 psia and 60 F
P_o = 14.7 psia
K = the isothermal bulk modulus

When written as finite difference equations in implicit form these two conservation laws become

$$\frac{\rho_j^{i+1} - \rho_j^i}{\Delta t} + \frac{\rho_{j+1}^{i+1} v_{j+1}^{i+1} - \rho_{j-1}^{i+1} v_{j-1}^{i+1}}{2\Delta x} = 0$$

and

$$\frac{\rho_j^{i+1} v_j^{i+1} - \rho_j^i v_j^i}{\Delta t} + \frac{\rho_{j+1}^{i+1}\left(v_{j+1}^{i+1}\right)^2 - \rho_{j-1}^{i+1}\left(v_{j-1}^{i+1}\right)^2}{2\Delta x} + \frac{P_{j+1}^{i+1} - P_{j-1}^{i+1}}{2\Delta x} + \frac{f\rho_j^{i+1} v_j^{i+1}\left|v_j^{i+1}\right|}{2D} = 0$$

These equations are only accurate to first order in time step or knot spacing so small knots have been used; we found that the model running by itself converged to within better than 1 psi in the transient scenarios described above with knot spacings of 0.5 mile in both gas and liquid, so this knot spacing has been used for all tests.

A 60-second timestep was used for the gas model and a 10-second timestep for the liquid model, representing typical SCADA scan rates for these systems. The theoretical basis of the two more sophisticated techniques below, the Equal Error Fractions and Maximum Likelihood approaches, requires that the noise on successive model steps be uncorrelated, so they would need to be adjusted to work with multiple model timesteps per SCADA scan. In this article we are avoiding that difficulty by letting the model timestep and the scan rate be the same.

Error Sources

State estimation helps correct for errors in the physical model of the system. These errors can come from many sources. In this article the sources that are considered are instrument noise and instrument quantization error (due to a limited number of bits being available in the A-to-D converter).

Instrument noise is assumed to be normally distributed and uncorrelated from step to step. Instrument roundoff error is meant to represent the limited precision of each instrument available through the SCADA system, and is represented in this work by artificially reducing the precision of the actual measurements. We have taken care that the precision does not artificially remove the noise (as might happen if the true value were 1000 psi, the roundoff was to the nearest 10 psi, and the instrument noise happened to add 3.5 psi.)

Instrument drift is not considered, as it is not something that state estimation can remedy – it falls more in the realm of tuning, to the extent that it can be handled in software at all.

RESULTS

In order to test the various state estimation methods, we first generated a schedule of upstream pressure changes which were to occur while holding either the downstream pressure fixed (shown in Figure 3). These included some simulated pump or compressor starts and stops and a brief shutdown of the line. We then ran a model (a fully implicit model with pressure-pressure boundary conditions) to simulate that schedule of changes. The upstream and downstream pressure and flow rate measurements computed by that model were then modified by adding normally-distributed noise and possibly other effects (emulating problems with real meters) such as quantization errors. These noisy measurements were then used to drive a second model run using a state estimation technique. Finally, the results of the second noisy model run were compared with the true results from the first exact model run and the RMS and maximum pressure and flow rate deviations were computed. The RMS averages were calculated over the entire length of the pipeline and the entire length of the run. Each state estimation run was performed 10 times with 10 different sets of noise; this allowed us to estimate the average errors in the results and their standard deviations.

This test procedure is illustrated schematically in Figure 2.

Note that the PP, EEF, and ML methods all require estimates of the accuracy of meters. In these tests, the actual standard deviation of the generated noise was used, and for the quantization error tests the quantization was used; in the real world, it might not be so easy to estimate these accuracies.

For comparison, a model with no state estimation was also run, using upstream pressure and downstream flow boundary conditions. The boundary conditions were taken to be the noisy meter readings generated from the original model run.

Ease of Implementation

The three methods were reimplemented with an implicit hydraulic model for this work. The sizes of the implementations (which are in C and which were all done by the author) are shown in Table 1. For a qualitative idea of the relative complexity of the methods, Table 1 shows their sizes in additional lines of code beyond that of the basic model as well as how long each method took to run the transient scenario. The EEF and PP methods did not take measurably longer than the base model; the ML method took about four times as long. The ML method requires the solution of a system of about twice as many linear equations than the other methods, and since sparse matrix techniques were not used here, this resulted in the matrix inversion taking considerably longer. The author's experience has been that with a sparse solver in real pipeline systems, the model doesn't spend much of its time in the solution of the linear system, and so the ML method does not take noticeably longer than the other methods.

Note that only the bare minimum of features necessary to produce results shown in the next section were implemented in each model, so it might be easier to add e.g. automatic roughness tuning to the code for some approaches than to others.

The knot size necessary to converge the steady-state results to within 1 psi for was 0.5 miles for both gas and liquid. This knot spacing is used for all results presented herein.

Performance

The performance below is measured by the average (RMS) and worst deviations of the modeled pressures and flows from their true values at either end of the pipe. The worst deviations are of interest because they are the sort of thing that might cause false alarms in a leak detection application.

There is no clearly correct way to equate a certain pressure error with the same flow rate error. For this reason, the pressure noise magnitude was varied independently of the flow rate noise magnitude. The baseline pressure error was normally distributed noise with a standard deviation of 3 psi (about 0.7% of the steady-state pressure drop), and the baseline flow rate error had noise of 0.1 mmscfd for the gas pipeline (with a steady-state flow rate of 188 mmscfd) and 2 bph for the liquid pipeline (with a steady-state flow rate of 2200 bph). For each scenario, four plots are shown: the effects on pressure (left column) and flow rate (right column) of increasing pressure measurement noise with fixed flow rate noise (top row), and of increasing flow measurement noise with fixed pressure noise (bottom row).

First, results are examined for a gas pipeline (RMS errors in Figure 4 and worst errors in Figure 5) and then a liquid pipeline (RMS errors in Figure 6). Figure 7 shows a comparison of the effects of noise with the effects of a similar level of quantization error.

The author has tried to cover a range of noise from "significant" to "tremendously much". Very small levels of noise were not examined because it is unlikely that a model will be so accurate, given the other unknowns about the pipeline, that the behavior of the state estimation at levels of 1 psi or less is going to be meaningful. In other words: if the instruments are accurate to better than 1 psi and an analogous flow, it's unlikely that the limiting factor in model accuracy is going to be something that a good state estimation technique can help.

One immediate thing that is apparent is that for all of the gas applications, the EEF and ML methods perform much better than the PP method, and it in turn performs much better than no state estimation at all. For the smaller examined values of meter error (which are still fairly significant amounts of noise), the ML method performed better than EEF, especially in the presence of errors caused by flow meter noise (Figure 4, bottom row).

The RMS pressure error as a function of flow meter noise using the EEF method has interesting behavior at the lower noise values: it decreases as the noise increases. At first this seems counterintuitive, but the author believes it is correct for transient scenarios like the one used here: the EEF method (like the ML method) provides relative weights to the flow and pressure measurements that are dependent on the accuracy of the meters; in the real world, the accuracy comes from manufacturer specs or from observing the meter's repeatability. In these tests we have used the standard deviation of the noise. The behavior seen in Figure 4 suggests that perhaps using the error fraction as the thing to equalize in EEF is not optimal, at least during transients. Indeed, the effect of a sudden spike in flow (due to noise) is dependent not just on the magnitude of the spike but also on the SCADA scan length – a big flow spike over a short scan will cause more significant noise-induced hydraulic changes in the model than that same spike would over a long scan. There may be a scan-length-dependent effect on the relative effects of a given size of flow rate spike and a given size of pressure spike; that would explain this behavior.

The Worst errors (Figure 5) also show that ML and EEF methods are vastly better than the PP or no-state-estimation approaches, and at low noise values ML performs somewhat better than EEF.

Tests on a simulated liquid pipeline (Figure 6) showed that the EEF and ML methods performed similarly, although at low error values the EEF performed better as far as resistance to flow meter noise producing pressure errors (the lower left plot). Again, the PP method and the no-state-estimation methods performed far worse than the other two methods.

Finally the effects of quantization errors were compared with the effects of noise (Figure 7). They are seen to be comparable except when the flow rate quantization error reaches what is probably an unrealistically large value, at which point the quantization errors become significantly worse than the effects of noise. (The tests in Figure 7 were performed using the gas pipeline and the ML state estimator.) This can be understood because the ML state estimator assumes that the errors on successive scans are statistically independent, but if the source of the error is quantization then they will tend to be correlated.

CONCLUSIONS

One issue with these results is that it's likely that all of these state estimation techniques can be improved by tweaking. The implementations used here are the most basic possible implementations. But, for instance, the author has found that the performance of the P-P method can be improved by smoothing the pressure measurements with a low-pass filter if there is a large noise component.

The performance of the ML method is as good as the other methods in almost all circumstances and is usually better than any of the other methods; it appears to be the most accurate estimator of the state of the line during transients. However, the ML method is fairly complex to implement and cannot be strapped on to an existing model as easily as the other methods. The EEF method, for example, can be implemented in an existing model with only the addition of a new sort of boundary condition, at least for a slingshot pipeline as seen here.

These results clearly show the benefit of using some sort of state estimation, rather than just running a model using some of the meters as boundary conditions and ignoring the remainder.

REFERENCES

1. "Gas Network State Estimation with the Equal Error Fraction Method", Tom van der Hoeven, Proceedings of the PSIG Annual Conference, Oct 1987

ACKNOWLEDGEMENTS

The author would like to thank ATMOS International for support of this work and especially Einac van Meurs and Jun Zhang for their help.

TABLES

Method	Size of Code (lines of C)	Total Time of Run (seconds)
Underlying model	383	0.5
PP	12	0.5
EEF	20	0.5
ML	122	1.9

Table 1 – Caption

FIGURES

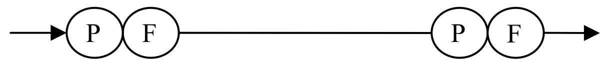

Figure 1 – Pipeline layout: pressure meter indicated by P, flow meter indicated by F

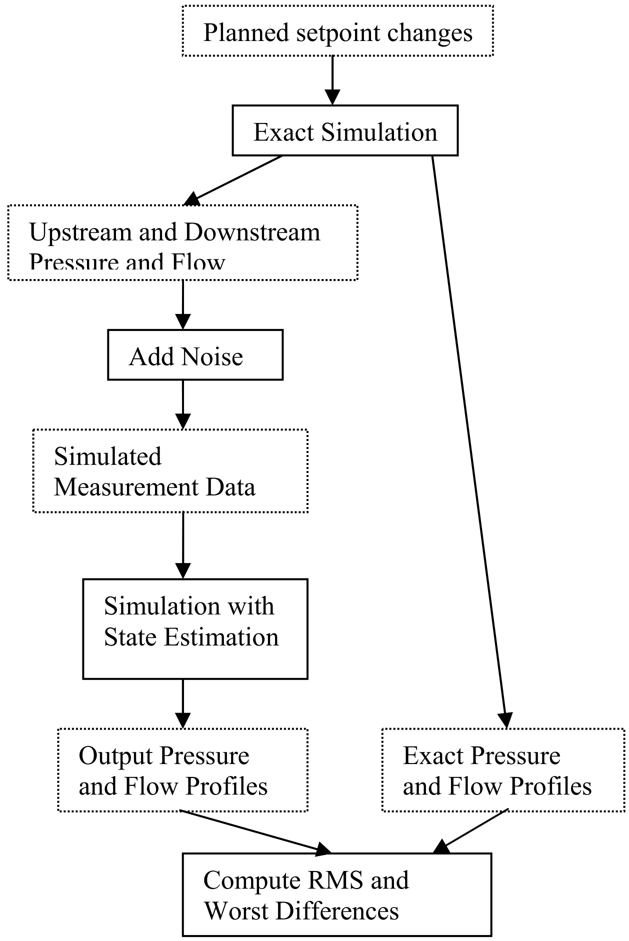

Figure 2 – Procedure for comparing the accuracy of state estimation techniques. The final RMS and worst differences are presented below

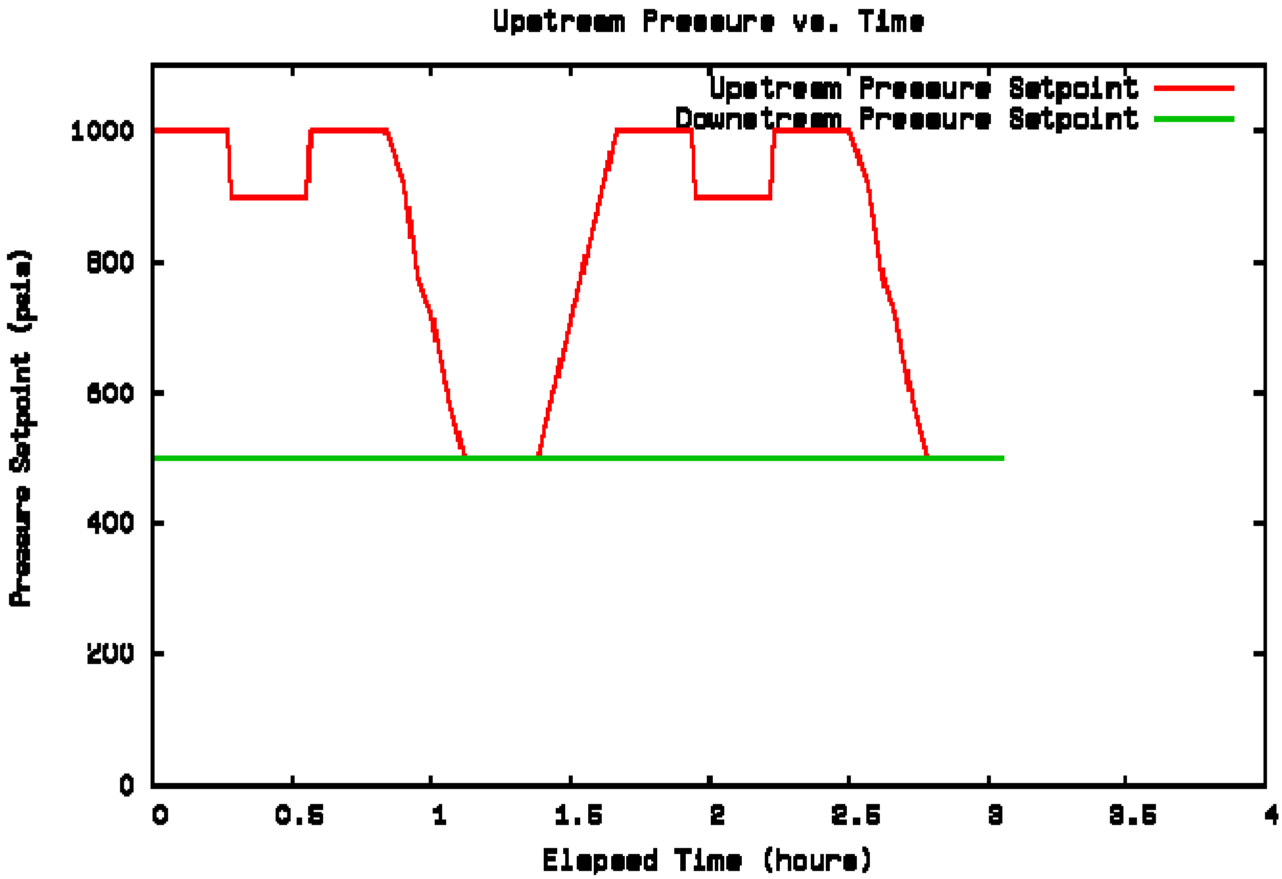

Figure 3: The upstream and downstream pressure setpoints used to drive both the gas and liquid models.

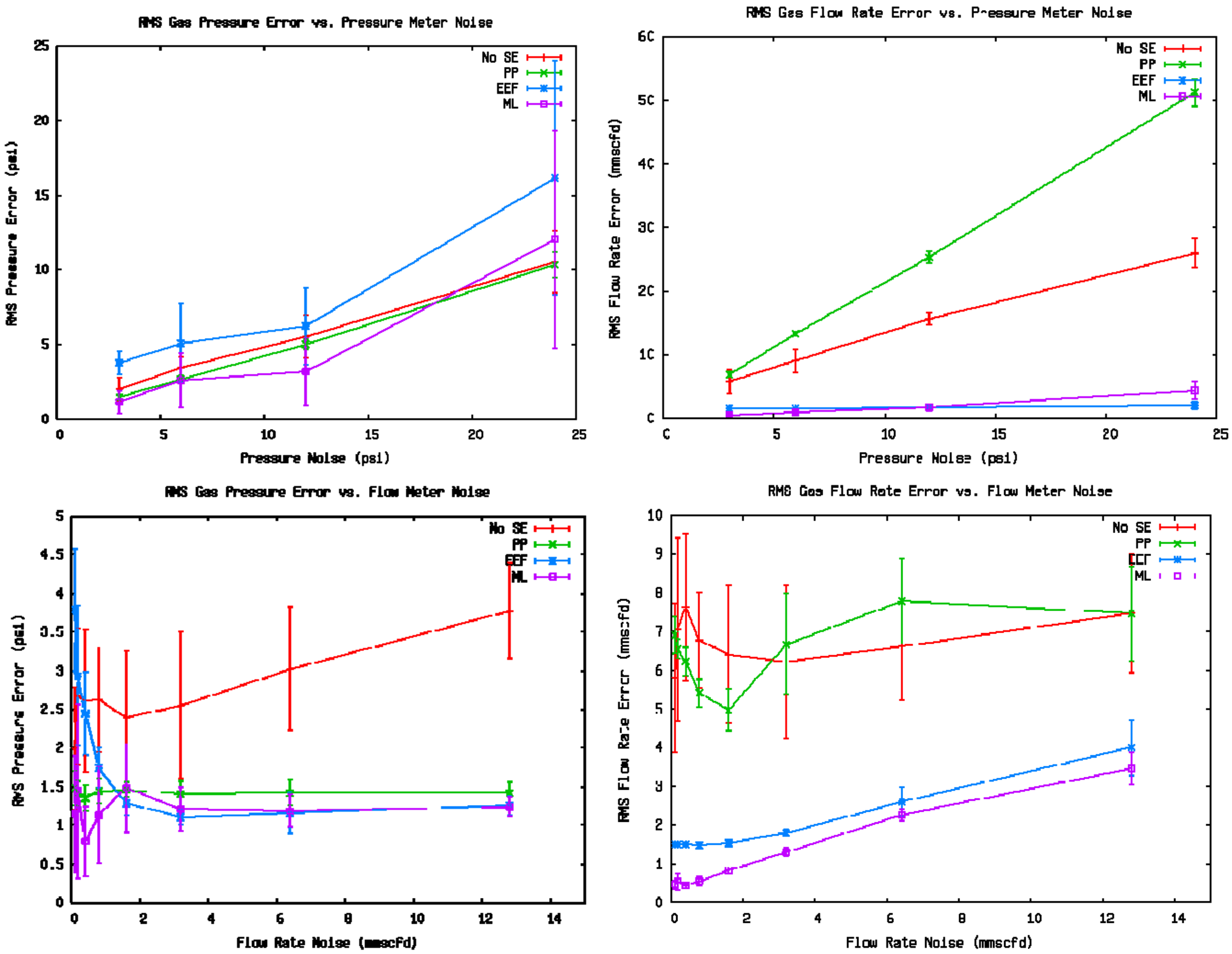

Figure 4 – Differences in accuracy between the various state estimation techniques in the presence of noisy instrumentation on a gas line. "No SE" indicates a simple model re-run using the noisy pressure measurements. PP, EEF, and ML are the pressure-pressure, equal error fractions, and maximum likelihood methods described in the text. RMS errors are averaged over the entire length of the pipe and the duration of the simulation. For reference, the maximum pressure drop in the pipe is 500 psi and the steady state flow rate at this pressure drop is 188 mmscfd.

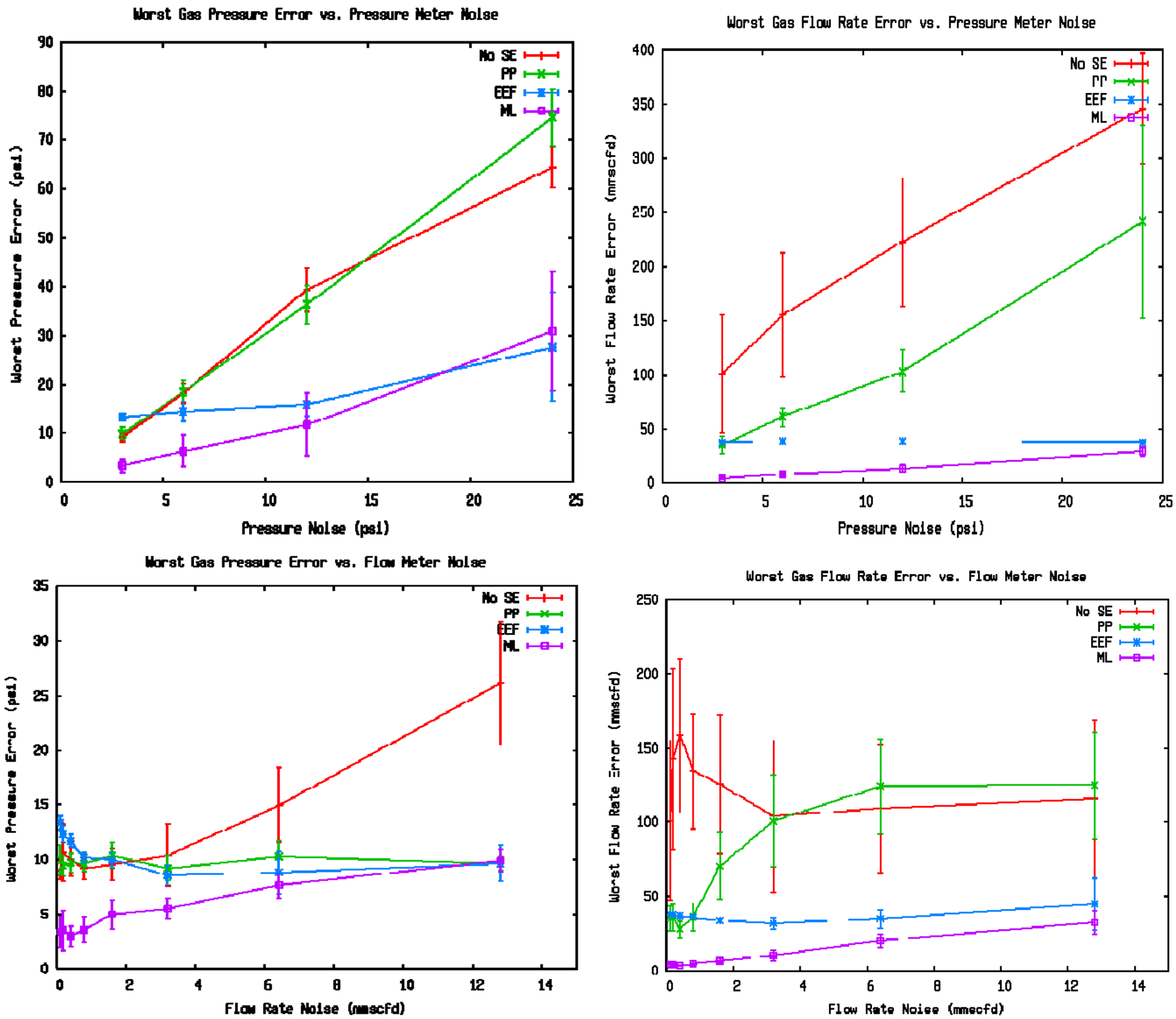

Figure 5: Differences in accuracy between the various state estimation techniques in the presence of noisy instrumentation. "No SE" indicates a simple model re-run using the noisy pressure measurements. PP, EEF, and ML are the pressure-pressure, equal error fractions, and maximum likelihood methods described in the text. Presented are the worst errors at any point from the entire length of the pipe and the duration of the simulation. These results are from the same simulation runs and those presented in Figure 4, above.

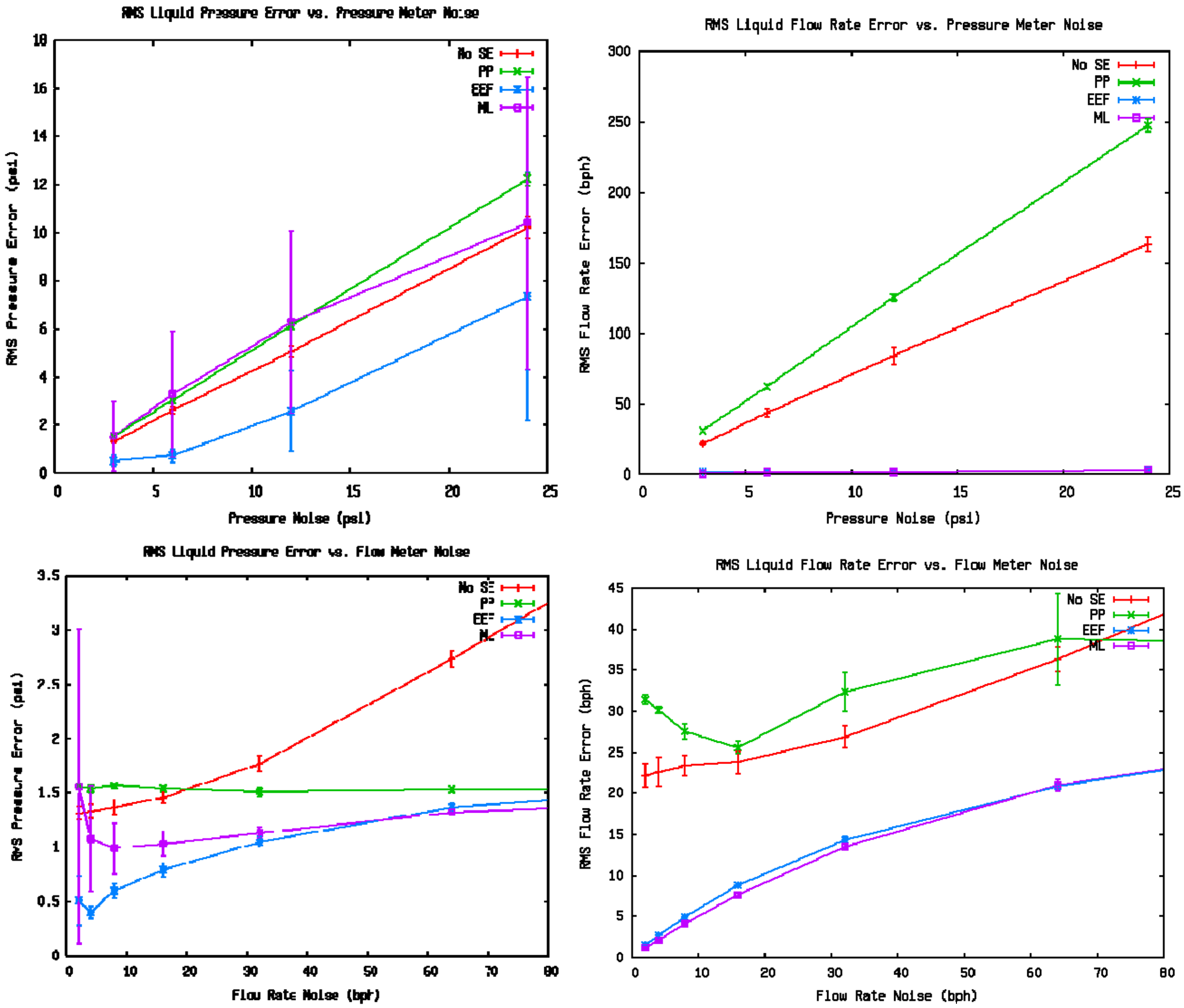

Figure 6: Differences in accuracy between the various state estimation techniques in the presence of noisy instrumentation on a liquid line. "No SE" indicates a simple model re-run using the noisy pressure measurements. PP, EEF, and ML are the pressure-pressure, equal error fractions, and maximum likelihood methods described in the text. RMS errors are averaged over the entire length of the pipe and the duration of the simulation. For reference, the maximum pressure drop in the pipe is 500 psi and the steady state flow rate at this pressure drop is 2200 bph.

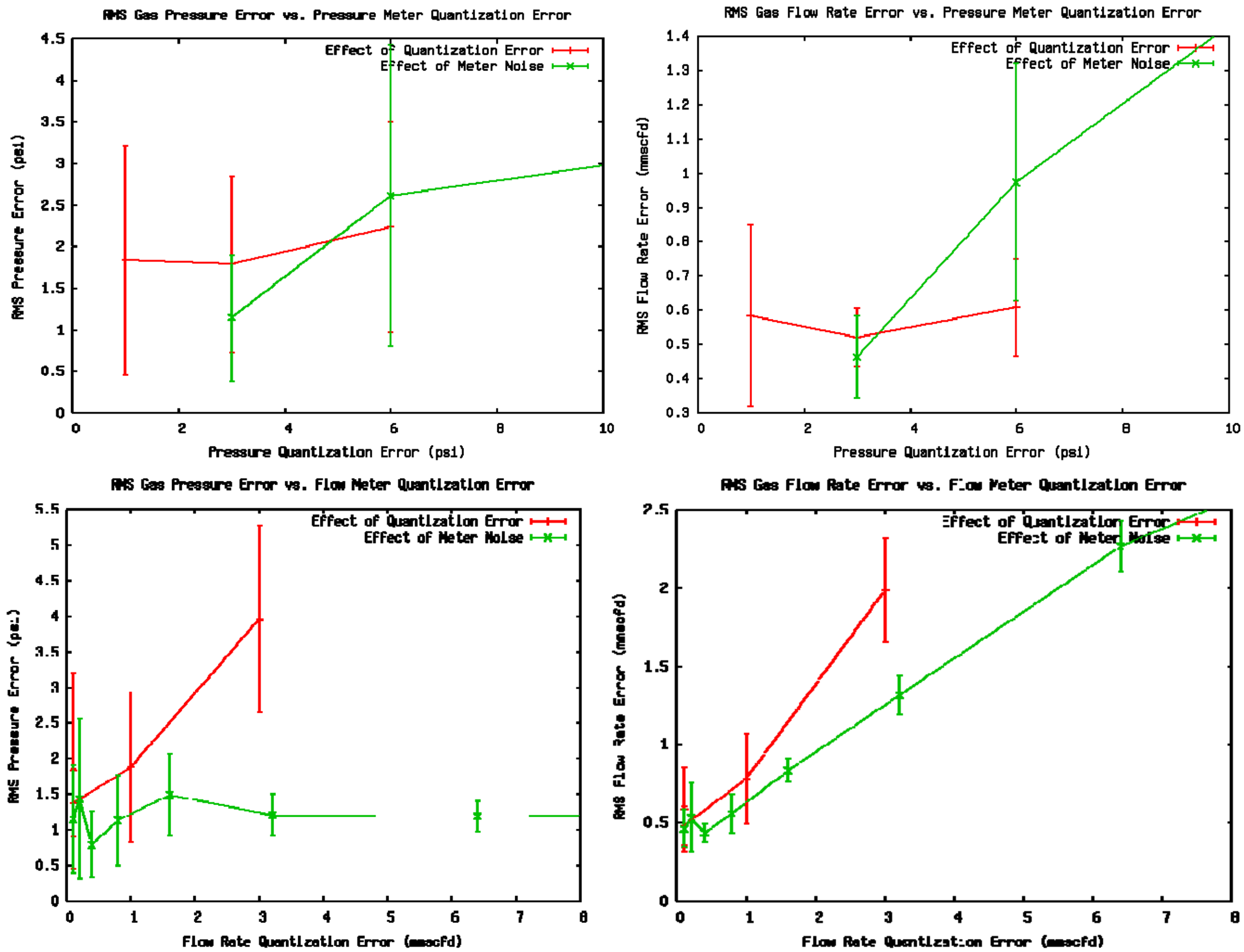

Figure 7: Comparison of the effects of a given magnitude of quantization error vs. the same magnitude of normally distributed noise for a gas pipeline using the ML state estimation technique. RMS errors averaged over the length of the pipeline and duration of the simulation are shown.