# Cointerpretation of Flow Rate-Pressure-Temperature Data from Permanent Downhole Gauges

CS 229 Course Final Report

Chuan Tian chuant@stanford.edu

Yue Li yuel@stanford.edu

## Abstract

*This report documents how we developed appropriate features and applied different algorithms to interpret flow rate, pressure and temperature data from permanent downhole gauges.*

## 1 Introduction

Permanent Downhole Gauges (PDGs) have been installed in modern oil wells to provide a continuous record of flow rate, pressure and temperature (q-p-T) during production. The frequency for data recording could be as high as once per second. After years of operation, one PDG stores millions of data, which provides us rich information about the reservoir. However, currently this data is still primarily used for well monitoring due to the following difficulties in PDG data interpretation.

- **Deconvolution**

In traditional well testing, the flow rate is carefully controlled (usually constant or zero over time), and the pressure response is recorded. The derivative plot of the pressure ($tdp/dt$) can be used to calculate reservoir parameters such as wellbore storage, reservoir permeability and boundary (Figure 1).

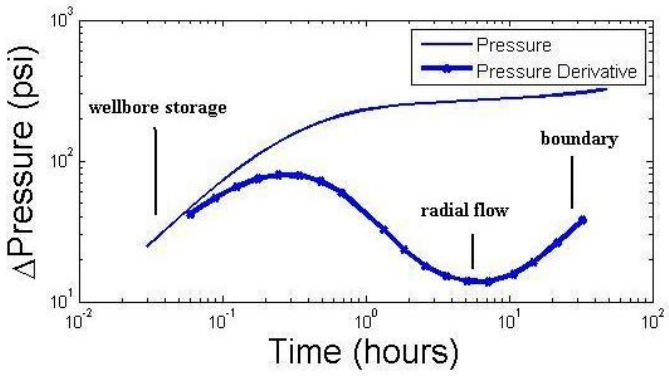


Figure 1: pressure and pressure derivative plot corresponding to constant flow. Reservoir behaviors, e.g. wellbore storage, radial flow and reservoir boundary can be detected from pressure derivative plot

However, well operation includes multiple transient (flow rate changes), and the pressure response is a convolution function of these transients $\Delta p_0(t)$:

$$\Delta p_w(t) = \int_0^t q'(\tau)\Delta p_0(t-\tau)d\tau$$

A real data set from PDG shows the complex rate and pressure history.

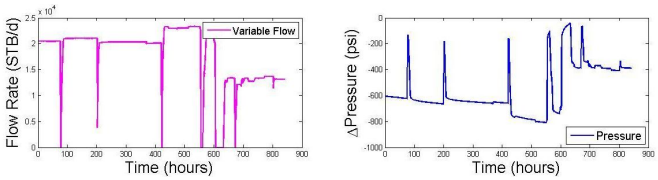


Figure 2: real PDG data (left: flow rate; right: pressure)

To extract the reservoir model from this complex pressure signal, deconvolution is needed, but the solution is usually subject to pathological mathematical instability.

- **Breakpoint detection**

To analyze multi-transient signal, we usually need to break it into individual transient. The quality of the interpretation is very sensitive to the accuracy of breakpoint detection.

- **Volume of data**

Traditional well testing normally lasts for several days, while data recorded by PDGs covers years of production history. This requires a computationally efficient method for data interpretation.

- **Noise**

Noise is commonly seen in PDG data, and makes it challenging for us to discover the real reservoir model behind the noisy data.

Some previous work has been done to interpret p-q data from PDGs using convolution kernel method (Yang Liu, 2013), but there're certain limitations about incomplete interpretation (wellbore storage missing) and computation efficiency. In this project, we tried to develop the features to cover all reservoir behaviors, and applied different algorithms to solve the p-q deconvolution problem on large volume of noisy data. We also tried to recognize the patterns between p-T data. This could be used to construct pressure history based on temperature records, when pressure gauge is not available.

## 2 Background Work

Both synthetic data and real data from PDGs are used in this project. p-q synthetic data is generated by using pressure generator from SUPRI-D research group in Stanford University. It takes flow rate history and reservoir properties as input, and generates pressure response as output. T-p synthetic data is generated by Kappa Rubis reservoir simulator. The real PDG data is also from SUPRI-D. Necessary preprocessing (e.g. sampling) is applied on the data before using.

## 3 Methodology

### 3.1 p-q interpretation

Regularized kernel method, convolution kernel method and kernel SVM were used for p-q interpretation.

#### 3.1.1 Regularized kernel method

- **Feature selection**

Training set: $\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$

Target: $y^{(i)} = p^{(i)};\ i = 1, \dots, m$

Feature: $x^{(i)} = \begin{bmatrix} 1 \\ \sum_{j=1}^{i-1} \Delta q \left(1 - w(\Delta t)\right) \\ \sum_{j=1}^{i-1} \Delta q\, log\Delta t\left(1 - w(\Delta t)\right) \\ \sum_{j=1}^{i-1} \Delta q\, \Delta t\left(1 - w(\Delta t)\right) \\ \sum_{j=1}^{i-1} \Delta q^{\Delta t}\, w(\Delta t) \end{bmatrix}$;

Where $\Delta q = q^{(j)} - q^{(j-1)}$, $\Delta t = t^{(i)} - t^{(j)}$,

Weight function $w(\Delta t) = 1/(1 + e^{4\Delta t - 2})$

These features were selected based on the physical behaviors of Δp as a convolution function of $\Delta q$. $\Delta q^{\Delta t}$ represented wellbore storage; $\Delta q log \Delta t$ represented radial flow; and $\Delta q \Delta t$ represented reservoir boundary effect. Since we calculated $\Delta q = q^{(j)} - q^{(j-1)}$ for every j before query point i, we treated every point as a breakpoint (though it doesn't have to be, e.g. $\Delta q = 0$). This frees us from the inaccuracy in breakpoint detection. The weight function was used to represent the significance of different features in constructing Δp at different flow regimes, since different flow regimes are dominated by different features (Figure 1).

- **Algorithm**

Hypothesis:

$$h_{\theta^{(i)}}\left(x^{(i+1)}\right) = \theta^{(i)^T} \phi\left(x^{(i+1)}\right) = \sum_{k=1}^{m} \beta_k^i K\left(x^k, x^{i+1}\right)$$

Objective: $J(\theta) = \frac{1}{2}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$

Closed-form solution: $\beta = (K + \lambda I)^{-1}\vec{y}$

We expect the close-form solution to be an efficient method that can be applied to large volume of PDG data.

- **Workflow**

| Tests | Input | Output |
|---|---|---|
| Training | Noisy q1, p1 | Parameter |
| Denoising | Clean q1 | Prediction p1 |
| Deconvolution | Constant q2 | Prediction p2 |
| Performance | q3 | Prediction p3 |

Table 1: workflow for p-q interpretation

Denoising test used the same flow rate history q1 as training step but without noise, to test if the algorithm is able to distinguish noise in data. Deconvolution test used constant rate history q2 as input to deconvolute pressure. The derivative plot of p2 can be used to diagnose reservoir parameters. Performance test used a flow rate history q3 that the algorithm has never seen before to test its performance on a new rate input. For all these tests, the pressure predictions were compared with the pressure generated by the pressure generator (denoted as true data in the plots).

**Case 1: synthetic data + linear kernel**

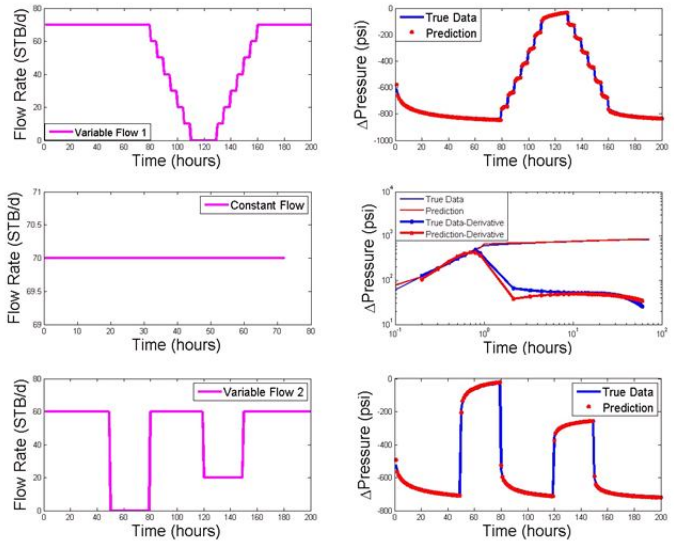


Figure 3: flow rate input and pressure predictions of case 1

We used linear kernel $K(x,z) = x^T z$ in this case. The two plots in the first row represent the data used for training (noise not shown here) and denoising test (clean data shown). The plots in the second row represent deconvolution test with constant rate input. The plots in the third row represent performance test with a different variable flow input from training flow history.

For all the three tests, the pressure predictions (red dots) matched fairly well to the true pressure (blue curve). The pressure derivative plot with constant rate input clearly showed wellbore storage (unit slope), radial flow (flat region) and constant pressure boundary (downwards). These were exactly the reservoir properties we used in the forward model to generate the pressure data. This showed that our algorithm was able to recognize the reservoir model from given p-q data. Define error as:

$$\varepsilon = \|p_{pred} - p_{true}\| / \|p_{true}\|$$

| Training ε | Test 1 ε | Test 2 ε | Test 3 ε |
|---|---|---|---|
| 0.0226 | 0.0061 | 0.0020 | 0.0055 |

Table 2: training and test errors of Case 1

The errors were pretty stable for different tests. Again, this showed an appropriate selection of features.

One defect of this case is that there was a mismatch between prediction and true data in the transition of wellbore storage and radial flow (t=2 hr on pressure derivative plot). This is might because we don't have well defined features for the transition zone. That's why we decided to increase the kernel order to put more features into the system.

**Case 2: synthetic data + 2nd order kernel**

We used 2nd order kernel $K(x,z) = (x^T z)^2$ in this case. The results for most tests remained the same except for the performance test.

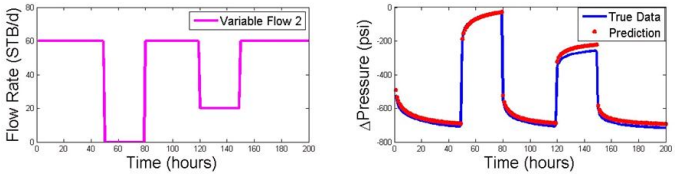


Figure 4: flow rate input and pressure prediction of Case 2

It showed a larger error compared with Case 1. We also tried to user even higher kernel order, but mismatch still existed. The intuition is that higher order kernels brought in unnecessary features, making the problem over-fitted.

**Case 3: synthetic data + Gaussian kernel**

In this case we chose the Gaussian RBF kernel defined as

$$K(x,z) = \exp\left(-\frac{\|x-z\|_2^2}{\sigma^2}\right)$$

The choice of $\lambda$ and $\sigma$ greatly influenced the data fitting. We selected these parameters using simple cross validation, which randomly takes out 20% of the training data as testing sets, and pick the optimal $\lambda$ and $\sigma$ that gives the least testing errors, as shown in Figure.

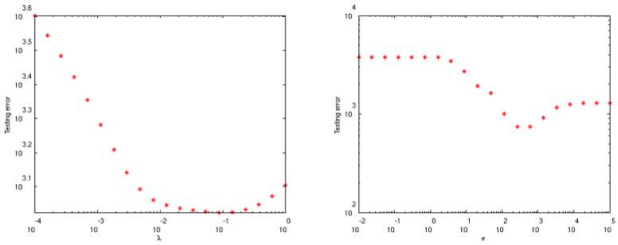

Figure 5: Gaussian kernel parameters selection

Again, the results by using Gaussian kernel didn't show a big difference to linear kernel, and the mismatch in the transition zone still existed.

The performance by different kernel methods showed that linear kernel might be a reasonable approach, if we were able to define the features appropriately based on our understanding of the problem. More complicated kernels gave us more features, but may also make the problem over-fitted.

**Case 4: real PDG data + linear kernel**

Given the performance of different kernels, we chose linear kernel for the interpretation of a real PDG data set. In this

case, we don't know the underlying reservoir model in advance so there's no true pressure to compare with.

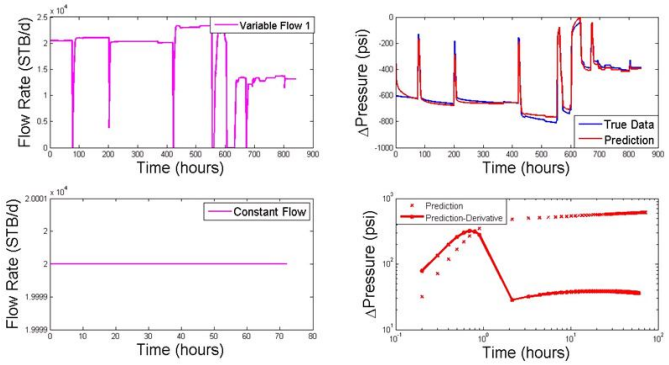


Figure 6: flow rate input and pressure predictions of Case 4

The pressure prediction matched well with the training pressure. The derivative plot corresponding to constant rate showed wellbore storage (unit slope) and radial flow (flat region). This realized the objective of this project to deconvolute multi-transient pressure signal. While in traditional well testing, people usually have to shut-in the well (zero flow rate) to 'manually' deconvolute pressure signal, which brings in additional operation cost.

Besides, about 4000 real PDG data points were used in this case, and it only took less than a minute. This gave us the confidence to apply this method on even bigger PDG data.

### 3.1.2 Convolution kernel method

Another issue with higher order kernel $K(x,z)=(x^Tz)^n$ is that kernel function comes after convolution: $h_{\theta^{(i)}}\left(x^{(i+1)}\right)=\sum_{k=1}^{m}\beta K(\sum \Delta q\ events, \sum \Delta q\ events)$

But physically, the inner product of $\sum \Delta q\ events$ doesn't represent reservoir behaviors. We should use kernel function to construct features first, and then apply convolution,
namely: $\sum_{k=1}^{m}\beta\sum\sum k(\Delta q\ events, \Delta q\ events)$

Where $K\left(x^{(i)},x^{(j)}\right)=\sum_{k=1}^{i}\sum_{l=1}^{j}k\left(x_k^{(i)},x_l^{(j)}\right)$

Though the convolution kernel method is more physically reasonable, its results were quite similar to what we got by using regularized kernel method.

### 3.1.3 Kernel SVM

An alternative solution can be given by SVM regression. We adopt the $\epsilon$-SVM regression approach that minimizes the $\epsilon$-intensive loss function

$$\min\frac{1}{2}\|w\|^2+C\Sigma_i^n(\xi_i+\xi_i^*)\ \text{ subject to}$$

$$\begin{cases} y_i-f_w(x_i)\le\epsilon+\xi_i^* \\ f_w(x_i)-y_i\le\epsilon+\xi_i \\ \xi_i,\xi_i^*\ge 0, i=1,\dots,n \end{cases}$$

Minimizing the $\epsilon$-intensive loss function is equivalent to penalizing errors that are greater than a threshold $\epsilon$. The above optimization problem can be solved efficiently by solving its dual problem, which gives the solution

$$f(x)=\sum_{j=1}^{n_{SV}}(\alpha_j-\alpha_j^*)K(x_j,x)\ \ s.t.0\le\alpha_j,\alpha_j^*\le C$$

where $n_{SV}$ is the number of support vectors, $K$ is the kernel function we have introduced before.

The solution given by SVM is greatly affected by the choice of the parameters C, $\epsilon$ and the kernel parameters. Parameter C determines the trade-off between model complexity and the data misfit, which functions similarly as the $\lambda$ in Ridge regression. Select an infinitely large C is equivalent to minimizing the empirical risk only. Maybe the most appealing feature of SVM is that it can give a sparse solution, in which case the number of support vectors selected is only a small set of the data set. The larger $\epsilon$, the more data fall within the $\epsilon$-insensitive zone (which gives them a zero empirical error), the fewer support vectors used in the regression function.

We further investigate the effect of the number of support vectors on the training errors. By reducing $\epsilon$, in other words, decreasing the number of SVs in the regression function, the training error gradually decreases. However, adding the first 10 training data sets constitutes 99.89% of the error production, and add the following data sets from 10 hour to 100 hours contributes another 0.1% of the error reduction. This suggests that by capturing the early time behavior in each pressure transient that immediately follows the flow rate change, one can predict accurately the reservoir response. Running Kernel SVM provided us a deeper understanding of the p-q problem.

## 3.2 p-T interpretation

Regularized kernel method was used for p-T interpretation.

- **Feature selection**

Training set: $\{(x^{(i)},y^{(i)});i=1,\dots,m\}$

Target: $y^{(i)}=p^{(i)};\ i=1,\dots,m$

Feature: $x^{(i)} = \begin{bmatrix} 1 \\ T^{(i)} \\ {T^{(i)}}^2 \\ \vdots \\ {T^{(i)}}^n \end{bmatrix}; \ i = 1, \dots, m$

Physically, well downhole temperature is a function of pressure and flow rate: $T = T(p(q,t), q(t))$. We did it inversely from p to T, because T data is more accessible than p and p is more widely used for analysis.

This inversion also brought in challenges for feature definition. We first tried the polynomials of T based on intuition from the shape of p-T curve.

- **Algorithm**

The same as p-q interpretation, using closed-form solution.

- **Workflow**

| Tests | Input | Output |
|---|---|---|
| Training | Noisy T1, p1 | Parameter |
| Denoising | T1 | Prediction p1 |
| Deconvolution | T2 | Prediction p2 |
| Performance | T3 | Prediction p3 |

Table 1: workflow for p-q interpretation

The workflow was similar to p-q interpretation. For the deconvolution test, the temperature input corresponded to a constant flow rate.

**Case 6: synthetic data + linear kernel**

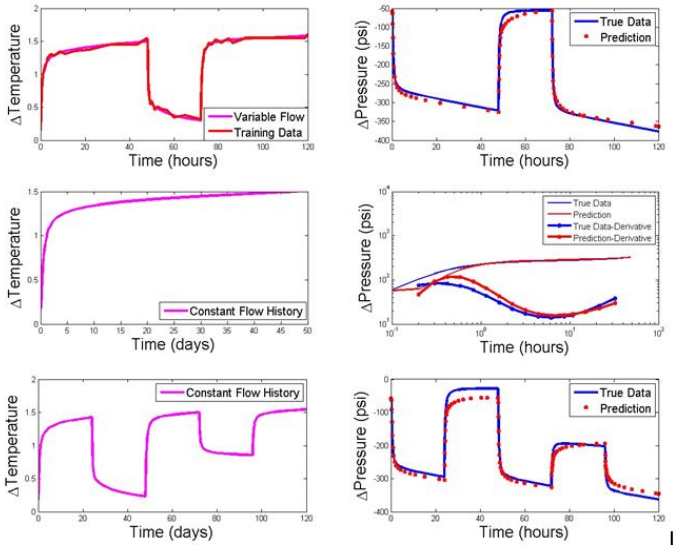


Figure 7: temperature input and pressure prediction of Case 6

For all the tests, it showed a reasonable match between the prediction (red dots) and true data (blue curve). The pressure derivative plot clearly showed wellbore storage (unit slope), radial flow (flat region) and closed boundary (upwards). This showed that the algorithm was able to capture the overall reservoir model based on given p-T data. we still need to have a deeper study on the feature selection to get a better match.

In practice, the availability of different data is $T > p > q$. A direct application of this method is to construct pressure data from distributed temperature sensor (DTS, while distributed pressure sensor not commonly seen due to cost). If we have one pair of p-T sensors in a reservoir, we can use this method to learn the pattern between p-T. Then given temperature data from DTS, we can construct the pressure map for the reservoir, which can provide crucial information for reservoir development and management. Moreover, if we were able to correctly construct temperate data corresponding to constant flow, then we can interpret the generated pressure signal to get reservoir properties, just like what we did in p-q interpretation.

## Conclusion

In this project, we developed appropriate features and applied different learning algorithms to describe the patterns of p-q-T data from PDGs. The results showed that our method was able to recognize the underlying reservoir model by solving the devonvolution problem. This work provides a way for us to better utilize data from PDGs.

## Acknowledgement

We would like to express our gratitude to Prof. Roland N. Horne, professor from Energy Resources Engineering Department, for his advice on this project.

## References

[1] Horne, Roland. "Listening to the Reservoir - Interpreting Data From Permanent Downhole Gauges." *Journal of Petroleum Technology* 59.12 (2007): 78-86.

[2] Liu, Yang, and Horne, Roland. "Interpreting Pressure and Flow-Rate Data From Permanent Downhole Gauges by Use of Data-Mining Approaches." *SPE Journal* 18.1 (2013): 69-82.