Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/specialization: Computer Science	Spring semester 2022 Open/ ~~Confidential~~

Author: Lejun Chen

Supervisor(s): Anton Shchipanov and Chunming Rong

Title of Master's thesis:
Transient identification and interpretation of real-time pressure data from modern well surveillance systems

ECTS: 30

Keywords: Permanent downhole gauge (PDG), Pressure transient analysis (PTA), Transient identification, Break points identification, Pattern recognition	Pages: 117 +Attachment/Others: **Scripts:** **click** **Deployed App:** **click** Stavanger 14 June 2022

Abstract

Permanent downhole gauges (PDG) that are installed in modern wells are capable of recording pressure and temperature data during well operations. These data in combination with flow rate measurements may be utilized for Pressure Transient Analysis (PTA) or its time-lapse version - time-lapse PTA. Unlike traditional PTA which would mainly interprets data from shut-in periods, time-lapse PTA is applied to sequential pressure transients and both flowing and shut-in periods to evaluate well-reservoir parameters that may vary with time. A PTA workflow may be divided into several steps. In this study, we mainly focus on the step of transient identification based on pressure measurements, which is crucial for time-lapse PTA and keeps being a challenge especially when the data are very noisy. Another step - denoising is also carried out and discussed in the part of computational experiments.

Previous work in the area of the transient identification has been studied before developing and testing of own algorithms to address this problem. Seven approaches have been reviewed, compared and summarized in the thesis.

In this master thesis, the transient identification problem is considered as a two-stage process consisting of two identification tasks: 1) splitting the data set into shut-in and flowing periods; 2) task 1 + detecting multi-rate break points in flowing periods. Three methods based on pressure derivative, tangent of pressure data and pattern recognition were proposed and studied and a workflow to implement these methods was designed.

Through a number of computational experiments, the derivative-based method provided the best result for the task 1, while tangent-based method - best results for the task 2. In the comparison section, we discussed the results of testing of these methods taking an insight into performance of the break-point detection in the different tasks.

The thesis is finalized by establishing an easy-to-use web application, where the users could experiment with different methods using own data sets. The web-app provides a friendly user interface to upload and test different data sets and download results of the break-point detection.

Acknowledgments

Firstly, I will give my deepest gratitude to my supervisors Dr. Anton Shchipanov and Prof. Chunming Rong for their constant guidance, advice, and encouragement, and these will be wealthy of my lifetime.

I also would like to express appreciations to PhD. Candidate Boyu Cui and Dr. Nan Zhang for their valuable comments and suggestions.

I am especially thankful to my husband and two lovely daughters for their unconditional support and love.

Contents

Chapter 1

Introduction

1.1 Background

Since 1990, a great number of wells have been equipped with permanent down-hole gauges (PDG) which record real-time data, including pressure measurements and temperature measurements, which are usually called Bottom Hole Pressure (BHP) and Temperature (BHT). Especially with the steadily falling of the cost for the installation and the rise of the reliability, there has been a significant growth in the number of PDG systems that had been installed in oil and gas well worldwide. Thus, it become crucially important to utilize these real time data not only to monitor the reservoir characterizations, but also to improve field development and performance. One of the most important tools to utilize the PDG data is time-lapse Pressure Transient Analysis (PTA) which mainly focus on interpretation of BHP.

PTA had been investigated for decades, which is even earlier than the emergence of PDG. Ramey et al.[1] described the usage of PTA in diagnosing well condition. It took the pressure measurements and flow data set as input and involved matching a model of the reservoir to the data. Through PTA, some information and parameter of the reservoir will be obtained and estimated, e.g., the reservoir transmissivity (kh), skin factor, the initial reservoir pressure, permeability, etc.

Many researchers, e.g. Allain et al.(1988) [2], Ershaghi et al.(1993) [3], Sinha and Panda(1996) [4], AlMaraghi and El-Banbi (2015) [5], etc., contributed to this tool.

Traditional PTA only interpret pressure data in shut-in periods, time-lapse PTA is applied to both flowing shut-in periods. Certain data proecessing and transient identification are needed before the application of time-lapse PTA [6]. In 2002, Athichanagorn et al.[7] proposed a seven-step workflow to address the problem.

1. Outlier removal
2. Denoising
3. Transient identification / break point detection
4. Data reduction
5. Flow history reconstruction
6. Aberrant segment filtering
7. Transient analysis on moving windows

A brief description of each step is given below for a better understanding.

Outlier Removal Outliers are data points which locate far from the regular trend of pressure response due to unpredicted environment, significant disturbances at the down-hole, temporary sensor failure, unstable transmitter and etc. Wavelet-based methods were recommended by Athichanagorn (1999) [8] and Ribeiro et al. (2006) [9]. Olsen et al. (2005) [10] demonstrated that median filter is more efficient comparing with wavelet-based methods.

Denosing The target of denoising is to smooth the fluctuations caused by noise and keep the information of data set as much as possible. Wavelet filtering (Donoho et al.,1994[11]) is proven to be one of the most effective approaches. Haar wavelet combined with soft-thresholding was suggested by Kikani et al.(1998)[12]. More denosing technique will be discussed in other chapters of this thesis.

Transient identification Pressure measurements response when the flow rate change. The abrupt change of pressure measurements will occur due to the sudden change of flow rate. By recognizing these abrupt changes of pressure measurement, the distinct flow periods will be separated. Since it is the main objective of this study, further demonstration will be made in the following chapters.

Data reduction Numerous data are recorded by PDG. To speed up the processing time, data reduction techniques are developed to reduce the number of data points while keeping the features carried by the data set. Athichanagorn (1999) [8] suggested a hybird method of pressure thresholding combined with time thresholding.

Flow history reconstruction The flow rate sometimes are fail to be recorded by PDG. The flow history could be reconstructed based on pressure and temperature measurements. Various methods are proposed by many researchers, Athichanagorn (1999) [8] used the nonlinear regression method. McCracken et al.(2006) [13] use pressure data to allocate the rate period. A feature-based machine learning approach was developed by Tian et al. (2015) [14].

Aberrant segment filtering The aberrant segments which do not follow the regular trend of pressure response must be removed before interpreting these data. The variance between the data and the model response could be utilized to distinguish these segments.

Transient analysis on moving windows This step is to analyse the reservoir behavior in order to determine reservoir properties based on the moving average window.

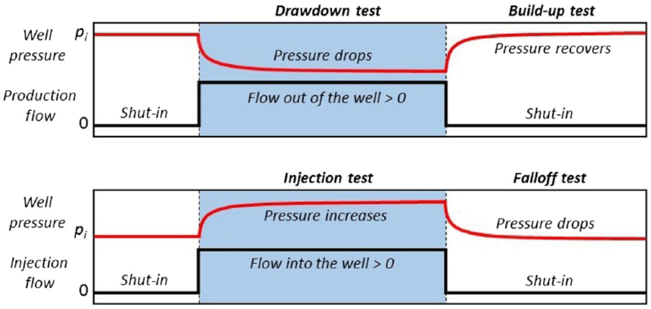


Figure 1.1: Schematic for the relationship between of change of pressure and the change of production flow or injection flow[15].

1.2 Problem statement

Among the seven steps that we discussed above, this study will mainly focus on the third step, transient identification, a.k.a. breakpoint detection, which is crucial for further analysis and interpretation of the PDG data. In addition, some denosing techniques also will be discussed and utilized to improve the accuracy.

1.2.1 The definition of pressure transient & break point

The term ‘pressure transient’ can be referred to any circumstance in which pressure is changing with time[15]. These pressure changes are usually caused by a change of flow rate which is usually measured at well tubing head (top of the well) with flow-meters. The flow rate can be either a production flow or injection flow, shown in Fig. 1.1. The start point of a pressure transient is called a ‘break point’.

1.2.2 The pressure & flow data sets

Two most relevant data sets to our study are pressure data and flow rate data. Both data sets will be inputted into our implemented models, however, the identification is only based on pressure data. The flow rate data, which are usually indications of true break points, are used to evaluate the performance of the implemented model. There are some characteristics for these two data types.

- For pressure data, various sampling rates at low and high frequencies depend on the system setup. Normally, it is recorded at several seconds interval. One year of data may consists of over three million measurements.
- For flow rate data, sometimes they will be measured only once a week or once a month while there are unmeasured rate changes in between.
- Both two types of data have noise because of different reasons. A number of outliers or zero values exist due to erroneous recording.
- An important relationship between flow rate and pressure is that the change of flow rate will produce the change of pressure. However the pressure and flow rate might be measured at different locations/gauges, which causes issues like synchronization, different sampling rate etc.

The more we know about the data, the better solution we will develop for the design of methods and the workflow. For instance, the large number of points will require a strategy to reduce the number before these points is about to go through a computational cheap module if it exists. And the unevenness of data sampling also needs to be taken into consideration.

1.2.3 The objective

The objective of this study is to explore some novel methods for the transient identification problem. A work flow will be designed to integrate these methods. The final model is expected to be:

- highly accurate
- minimum user interaction
- universal criteria

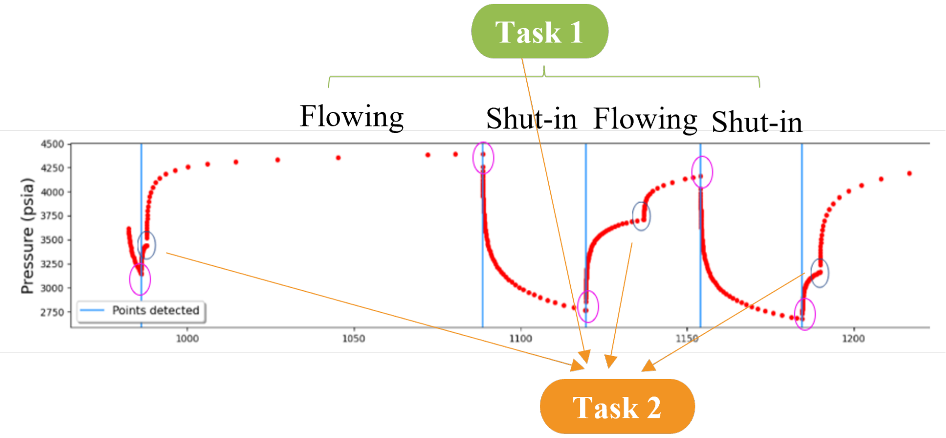


Figure 1.2: Schematic of break points identification objective[15]. *Task 1*: separate flowing and shut-in periods. *Task 2*: *Task 1* + multi-rate break points detection in flowing periods.

- tolerant to noise
- computational cheap

We model this problem as two-stage pipeline of sequence identification tasks, illustrated below:

- *Task 1:* Automated breakpoints identification which separate flowing and shut-in periods.

  As shown in Figure 1.2, the pressure measurements need to be splitted into shut-in flowing periods.

  Points(annotated by purple ellipse), which are the start points of different periods must be identified.

- *Task 2:* *Task 1* + Automated breakpoints identification for multi-rate flowing periods with dominating transient.

  Points, annotated by grey circles in Figure 1.2, indicate the major multi-rate transients in flowing period.

1.3 Thesis outline

This thesis proceeds as follows.

Chapter 2 reviews seven approaches proposed in last two decades. The detecting theories are introduced, experiments results are also discussed. The comparisons of these methods are made in the aspects of: main idea, pros, cons and user input.

Chapter 3 designs the methods based on the derivative values of the pressure measurements. We propose two algorithms, named as *RangeFOD* and *DeltaFOD*. Two other useful approaches which could help with transient identification are also introduced.

Chapter 4 illustrates tangent method for detection. The theory is introduced first. Then an algorithm name as *DeltaTangent* is suggested.

Chapter 5 implements an approach of pattern recognition. A new idea about how to learn the patterns is demonstrated. Two tuning methods, named as *Percentile Tuning* and *Fine Tuning* are suggested to improve the learned patterns.

Chapter 6 proposes a new workflow to solve this two-tasks problem of this study. After that experiments with various combination of different methods are carried out.The results are compared and analyzed. The discussion of methods and patterns of break points are made at the end of this chapter.

Chapter 7 deploys the model which embeds these designed methods and workflow in this study.

Chapter 8 concludes major outcomes of this study and gives suggestions for the future work.

Chapter 2

Previous work

In last two decades, many methods for transient identification were suggested. The thorough research of previous work will help to form a solid foundation to develop our own methods. We will do a comparative analysis to seven approaches in this chapter. The summarization will be made at the end of this chapter.

2.1 Spline wavelet decomposition

A spline wavelet decomposition method was discussed by Athichanagorn[8] in his doctoral dissertation in 1999. Wavelet analysis has the capability of analysing data at multiple levels of resolution. The spline wavelets use the concept of wavelet modulus maxima which are suitable for singularity detection to determine the breakpoints.

- At high level, the signal singularities for detecting the beginning of transients are group as a single singularity.
- At low level, signal singularities and noise singularities both present.
- At intermediate level, only signal singularities present. Thus, it is difficult to distinguish signal singularities in both high level and low

level. We make use of intermediate level to detect the transients.

One thing needs to note is that two criteria should be determined first before applying this method, listed as follows:

- Δt_{min} : the length of shortest transient to be detected.

  The intermediate level for different data sets may vary. In order to determine which level is the most suitable for an arbitrary spacing, the length of the shortest transient to be detected Δt_{min} should be chosen at first.

- Threshold: to determine which change is actually a singularity.

  The rule of thumb in choosing the threshold is to determine how much pressure difference between two data points is considered as the beginning of a new transient.

The detailed detecting steps were also discussed by the author in his paper[8].

The author did many experiments towards different data sets including raw data and denoised data.

Through these experiments, the author reached the conclusions that denoising would benefit the transient detection towards the noisy data. However, it might produce the problem of over smoothing when the associated noise is minimal.

2.2 Pattern-recognition approach

As shown in the discussion of the previous section that the spline wavelet decomposition has the demerits as requirements for expert use and large computation. Olsen et al.[10] proposed a pattern recognition method to identify the pressure transients.

There are two ways to determine the pattern used in this method.

- To predefine a pattern.
- To use a system learning process, like neural networks, to train the system for pattern recognition.

The author illustrated the first way in his paper. He predefined two patterns for transients, one for build-up transients, one for draw-down transients as shown in Figure 2.1 and Figure 2.2.

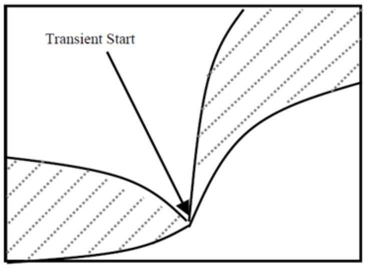


Figure 2.1: Pattern recognition for build-up transients[10].

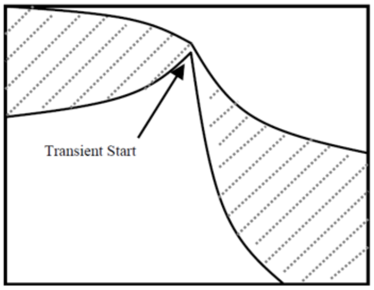


Figure 2.2: Pattern recognition for draw-down transients[10].

When the entire transient image falls in the range of the predefined transient pattern, it will be identified as a transient.

There are three prerequisites for apply this method:

- Outliers must be removed.
- Denoising is recommended especially when the noise level is large.
- Not to use a noise filter that will seriously over smooth the transients.

The author also proposed a complementary measure to remove the smallest transient pattern by adding some statistical validation criteria, which will miss some smallest transients, but also decrease the number of false detections.

The experiment result carried by the author showed the false detection was fairly reduced while approximately 10% of transients were missed because of small amplitude.

2.3 Three approaches by Rai

In 2005, Rai[16] investigated the limitations of spline wavelet-based approach and proposed four algorithms as follows.

- Haar wavelet (Stationary wavelet)
- Savitzky-Golay FIR smoothing filters
- Segmentation method
- A variant of segmentation method

Through author's experiments with real field data, the first approach, which used one of the orthogonal wavelet transformation, did not show any significant improvements comparing with spline wavelet-based (non-orthogonal) approach. The other three however made a considerable improvement in accuracy and reliability.

In this section, we will mainly discuss the latter three approaches.

2.3.1 Savitzky-Golay FIR smoothing filters

The basic idea of this method is to apply the Savitzky-Golay smooth filter to the pressure data, then analyze the first and higher order derivative of the smoothed pressure measurements to obtain the break points.

The derivatives of pressure data exhibit a peak when there is a sudden change in the data. Hence the peaks in first and other higher derivatives indicate the vicinity of the break point location. For an abrupt change in the data, the first and second derivative peaks will be closer to the actual break point location in the data, while for a smoothly changing data, the higher derivatives namely third and fourth will be closer to the break point location.

According to the above theory, the detecting procedure designed by Rai could be concluded as the following two steps. Firstly, to find the peaks of first order derivative of pressure measurements above a certain low threshold, then search peaks in the second, third and fourth derivatives of the data. All these positions of the peaks indicate the possibility of break points. Secondly, to check the slopes between each pair of locations to see if they are above a certain threshold which is set as the average noise variance estimated from the data. Thus, the interpolation between these locations could be finally carried out.

The window size of Savitzky-Golay smooth filter plays a crucial role in the performance, thus should be carefully determined. The author provided a example to show how to decide the window size.

Figure 2.3 and Figure 2.4 show the derivative plots for window size of 21 and 51 respectively. By comparing these two figures, both of them could clearly show the positions of break points through the first order derivative. However the higher order derivative are still very noisy for window size 21. For the data set used in this example, obviously the window 51 performs much better and thus should be chosen.

The author also did experiments to compare the results obtain from spline wavelet and S-G smoothing filter and draw conclusion that the latter performed much better than the former.

2.3.2 Segmentation method

In order to achieve a full automation, the minimum user inputs are desired. Segmentation method just require one user input parameter for the breakpoints identification. This parameter's value can be obtained by the

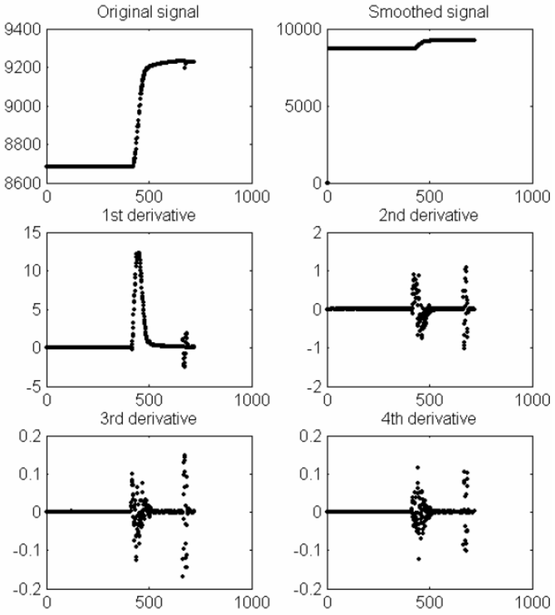


Figure 2.3: S-G smoothing and derivative calculation for window size of 21[16].

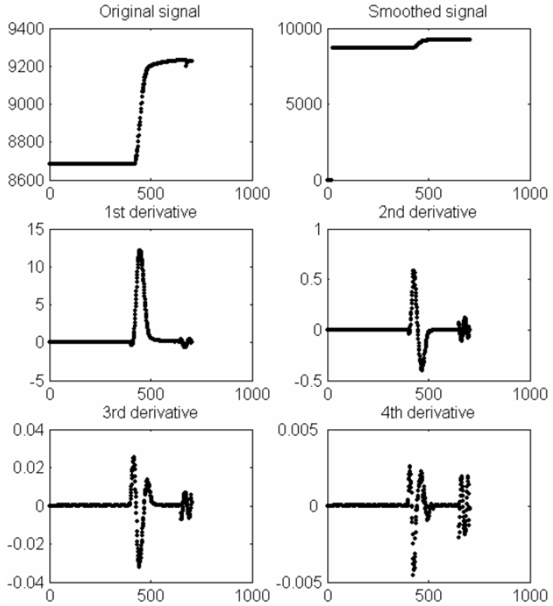


Figure 2.4: S-G smoothing and derivative calculation for window size of 51[16].

knowledge of the resolution of the instrument or from an estimation of the noise level of the data. Another advantage for this approach is that no denoising techniques are required.

The algorithm involves the following three steps.

- The first step is to collect a set of strategic points, by solving a sequence of maximum orthogonal (Euclidean) distance problems.

  To begin with, the first and the last points in the dataset are marked as two strategic points, see Figure 2.5. Then another point is selected whose orthogonal distance from the line segment joining the two strategic points is greatest. This third point joins the collection of strategic points and, in turn, becomes an end point for two new line segments, see Figure 2.6 and 2.7.

  This iterative numerical scheme is performed until the greatest orthogonal distance of data point from the associated line segments falls below a prescribed tolerance τ. This tolerance can be estimated statistically from the dataset itself. Also the selection of these points does not require equally spaced data in time.

- The second step is to calculate the area under each strategic point by considering the two other strategic points to its either side.

  The area can be approximated numerically through a trapezoidal rule or polygon method. The small or false break points are then deflated using a very small area cutoff which is taken as a very small fraction of the highest area in the dataset.

- The third step is to screen further by checking the forward and backward slope.

  As a rule of thumb, if the forward slope is less then half the backward slope, the point is discarded. The slopes were obtained by least square fitting.

From the above description of steps we could know that the tolerance τ will affect the number of detected points. As suggested by the author, the lower bound for the tolerance τ can be considered as an estimate of noise in the data. The estimated denoising threshold in the data are denote as σ. The author utilized an algorithm proposed by Khong [17] to determine σ.

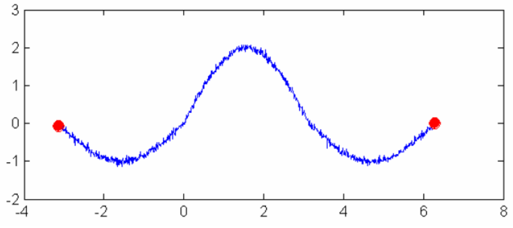


Figure 2.5: The first and the last point in the dataset marked as two strategic points.[16].

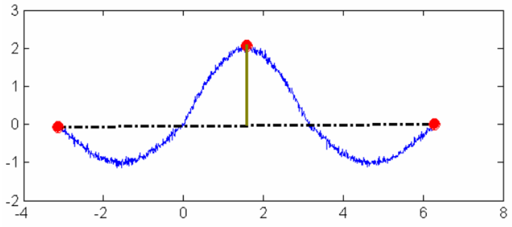


Figure 2.6: Point with maximum orthogonal distance from the line segment joining the first two strategic points selected[16].

The author tested two data sets with various values of the tolerance τ ranging from 4σ to 40σ. When the value of τ increase from 4σ to 40σ, for the first data set, the detected break points decreased from 13 to 11, while for second data set, it decreased from 16 to 13. The author reached conclusions that i) this method is reliable since all the significant break points have been identified for a wide range of the τ; ii) for a wide range of tolerance τ the number of detected break points do not vary significantly; iii) with the increase of τ, only a small number of minor break points will be missed.

2.3.3 Variant of segmentation method

Most of the time only pressure data are measured from permanent downhole gauges. However, if simultaneously measured flow rate is also available along with the pressure data, then it is possible to use this additional information to aid the detection break points.

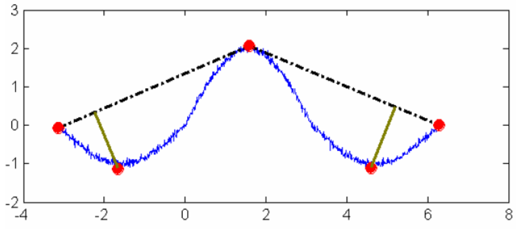


Figure 2.7: Points with maximum orthogonal distance from the line segment joining each consecutive pair of strategic points selected[16].

This section describes an algorithm suggest by Rai, which uses a concept similar to the segmentation method and can be called a variant of the segmentation method. However the variant also use the flow rate data to aid the detection. This variant can be used either independently, to determine the significant break points, or in conjunction with other methods, to validate and screen smaller or false break points.

The detection includes the following two steps.

- The first step is to identify the beginning and end of the time period when the liquid rate data is zero.

  Because when the well is closed for build up, the liquid rate will go to zero after some time and when the well is again opened, the liquid rate will increase from zero to some finite value. The time axis values of the strategic points obtained in the liquid rate data can be transferred onto the pressure data.

- The second step is to use the concept of the segmentation method and look for points, which derived from step 1, in the pressure data having maximum or minimum orthogonal distance from the time axis, between a pair of strategic points.

  The point with minimum orthogonal distance will be the break point corresponding to the beginning of a drawdown, while the point with maximum orthogonal distance will be the break point corresponding to the beginning of a buildup.

The author also tested this approach with two field data sets, where both pressure and flow rate data are simultaneous recorded. For both two data sets, only one correct point is missed and one break point is false detected.

2.4 Data smoothing technique

Nomura[18] proposed a data smoothing techniques, in which the identification problem was treated as a pressure fitting problem. Then the problem becomes finding the break points in the pressure signal with the nonlinear smoother.

The detecting procedure was divided into two stages: insertion stage and deletion stage, shown as Figure 2.8. The wavelet processing results are taken as the initial guess. The basic idea of the inserting stage is to insert additional break points by the iterative procedure of pressure fitting, checking GCV score and residual trend. The deletion stage is to delete the break points by the iterative procedure of pressure fitting, checking GCV score and effective flow rate. The final results will be chosen according to the minimum GCV score.

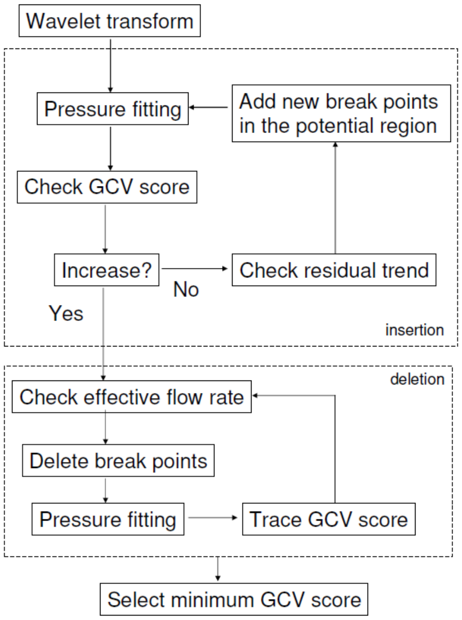


Figure 2.8: Nomura's procedure for the transient identification.[18].

Nomura also tested his approach using computational experiments and illustrated in details how he applied his method in practice, shown as Figure 2.9. For the testing data that used by Nomura, the initial guess obtained from wavelet method was 18, while the number of true break points was 57. In the insertion stage, the GCV score kept decreasing with the insertion of break points. After the GCV score reached its lowest value (see the blue arrow in Figure 2.9). The increase of break point will raise the GCV score, indicating the over fitting. Then the deletion stage began to work, and the GCV score was reduced further, and the lowest GCV score in the deletion stage was observed at the position pointed by red arrow in Figure 2.9, which was selected as the final result shown as Figure 2.10.

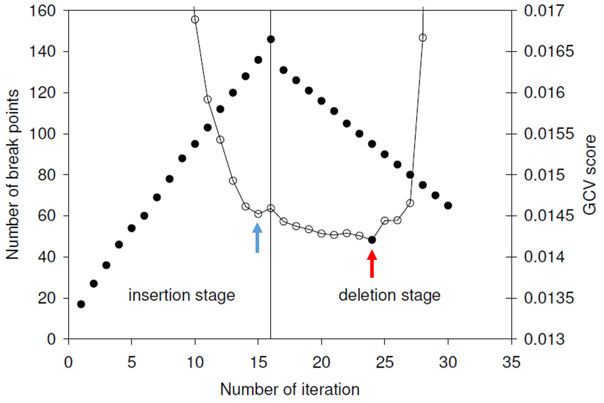


Figure 2.9: The GCV score and the number of break points[18]. Blue arrow & red arrow: indicate the lowest value of GCV score in insertion stage and deletion stage respectively.

Normura concluded that the break points detection was successful with this algorithm, however there are several difficulties.

The first difficulty is that the insertion and deletion scheme is not guaranteed to produce the best result. The second difficulty is the GCV minimization is not guaranteed to always produce a reasonable model due to the lack of local information.

Based on the results obtain from the experiments and lacking a better estimator, the author adopted the sequential GCV minimization and claimed resulting estimates were all reasonable in practice.

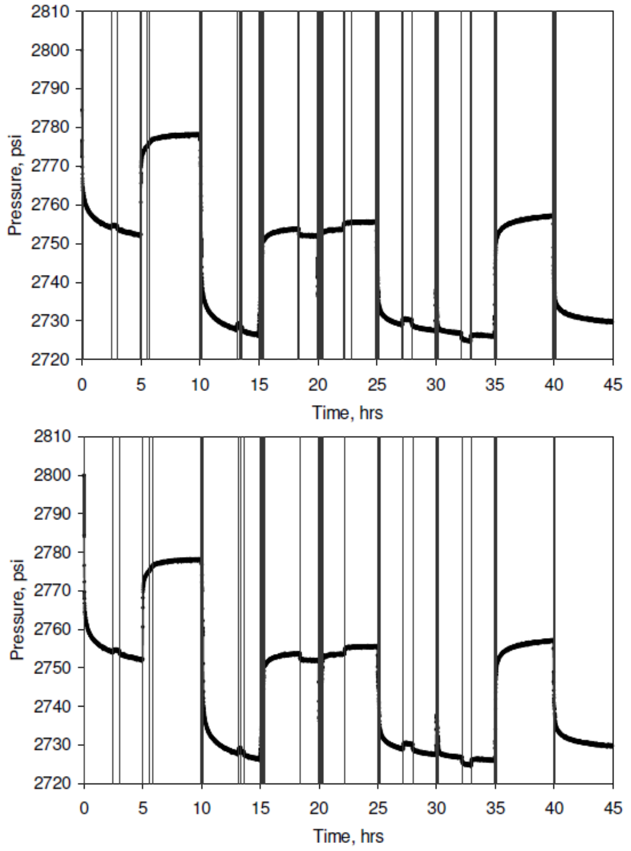


Figure 2.10: The estimated break points location. Upper: estimated location of the break points. Lower: location of the true break points[18].

2.5 Filter convolution technique

Suzuki et. al[19] proposed a detection method using filter convolution. The filter is designed to detect a paritcular pattern of pressure response rather than just detecting a frequency change. Also, it can be combined with noise-removal filters, thus enhancing the tolerance to noise.

Suzuki et. al observed that for a noise free pressure data, the positions of break points usually locate at where i)the sign of pressure derivative(dp/dt) switch from negative to positive or positive to negative; ii) or the absolute pressure derivative changed from a close to zero value to a relatively large value, shown as Figure 2.11. Suzuki et. al used the Equation 2.1 and 2.2 to depict these patterns.

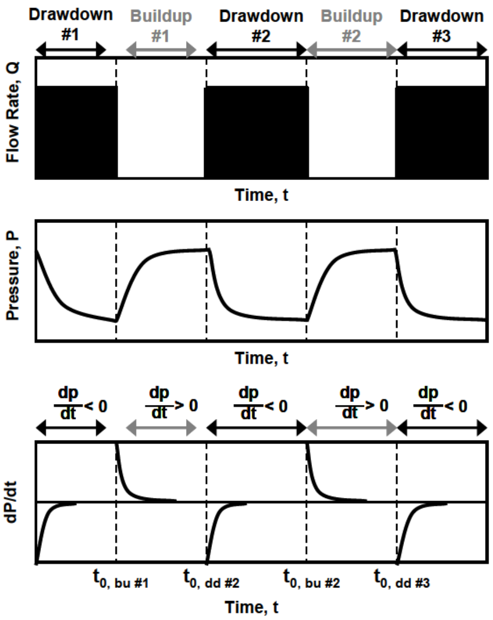


Figure 2.11: Schematic of pressure response[19].

$$\begin{aligned} \text{Start of buildup:} \quad & \left.\frac{dp}{dt}\right|_{t=t_{0,bu-\varepsilon}} < 0, \quad \left.\frac{dp}{dt}\right|_{t=t_{0,bu+\varepsilon}} > 0 \\ \text{and/or:} \quad & \left.\frac{dp}{dt}\right|_{t=t_{0,bu-\varepsilon}} \approx 0, \quad \left.\frac{dp}{dt}\right|_{t=t_{0,bu+\varepsilon}} \gg 1 \end{aligned} \tag{2.1}$$

$$\begin{aligned} \text{Start of drawdown:} \quad & \left.\frac{dp}{dt}\right|_{t=t_{0,dd-\varepsilon}} > 0, \quad \left.\frac{dp}{dt}\right|_{t=t_{0,dd+\varepsilon}} < 0 \\ \text{and/or:} \quad & \left.\frac{dp}{dt}\right|_{t=t_{0,dd-\varepsilon}} \approx 0, \quad \left.\frac{dp}{dt}\right|_{t=t_{0,dd+\varepsilon}} \ll -1 \end{aligned} \tag{2.2}$$

In Equation 2.1 and 2.2, ε denotes a positive infinitesimal time interval.

$t_{0,dd}$ denotes the draw-down break point in time. $t_{0,bu}$ denotes the build-up break point in time.

In order to detect the position where the switch of the sign of dp/dt occurs, Suzuki et. al use the filter convolution method, shown in Figure 2.12.

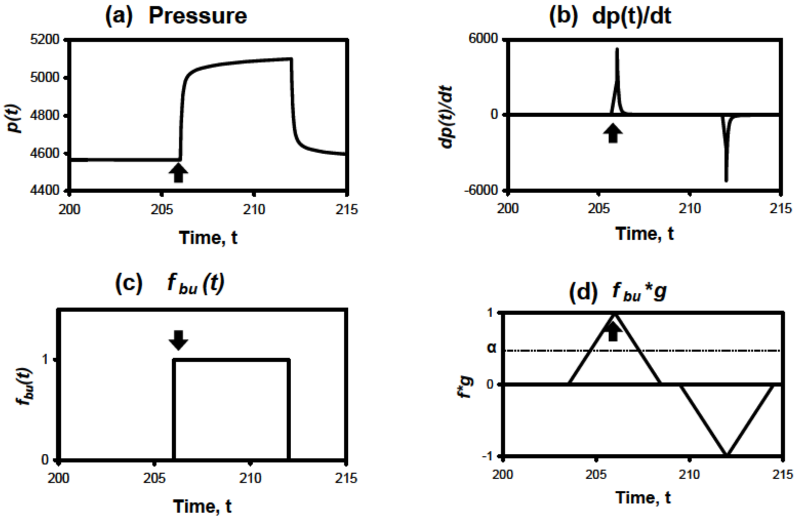


Figure 2.12: Example of buildup detection through filter convolution applied to synthetic pressure data[19].

The main idea of this method is to convolve the pressure derivative indicator function $f_{bu}(t)$ (see Figure 2.12 (c)) or $f_{dd}(t)$ with filter function $g(t)$ (see Figure 2.13), and the peak above a certain threshold α will indicate the position of break point. The indicator functions for build-up pattern and draw-down pattern are defined in Equation 2.3 and Equation 2.4.

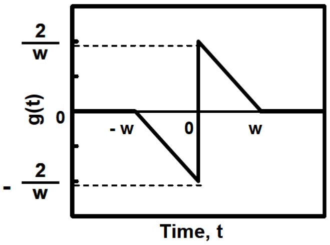


Figure 2.13: Filter function for detecting the start of buildup/drawdown[19].

$$f_{bu}(t) = \begin{cases} 1, & \text{if } \frac{dp}{dt} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

$$f_{dd}(t) = \begin{cases} 1, & \text{if } \frac{dp}{dt} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

Suzuki et. al applied this approach both to oil well and gas well without data denosing. This method achieved a perfect result for oil well, however poorly performed for gas well. After that the author applied moving average filter and dp/dt thresholding to the gas pressure data and finally achieved a satisfactory result.

In 2018, Suzuki improved the performance of this method significantly by replacing the moving average filter with Hampel filter (Hampel, 1971 [20]; Hampel, 1974 [21]) for pre-processing pressure data [22].

2.6 Comparison

The approaches we investigated have different detecting schemes, merits and demerits, different requirements for user inputs/ criteria. Table 2.1 made the comparison.

2.7 Summarization

Figure 2.14 summarized methods discussed in this chapter from another view. These methods can be divided into two categories: (i) method need denosing techniques, (ii) method with no requirement for denosing techniques. All methods, except Segmentation and its variant, require denosing technique for a better performance.

Three denoising techniques utilized by these methods are: moving average, Savitzky-Golay smoothing filter and Hampel filter. In some methods,

Table 2.1: Comparison between different approaches for transients detection

	Main idea	**Pros**	**Cons**	**User input**
Spline wavelet	• Use the concept of wavelet modulus maxima which are suitable for singularity detection to determine the breakpoints.	• Effectively detect significant breakpoints	• Difficult to define universal criteria • Many false break points were detected • Require expert use • Impossible for real time analysis	• Δt_{min} • Slope threshold
Pattern recognition	• When the entire transient image falls in the range of the predefined transient pattern, it will be identified as a transient.	• Most of major transients are correctly detected. • Faulty detection is rare. • Require minimum user interaction.	• Require data preprocessing, e.g. outliers removal & denoising, before identification • Additional requirement for not to use a noise filter which will over smooth the transients • Miss the transients with too small magnitude.	• Predefined pattern
S-G smoothing filter	• First and higher derivatives of the signal, calculated in an efficient way using the S-G filters.	• perform better than the wavelet based approaches. • Screen out false break points in high noisy data.	• Had difficulty in screening smaller transients.	• Window size
Segmentation method	• Reduces the entire dataset into strategic points. • Smaller and false break points were further screened out by using the area, calculated under a transient.	• perform better than the wavelet based approaches. • effective in identifying all the significant break points. • The performance of this method is not sensitive to the value of stopping criteria (tolerance τ).	• Had difficulty in screening smaller transients.	• Tolerance τ
Segmentation method variant	• Identifies the zeros in the flow rate. • Between two consecutive zeros finds the peak using the segmentation approach.	• perform as well as segmentation method. • Easy to implement.	• Both pressure and rate data are required. • Faulty detections are likely to happen when some points have smaller orthogonal distance than breakpoints.	• Tolerance τ
Data smoothing	• Use wavelet processing result as initial guess. • Insert and delete points through iterative procedure, get the final results until the stopping criteria meet.	• An improvement to wavelet decomposition approach. • Data driven and less subjective.	• GCV score and minimization are not guaranteed to produce a good result of detection. • Need wavelet processed results as prerequisite. • The procedure is complex.	• Δt_{min} • Slope threshold
Filter convolution	• Convolve a fliter function with the indicator function of the slope of pressure data. Breakpoints are identified by location the maximum value of the convolution.	• Tolerant to noise to some extent. • Perfect accuracy for oil well. • About 80% accuracy for gas well.	• Difficult to define universal criteria • Need denosing techniques combined to achieve better results • Need 3-5 iteration to determine the slope threshold criteria.	• 2ω • Slope threshold

the noise are filtered further by slope threasholding (Spline wavelet, Data smoothing & Filter convolution). Theoretically, any of these three denoising techniques (puple dashed reactange in Figure 2.14) can be combined with any of the transients identification approaches (orange dashed reactange in Figure 2.14).

To extend further, any method which involves the calculation of derivatives of signal data, need to denoise the data especially when existing severe noise. The first order derivative of pressure data (dp/dt) for some points with severe noise will be as large as the true break point, thus causing the false detection.

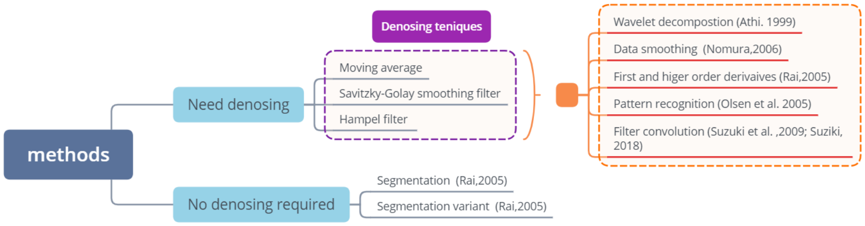

Denosing teniques
methods
Need denosing
Moving average
Savitzky-Golay smoothing filter
Hampel filter
No denosing required
Segmentation (Rai,2005)
Segmentation variant (Rai,2005)
Wavelet decompostion (Athi. 1999)
Data smoothing (Nomura,2006)
First and higer order derivaives (Rai,2005)
Pattern recognition (Olsen et al. 2005)
Filter convolution (Suzuki et al. ,2009; Suziki, 2018)

Figure 2.14: Summarization of methods discussed in section 2-8.

Chapter 3

Methods based on derivative

Derivatives values of pressure measurements are fairly important metrics utilized by many researchers for the transients identification task. However derivative is fairly sensitive to noise. The derivative methods will easily detect false points because of noise if the algorithm is not well designed.

Through careful research as well as experiment based analysis, we design our own algorithms based on derivative.

3.1 Theory description

As illustrated by in Chapter 2.5, for a noise free pressure data, the positions of break points usually locate at where i)the sign of pressure derivative(dp/dt) switch from negative to positive or positive to negative; ii) or the absolute pressure derivative changed from a close to zero value to a relatively large value.

The above two statements are true for synthetic data set which is noise-free. However it is not the case for real field data set which is usually full of noise. We would like to provide two examples to show how derivative of pressure measurements will behave for real field data. Figure 3.1 is the first 60 hours of the testing data that will be used for our computation experiment for

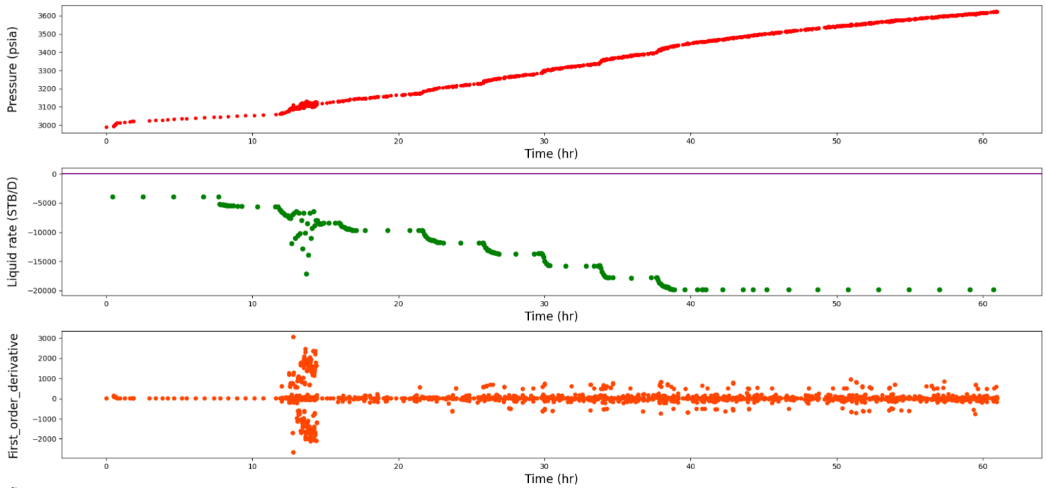


Figure 3.1: A example of pressure response and dp/dt of real field data.

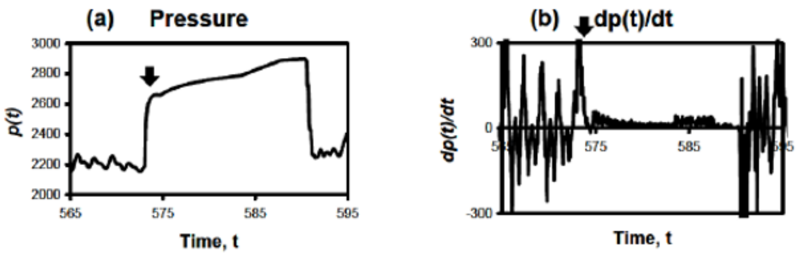


Figure 3.2: A example of pressure response and dp/dt of real field data[19].

this thesis. Figure 3.2 is from Suzuki et. al [19]. From both of these two examples, we will see derivative values fluctuate greatly because of noise. And the change of the derivative's sign seems not to be a good criterion for break points detection. It has also been proved by the computational experiments that we carried out, although the procedure and results of these experiments are not shown in this thesis.

Thus, we try to develop algorithms according to the second statement.

3.2 Algorithm design

3.2.1 Algorithm for RangeFOD

The algorithm according to the second statement can be designed as follows: for a break point, at the left side, the dp/dt is prone to zero, while at the right side, it will have a significant jump. To resist the severe noise further, we will use a window for both sides, applying the criteria to the average value of dp/dt in two windows, as shown in Algorithm 1 (named as **RangeFOD**).

Algorithm 1 RangeFOD

```
$\{p_1,..., p_n\}$                                   ▷ pressure measurements for n points
$dp_i/dt$                                            ▷ first order derivative(FOD) of $p_i$
$std$                                                ▷ standard deviation of $\{dp_1/dt,..., dp_n/dt\}$
$m$                                                  ▷ point window
$\tau_1$                                             ▷ a positive number close to zero
$\tau_2$                                             ▷ tuning paramter, a positive float

while $i <= n - m$ and $i >= m$ do                   ▷ for all valid points in data set
    $\text{avg}_{left} = average(\{\text{dp}_{i-m}/dt,..., dp_i/dt\})$      ▷ average FOD of left m points
    $\text{avg}_{right} = average(\{\text{dp}_i/dt,..., dp_{i+m}/dt\})$     ▷ average FOD of right m points
    if $abs(avg_{left}) < \tau_1$ and $avg_{right} > \tau_2 \times std$ then
        point $i$ is buildup breakpoint
    else if $abs(avg_{left}) < \tau_1$ and $avg_{right} < -\tau_2 \times std$ then
        point $i$ is drawdown breakpoint
    end if
end while
```

3.2.2 Algorithm for DeltaFOD

The experiments based on algorithm 1 show that algorithm 1 works well for the synthetic data. Nevertheless, when it applied to the real field data, the absolute avg_{left} values for true break points sometimes are even larger than the one which is not break points. It is quite difficult to use a threshold of avg_{left} to screen out break points.

Thus, a variation of Algorithm 1 is designed, see Algorithm 2. In which, when the difference between avg_{left} and avg_{right} reach a certain theshold, the point will be determined as buildUp or drawDown.

Algorithm 2 DeltaFOD

```
$\{p_1,..., p_n\}$                                   ▷ pressure measurements for n points
$dp_i/dt$                                           ▷ first order derivative(FOD) of $p_i$
$std$                               ▷ standard deviation of $dp_1/dt,..., dp_n/dt$
$m$                                                                 ▷ point window
$\tau$                                          ▷ tuning paramter, a positive float

while $i <= n - m$ and $i >= m$ do                ▷ for all valid points in data set
    avg$_{left}$ = $average(\{dp_{i-m}/dt,..., dp_i/dt\})$    ▷ average FOD of left m points
    avg$_{right}$ = $average(\{dp_i/dt,..., dp_{i+m}/dt\})$   ▷ average FOD of right m points
    $\Delta FOD_i = avg_{right} - avg_{left}$
    if $\Delta FOD_i >= \tau \times std$ and $avg_{right} > 0$ then
        point $i$ is a buildup breakpoint
    else if $\Delta FOD_i <= -\tau \times std$ and $avg_{right} < 0$ then
        point $i$ is a drawdown breakpoint
    end if
end while
```

3.3 MaxFOD and MinFOD

As we discussed the value of derivative brings the signal of the change. The other two approaches based on the value of derivative are also used to aid the identification task. We name them as **MaxFOD** and **MinFOD**.

MinFOD For points whose absolute first order derivative are smaller than a certain threshold, we consider the derivative is caused by noise other than by the change of flow rate. These points could be removed before inputted into the identification algorithm (e.g Algorithm 2) which consumes a large portion of the running time of whole system.

MaxFOD In a point/time window, the point with maximum absolute first order derivative will be more likely to be a break point than other points in this window.

These two approaches are not used as main detection methods, however they can be used to optimize the detection, as we did in the experiments workflow, which will be discussed later.

Chapter 4

Method based on tangent

4.1 Theory description

Tangent has information which reflects the trend of points. We observe two points in Figure 4.1. For point A, $tangent\_left$ and $tangent\_right$ are very close, then $\Delta tangent$ is relatively small. For point B, $\Delta tangent$ is obviously larger. By setting a reasonable threshold for $\Delta tangent$, a break point could be identified.

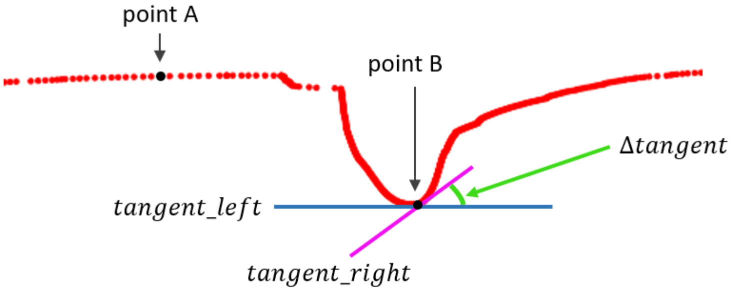


Figure 4.1: Schematic of tangents of pressure measurements. Red points: pressure measurements. Blue line: left tangent of point B. Purple line: right tangent of point B.

To calculate the tangents, we analyse the left m points (left window) and right m points (right window), find two fitted curves for both left window and right window. Then tangent of every points could be calculated, as illustrated in Figure 4.2.

For instance, a three - degree polynomial fitting curve for left window is

$$f(x) = ax^3 + bx^2 + cx + d \tag{4.1}$$

Then,

$$f^{'}(x) = 3ax^2 + 2bx + c \tag{4.2}$$

For a point (x_0, y_0) in left window,

$$tangent = f^{'}(x_0) = 3ax_0^2 + 2bx_0 + c \tag{4.3}$$

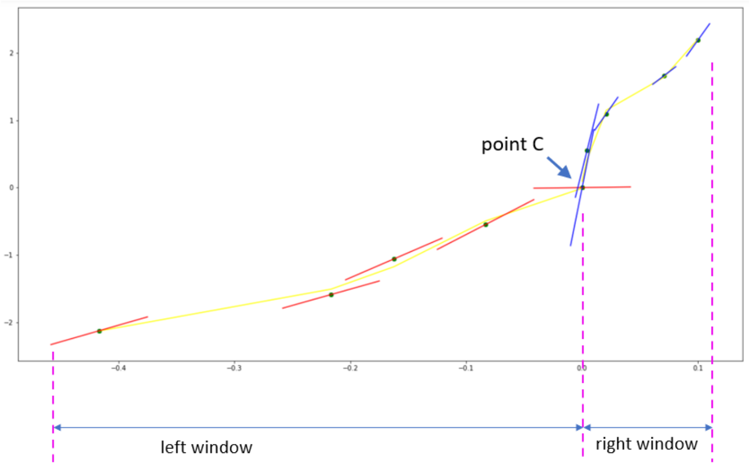


Figure 4.2: A example of tangents calculation for point C. Point window=5. Red lines: tangents for points in left window. Blue lines: tangents for points in right window.Yellow curves: fitting curves.

4.2 Algorithm design

The algorithm designed in this section use $\Delta tangent$ (see Figure 4.1) as a separating criterion. However, because of the noise, similar as Chapter 3.2.1 which use average derivative, we use average tangent instead of a single tangent as a tangent representative of a point.

In Figure 4.2, at Point C, the intersection of blue tangent line and red tangent line is large. If we use this intersection value as the criterion, the Point C will be classified as a break point. In fact, the Point C deviates its route because of noise and should not be detected as a break point. From this observation, we use average of all the tangents in left window (all 5 red lines, when the point window=5) as the left representative tangent of Point C, same for right window. Comparing with the single tangent, the average tangent performs much better, its experiment results will be shown and discussed in Chapter 6.

Algorithm 3 DeltaTangent

```
$\{point_1$,..., $point_n\}$                                   ▷ n points
$t_i$                                                          ▷ tangent of point $i$
$m$                                                            ▷ point window
$\tau$                                                         ▷ threshold, a positive number

while $i <= n - m$ and $i >= m$ do                             ▷ for all valid points in data set
    avg$_{left}$ = $average(\{$t$_{i-m}$,..., $t_i\})$          ▷ average tangent of left m points
    avg$_{right}$ = $average(\{$t$_i^{'}$,..., $t_{i+m}\})$     ▷ average tangent of right m points
    $\Delta t_i = avg_{right} - avg_{left}$
    if $\Delta t_i >= \tau$ and $avg_{right} > 0$ then
        point $i$ is a buildup breakpoint
    else if $\Delta t_i <= -\tau$ and $avg_{right} < 0$ then
        point $i$ is a drawdown breakpoint
    end if
end while
```

Chapter 5

Method based on pattern recognition

5.1 Theory description

In Chapter 2.2, we reviewed a pattern recognition method proposed by Olsen et al.(2005)[10]. As mentioned by Olsen et al., there are two ways to determine the patterns used in this approach.

- To predefine a pattern.
- To use a system learning process to train the system for pattern recognition.

Olsen et al. implemented the first way. The predefined two patterns for transients identification are shown in Figure 2.1 (for build-up transient) and Figure 2.2 (for draw-down transient). To determine whether a point is a break point, they will selected several points both before and after this point. If all these points fall into the shaded region of the pattern image. Then the point is considered as a break point accordingly.

Olsen et al. also utilize some statistics approach to aid the transient identification. For the case of denoised data, if the standard deviation of the points which are selected for identification is less than a factor of *the total signal standard deviation*, then it will not be considered as a transient. For the raw data, use the standard deviation of *signal noise level* instead of *total signal.* It should be noted that, the factors for these two scenarios are not the same.

In this chapter, we design our own way to learn patterns from ground-truth break points. And we will also adjust the criterion for determining the break points for noisy data.

In the following section, we will demonstrate our methods about how to learn patterns and predict using learned patterns in details.

5.2 Implementation

5.2.1 Learn from ground truth

Learning procedure

Unlike Olsen et al. used the predefined patterns, this study designs a learning procedure to learn patterns from the ground truth break points. The two patterns will be learned accordingly from buildup break points and drawdown break points.

Figure 5.1 demonstrated the learning procedure for draw-down pattern, which can be concluded as following steps.

1. For a ground truth drawdown break point, we select left m points(left window) to that point, then find a fitting curve for this window.
2. For n ground truth points, we will get n curves for the left side.
3. Find two borders to contain all these n curves.
4. Repeat step 1 $\sim$ 3 to the right side of each ground truth point. Another two borders will be obtained.
5. These 4 borders that we learned will be the pattern for identification.

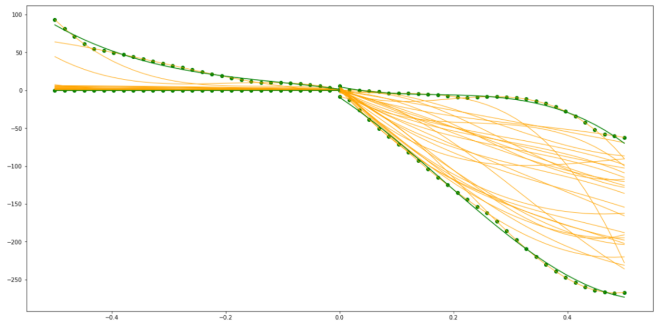


Figure 5.1: Schematic of learning drawdown pattern. Yellow lines represent fitting curves for every ground truth drawdown break points. Green lines represent learned pattern.

The build-up pattern also can be learned from ground-truth build-up break points by repeating above 5 steps.

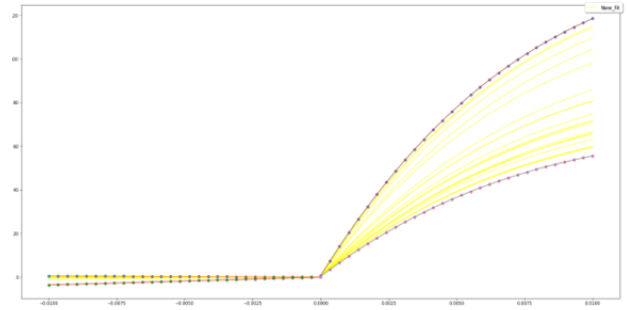


Figure 5.2: Learned buildup pattern from the ground truth of synthetic data.

Learned patterns

We tested the implementation with both synthetic and real case data. Figure 5.2 and Figure 5.3 show the learned patterns from the ground truth of synthetic transients. Figure 5.4 and Figure 5.5 show the learned patterns from the ground truth of real field transients. From these figures, we could observe that the slope of left borders of the pattern learned from synthetic data are more prone to zero. This could be easy to comprehend since synthetic pressure measurements is noise free.

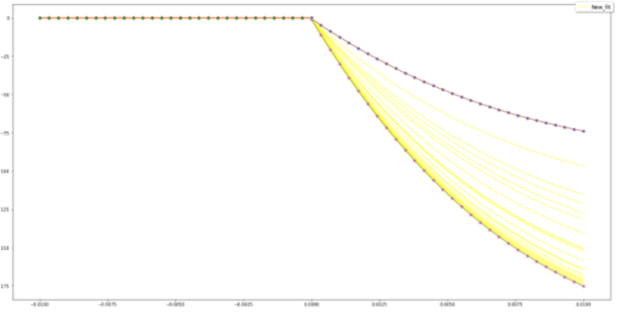


Figure 5.3: Learned drawdown pattern from the ground truth of synthetic data.

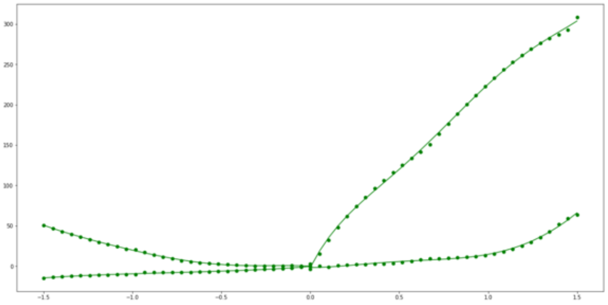


Figure 5.4: Learned buildup pattern from the ground truth of real field data.

5.2.2 Predict break points

In Olsen et al.'s paper, only when all points in left window and right window of a certain point fall into the pattern image, then that point will be classified as a break point. This criterion works well in many cases. However we had observed some false detection with this criterion when testing with our real field data set.

In Figure 5.6, points, annotated by green ellipse, do not fall into the drawdown pattern. However this point should be classified as a draw-down break point. Thus, in order to tolerant the imperfect learned pattern and the noise, we adjust the criterion as: if more than 80% of the total points fall into the pattern region, then it will be identified as a break point.

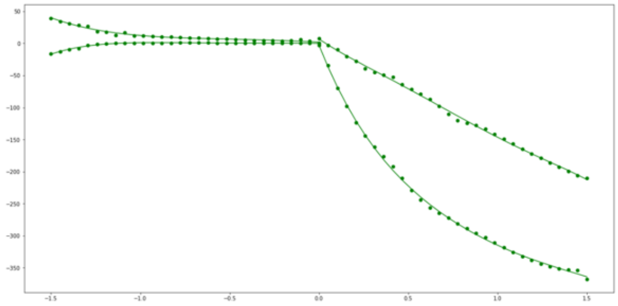


Figure 5.5: Learned drawdown pattern from the ground truth of real field data.

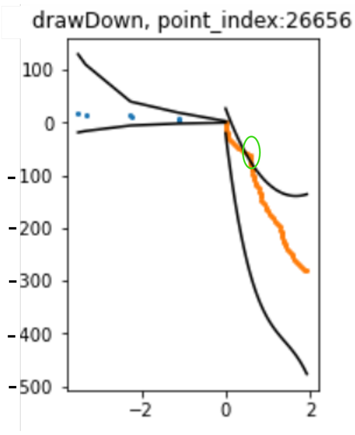


Figure 5.6: Point 26656 should classified as drawdown break point. Although the points, annotated by green ellipse, do not fall into the pattern.

5.3 Improve Patterns

The patterns we learned will affect the accuracy of detection dramatically. It requires us to improve the patterns as much as we can.

During the implementation and experiments, we observe some possible reasons which might compromise the learned patterns. Two main reasons will be discussed here.

- During the learning process, if there are many outliers and severe noise in the learning window, then it will increase the learned pattern area.

Thus, many minor changes or even false points will be detected.

- The fitting curve method used in the learning procedure has a nature to smooth data, thus some subtle information are lost during the learning process.

We designed two schemes to address the above problems, named as **Percentile Tuning** and **Fine Tuning**, as illustrated in Figure 5.7.

Percentile Tuning Percentile is a statistics concept. In learning process, we could choose to only learn from curves which are in a certain range of percentile, e.g., 10 $\sim$ 90 percentile. The curves below the lower bound and above the upper bound, which are quite likely the outliers and points caused by severe noise, will be removed and not be learnt. The two yellow curves pointed by blue arrows (in Figure 5.7) are removed by percentile tuning.

Fine Tuning To compensate the information lost by the fitted curve, we could multiply the pressure measurements of learning points with a factor (larger than 1) when calculating the borders. Then the pattern area learned will be enlarged, indicated by the red arrows (in Figure 5.7).

Learn from more ground-truth points Besides tuning the learned pattern, another effective way to improve the patterns is to learn from more ground truth points from different data sets. However, to obtain the ground truth requires quite some experts' work.

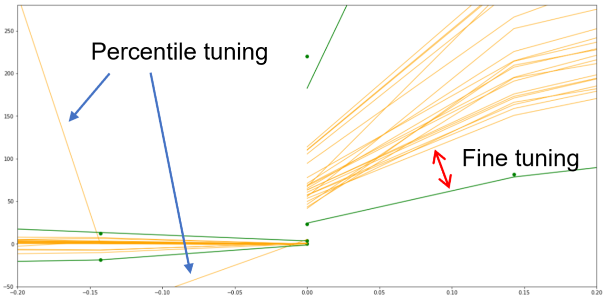


Figure 5.7: A example of zoom-in plot for the learned pattern after applying the percentile tuning and fine tuning. Green lines are learned borders, yellow lines are fitted curves of ground truth transient.

Chapter 6

Computational experiments and discussion

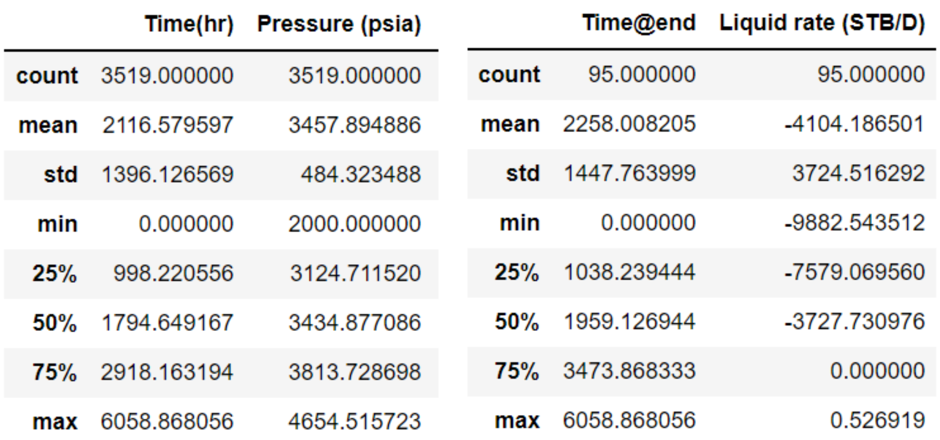
6.1 Experiment setup

6.1.1 Load & analyze data sets

There are two data sets for testing in this study, one is synthetic (noise free) data, the other is real field data from Shchipanov et al.(2017)[23]. The synthetic data is used to test the implementation is correct or not. The real case data is used to adjust the implementation to tolerate noise.

Figure 6.1 and 6.2 give an overview of synthetic and real field data used in this study. Both synthetic and real field data have pressure data & flow rate data that are not evenly distributed in time axis. And in some period, the flow rate data are absent.

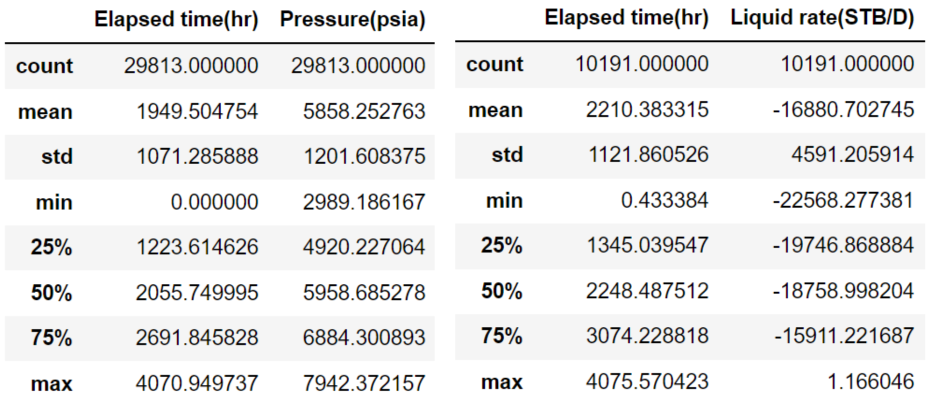
One thing should be noted is that although both pressure data & flow rate data are inputted into the model, only pressure data are used to identify the break points. The flow rate are also plotted as a reference for the positions of the break points. However the break points positions that flow rate measurements indicate is not guaranteed to be ground-truth due to the



	Time(hr)	Pressure (psia)
count	3519.000000	3519.000000
mean	2116.579597	3457.894886
std	1396.126569	484.323488
min	0.000000	2000.000000
25%	998.220556	3124.711520
50%	1794.649167	3434.877086
75%	2918.163194	3813.728698
max	6058.868056	4654.515723

	Time@end	Liquid rate (STB/D)
count	95.000000	95.000000
mean	2258.008205	-4104.186501
std	1447.763999	3724.516292
min	0.000000	-9882.543512
25%	1038.239444	-7579.069560
50%	1959.126944	-3727.730976
75%	3473.868333	0.000000
max	6058.868056	0.526919

Figure 6.1: Overview of synthetic data used in this study.



	Elapsed time(hr)	Pressure(psia)
count	29813.000000	29813.000000
mean	1949.504754	5858.252763
std	1071.285888	1201.608375
min	0.000000	2989.186167
25%	1223.614626	4920.227064
50%	2055.749995	5958.685278
75%	2691.845828	6884.300893
max	4070.949737	7942.372157

	Elapsed time(hr)	Liquid rate(STB/D)
count	10191.000000	10191.000000
mean	2210.383315	-16880.702745
std	1121.860526	4591.205914
min	0.433384	-22568.277381
25%	1345.039547	-19746.868884
50%	2248.487512	-18758.998204
75%	3074.228818	-15911.221687
max	4075.570423	1.166046

Figure 6.2: Overview of real field data used in this study.

asynchronization between the flow rate data and pressure data.

Synthetic data The pressure data have 3519 points, while rate data have much less points, only 95 points. The data set last roughly 6059 hours.

Real field data The pressure data have 29813 points, while rate data have much less points, 10191 points only. The time duration is roughly 4076 hours.

While all synthetic data and real field data are loaded and tested with our model, in the following chapters, we will only display and discuss the results produced by the real field data provided by Shchipanov et al. because of the limitation of chapters. Since the model targets to deal with real case data eventually.

6.1.2 The manual interpreted break points

The knowledge of the true break points in a data set is a prerequisite to verify the implemented methods. In our testing real field data, we consider the break points manual interpreted by Shchipanov et al.(2017)[23] as the ground truth. By comparing the detected results with these manual interpreted break points, we can reach a conclusion which method achieve the highest accuracy for the testing data sets.

As demonstrated in Chapter 1.2.3, the transients identification problem is modelled as two sequential tasks, denoted as *Task 1* and *Task 2*. In *Task 1*, which is going to separate the flowing periods and shut-in periods, we get 31 flowing periods and 30 shut-in periods through the manual interpretation. The detailed plot is attached as Appendix B[1]. However, in *Task 2*, the criterion of the major multi-rate break points in flowing periods varies in different user cases, depending on the system requirements of sensitivity to the change of pressure measurements. The number of multi-rate break

[1] This appendix is a sliced version. The link of the full version is provided in Appendix F.

points is not fixed. The statistics for the manual interpreted break points is listed in Table 6.1.

Table 6.1: The manual break points of real field data set provided by Shchipanov et al.

	Real field data	
	Flowing	Shut-in
Task 1	31	30
	Multi-rate break points	
Task 2	varied	

Notes: The break points identification in shut-in period is not in the scope of this study.

6.2 The workflow

We design a workflow for the two tasks of this study, see Figure 6.3. Raw pressure data are inputted into the model, then the denoised pressure measurements go through the coarse Filtering module which removes the points that are quite likely not break points. The remaining points will be inputted into methods pipeline and clustering representing modules. The methods we developed(**DeltaTangent**, **DeltaFOD**, **PatternRecogntion**) will be used individually or in a combination in the methods pipeline module. After that, all detected break points candidates will be handled by *Task 1* processing module, the flowing periods and shut-in periods will be screen out in this step. Both breakpoints candidates and flowing & shut-in periods are inputted into *Task 2* processing module, major break points in all flowing periods will be obtained.

6.2.1 Denoising

Denoising techniques play an crucial role for transient identification utilizing the derivative of pressure measurements, as discussed in Chapter 2.7. For many other methods, it will also help to improve the performance.

Our real field testing data set is fairly noisy especially at the beginning. The

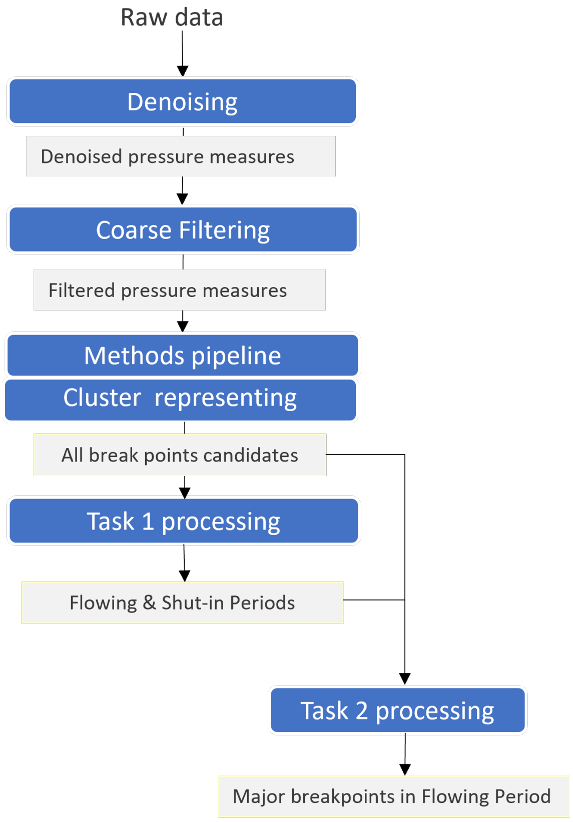

Raw data
Denoising
Denoised pressure measures
Coarse Filtering
Filtered pressure measures
Methods pipeline
Cluster representing
All break points candidates
Task 1 processing
Flowing & Shut-in Periods
Task 2 processing
Major breakpoints in Flowing Period

Figure 6.3: Workflow for transients identification model.

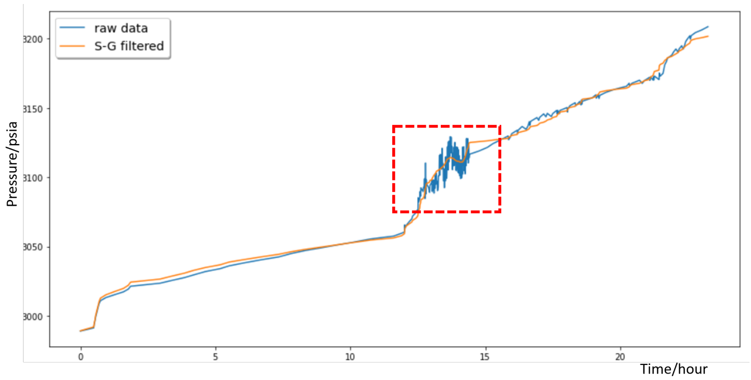


Figure 6.4: The pressure response of first 25 hours of raw data set from Shchipanov et al.(2017)[23] and denoised with Savitzky-Golay FIR smoothing filters.

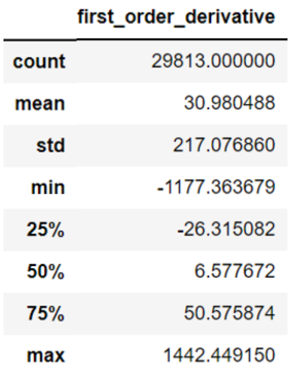
pressure measurements in the red dashed rectangle (Figure 6.4) fluctuated greatly because of noise. The pressure response is smoothed after being denoised as the gold curve indicates (Figure 6.4). The Savitzky-Golay FIR filters are chosen in this study to perform the denosing.

Comparing of the raw and denoised / filtered pressure data sets revealed some periods, where pressure data were over-smoothed, resulted in shift of break-points causing potential issues for further break-point identification. It was however decided to use these denoised / filtered pressure dataset in further testing of the transient identification methods to study their performance in a more complicated environment.

6.2.2 Coarse filtering

As we discussed in Chapter 3, derivatives contain the signal that how the pressure measurements change, which are caused either by noise, or by the change of flow rate measures, or because of both of them. Large absolute value of derivatives usually indicate the sharp change of pressure measurements. Although for the whole data set, the *min* value is smaller than -1000, and *max* value is approximately 1500 (see Figure 6.5), we observe that a large fraction of first order derivative are close to zero (see Figure 6.6), which is likely to be caused by noise. For this reason, filtering the points by setting a derivative threshold is a generic approach utilized by many researchers in the break points identification problem. The demerit is that the threshold value is varied for different data sets and requires experimental trials to determine a reasonable value.

In this study, we use the statistics percentile to filter the data set. The value of percentile is much more universal comparing with threshold values. Since we are not interested in the multi-rate breakpoints in shut-in period, in which the derivatives are usually negative, we keep most of points with positive derivatives but remove most of the points with negative derivatives. We set the upper bound of percentile to be 50, and lower bound to be 10. Thus, the number of points flowing to next module are reduced by 40%. The executing time for one run of the whole system will be decrease as well. The reason why we set 50,10 as the upper and lower bound for pencentile will be further explained in Chapter 6.5.1



	first_order_derivative
count	29813.000000
mean	30.980488
std	217.076860
min	-1177.363679
25%	-26.315082
50%	6.577672
75%	50.575874
max	1442.449150

Figure 6.5: Statistics of first order derivative of denoised pressure measurements of Real field data.

6.2.3 Detect all break points candidates

This step contains two sub-modules: i) Methods pipeline; ii) Cluster representing.

Methods pipeline Methods pipeline can be a single method or a combination of multiple methods. The methods that could be used in this sub-

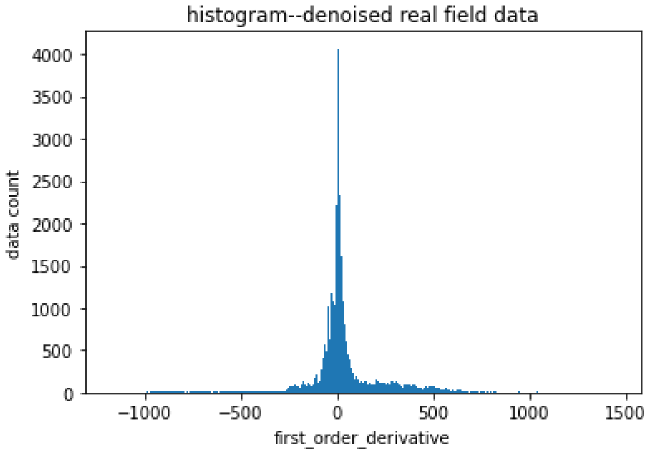


Figure 6.6: Histogram of first order derivative of denoised pressure measurements of Real field data.

module include: i) **DeltaTangent**, ii) **DeltaFOD**, iii) **PatternRecogntion**. After the denoised and filtered pressure measurements flow through this sub-module, all possible break points are detected, labelled as "buildUp" or "drawDown".

Cluster Representing For methods, especially those use the average algorithm, (e.g. average derivative, average tangent, etc.), the neighbour points of the true break points will also be detected since they are likely to have similar average values. Thus, it is necessary to find a representative in the cluster which contains both detected neighbouring points and true break points. And we want the representatives are close to true break points as much as possible.

We explored two strategies for finding such representative: i) choose the one with maximum FOD (first order derivatives). ii) choose the first point of this cluster. We choose the second strategy, since it meets derivative patterns for a large portion of break points, which will be further discussed in Chapter 6.5. The experiment results also verify this statement.

6.2.4 Produce results for *Task 1*

Task 1 processing module targets screening out all flowing periods and shut-in periods. We design a two-step procedure as follows.

1. Screen out all possible flowing periods and shut-in periods.

   Figure 6.7 shows a segmentation of detected break points candidates, which is a most common scenario when the pressure measurements change from flowing period to shut-in period.

   Based on the observation of this kind of scenario, we search the start point of flowing period and shut-in period in the following way:

   - If there are multiple "drawDown"s between two adjacent "buildUp"s, the one with minimum negative derivative (indicated by blue ellipse in Figure 6.7) is determined to be start point of a shut-in.
   - The "buildUp" break point right after this chosen "drawDown" (indicated by purple ellipse in Figure 6.7) is labelled as start point of next flowing period.

2. Remove the minor periods.

   Between the periods distinguished by step 1, there maybe exists some minor ones which need to be removed. Three examples are shown in Figure 6.8. Those in purple ellipses are minor transients which should be removed. We set a condition shown as Equation 6.1. The period meets this condition will be determined as a minor transient and removed.

   $$std_{transient} \leq \alpha \times std_{dataset} \tag{6.1}$$

   where, $std_{transient}$ is the standard deviation of pressure measurements of a certain period/transient; α is a positive number smaller than 1; $std_{dataset}$ is the standard deviation of pressure measurements of the whole data set.

6.2.5 Produce results for *Task 2*

As shown in Figure 6.3, both all detected break point candidates and determined periods are inputted into this module. This module uses a same

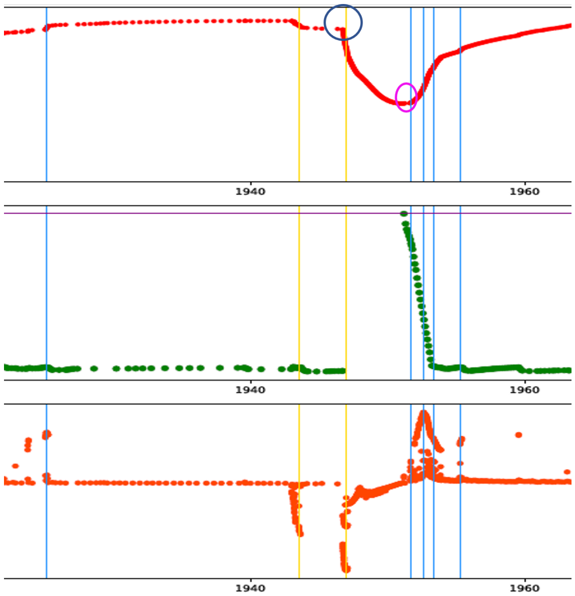
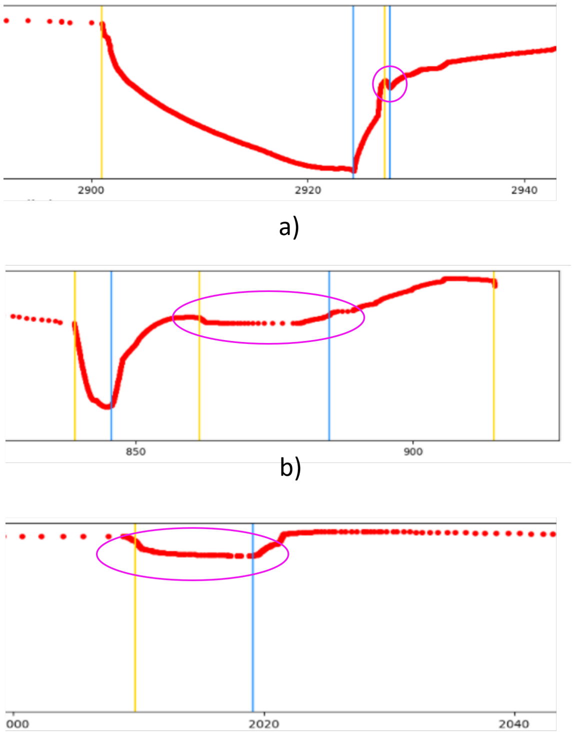


Figure 6.7: A segmentation of detected break points candidates. Top figure: pressure response. Middle figure: flow rate measurements. Bottom figure: first order derivative of pressure response. The point in blue ellipse should be chosen as start break point for a shut-in. The point in purple ellipse should be chosen as start break point for a flowing period.

criterion to remove the minor transients as *Task 1* except the value of the tuning number α is different. The corresponding steps are listed as below.

1. For a certain flowing period, group all the break points that fall into that period, see Figure 6.9.
2. For all transients in the group obtained from step 1, remove the minor one, the same condition (see Equation 6.1) is used.
3. Repeat step 1 and step 2 for every flowing period.



a)

b)

c)

Figure 6.8: a), b), c): Three segmentations of pressure measurements with their detected shut-in periods. Those in purple ellipses are minor transients which should be removed.

6.3 Discussion of experiment results

6.3.1 Experiment results

We carried out a great number of experiments and listed some of the results in Table 6.2.

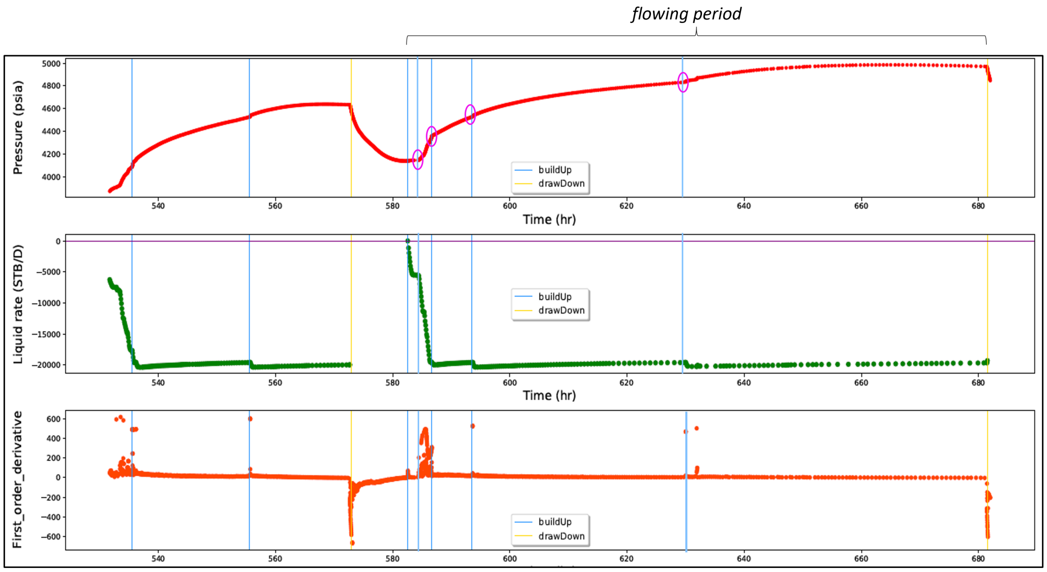


Figure 6.9: A example of a group of break points in a flowing period, annotated by purple ellipses.

6.3.2 The performance of three methods for two tasks

In this section, we will discuss the experiments results both for *Task 1* and *Task 2*. It should be noted here that, when we discuss the results for *Task 1*, only shut-in periods are discussed in the following chapters. Since the start point of a shut-in is the end point of a flowing period, the end point of a shut-in is the start point of a flowing period. Once the shut-in periods are detected, the flowing periods are detected as well. It is redundant work to discuss all of them. Thus in Table 6.2 we only list the numbers of detected shut-in periods, due to the limitation of space. And the detailed plots for the best results of *Task 1* and *Task 2* are attached as Appendix C and Appendix D respectively. It should be noted that both these two appendices are sliced versions, the links for the full versions are provided in Appendix F.

Task 1 The number of manual interpreted shut-in periods is 30. However, for real case data, sometimes it might be difficult to get rid of all false detections.

Table 6.2: The results of experiments

Methods	**User Input**			**BreakPoints Candidate**		***Task 1***		***Task 2***	
PR$^{\alpha}$	Point Window	Learned Pattern		BuildUp	Draw-Down	ShutIn Thresh-old	Filtered ShutIns	Multi-rate Threshold	Multi-rate Points
	15	border$^{\gamma}$ coefficients		578	185	0.02	31	0.03	119
DeltaFOD	Point Window	Tuning parameter		BuildUp	Draw-Down	ShutIn Thresh-old	Filtered ShutIns	Multi-rate Threshold	Multi-rate Points
	15	0.1		755	96	0.02	31$^{\delta}$	0.03	171
DeltaFOD + DeltaTan$^{\beta}$	Point Window	Tuning parame-ter	Tangent Thresh-old	BuildUp	Draw-Down	ShutIn Thresh-old	Filtered ShutIns	Multi-rate Threshold	Multi-rate Points
	15	0.1	20	327	69	0.02	31	0.03	117
DeltaTan	Point Window	Polyno-mial Order	Tangent Thresh-old	BuildUp	Draw-Down	ShutIn Thresh-old	Filtered ShutIns	Multi-rate Threshold	Multi-rate Points
	10	1	20	392	87	0.02	31	0.03	142$^{\epsilon}$
		3		544	102	0.028	31	0.03	159
		1	30	327	76	0.02	31	0.03	127
		3		465	88	0.028	31	0.03	152
		1	40	273	70	0.02	31	0.03	118
		3		402	83	0.02	33	0.03	141

$^{\alpha}$ PatternRecognition. $^{\beta}$ DeltaTangent. $^{\gamma}$ See Appendix A. $^{\delta}$ Best result for *Task 1*, see Appendix C.
$^{\epsilon}$ Best result for *Task 2*, see Appendix D.
Pink colored rows indicate successful detections of all shut-ins.

The results colored with pink(see Table 6.2) contain same 31 transients although the positions of detected points might have tiny deviations from those manual interpreted points. The overview plot is shown as Figure 6.10a). Comparing with the manual interpreted results, we would see that the period annotated by purple ellipse is the one which is false detected, a zoom-in plot is shown as Figure 6.10c).

However, we would like to argue that for real case data, this kind of false detection sometimes is difficult to avoid. Green ellipse in Figure 6.10 annotate a correctly detected period, a zoom-in plot is shown as Figure 6.10b). Comparing b) and c), the standard deviation and decreased pressure measures of c) is even large than b), if we want to remove c), the correctly detected period b) will be removed also. Thus, even through we detected one more period than the manual interpreted results, we still consider the detection achieve a good performance.

The shut-ins in uncolored rows contain some false transients and miss some true break points even though some of them still detect 31 shut-ins.

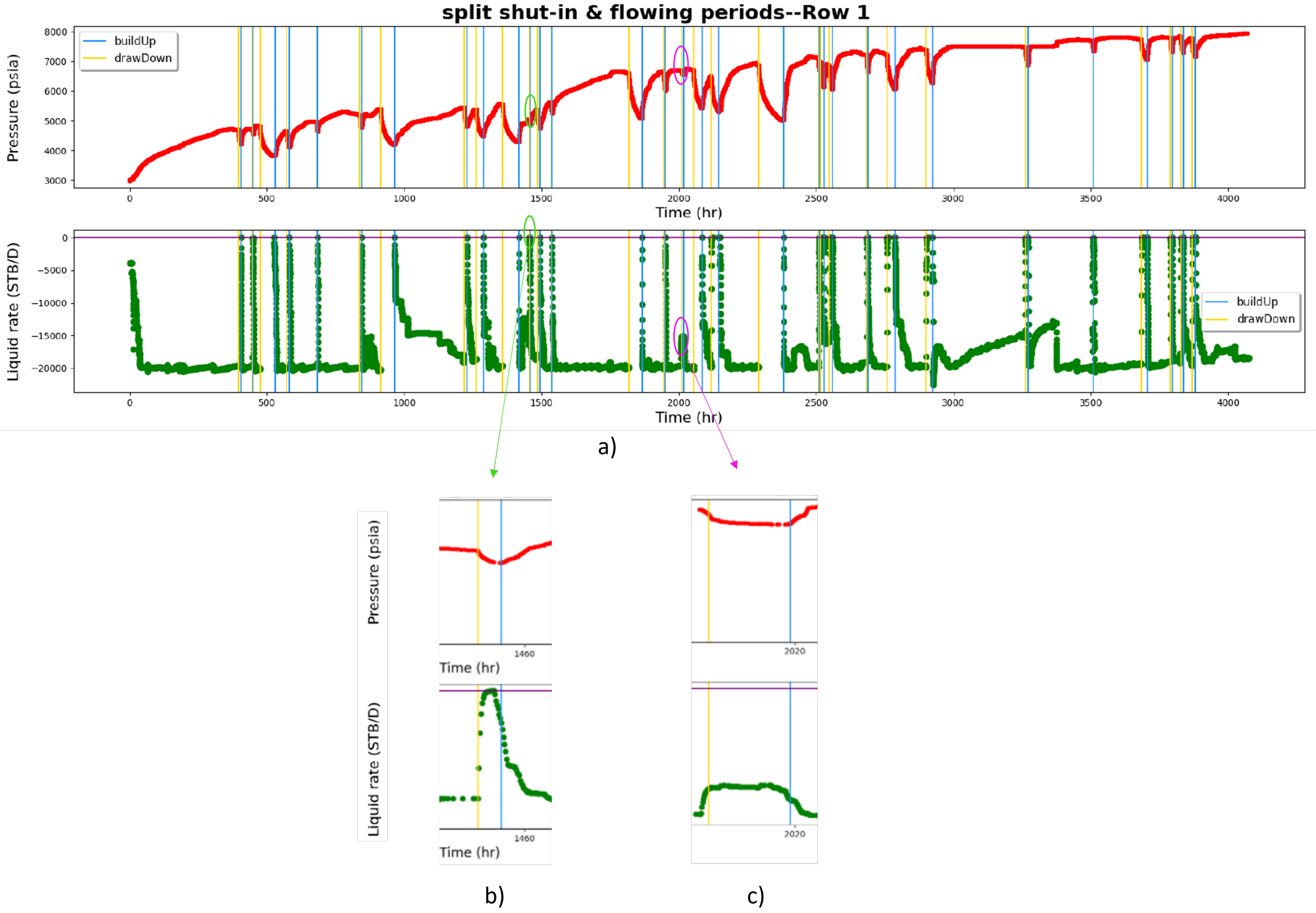


Figure 6.10: a) Overview plot of result for *Task 1*; b) Period correctly identified; c) Period falsely identified.

Among these correct identifications, **DeltaFOD** (purple cell in Table 6.2) has the best result, whose detected positions deviate least from the positions of manual interpreted break points. The position deviation will be illustrated in details in Chapter 6.3.3.

Task 2 In this task, **DeltaTangent**(green cell in Table 6.2) performs the best since its detected positions have least drift with an overall evaluation, while **DeltaFOD** has worst performance.

6.3.3 The deviation of detected break point's position

As we illustrated in last section, our detected positions of break points might deviate a little bit from true break points.

Through analysing the whole detecting procedure, we conclude two possible reasons for the position deviations.

The first reason is the average algorithm used in our methods and *Cluster representing* sub-module of the workflow. The neighbour points will have similar values as a consequence of the average algorithm. Thus we need to elect only one representative between these neighbour points without taking all of them as break points. Otherwise too many points will be detected at almost the same position.

Cluster representing sub-module are designed to do this job. As we discussed in Chapter 6.2.3, we choose the first point of a cluster as the representative. A large portion of true points could be selected in this way, however there still exists some points that do not fall into this pattern. When the first point of a cluster is not a true point, the deviation is derived in this way.

The second reason is the *Task 1 processing* module. In this step, we need to choose one draw-down point out of multiple draw-down points as a start point of shut-in period. The deviation will be produced if we do not select the correct one.

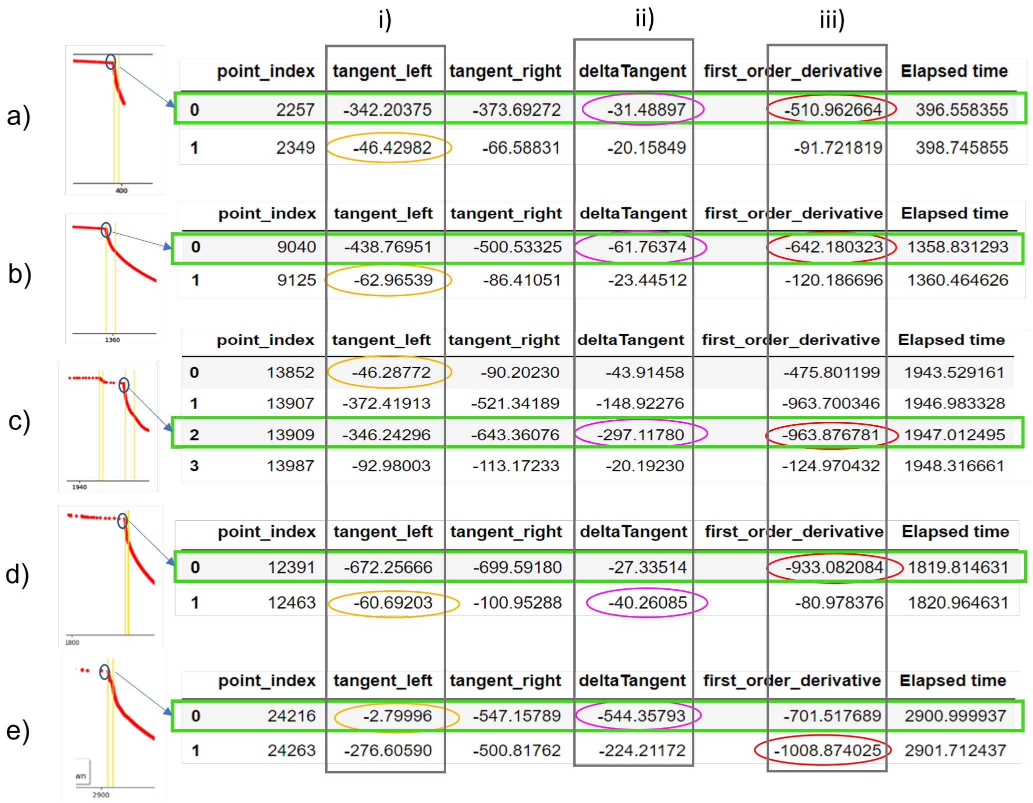
Figure 6.11 shows five segmentations a) $\sim$ e), which are the pressure measurements input of the *Task 1 processing* module. For each segmentation, only one point should be seleted as the start point of a shut-in after processed by *Task 1 processing* module. In segmentaion a), b), d), e), the first point should be identified as start point of a shut-in, while in segmentation c), it should be the third point. The left side of each segmentaion show the pressure measurements plot. The right side shows the value of different metrics for detected points.

We experiment with three criteria to choose the correct start points for shut-in: i) *Minimum Derivative*, ii) *Minimum Delta Tangent*, iii) *Minimum of absolute value of Left Tangent*, corresponding black i) $\sim$ iii) rectangles in Figure 6.11.

In Figure 6.11, green rectangles represent the point that should be chosen. The ellipses in yellow, purple and red color annotate the actual points selected by three different criteria. If the ellipse overlaps the rectangle, then it is a correct identification.

It could be seen from Figure 6.11, if we use the *Minimum of absolute value of Left Tangent*(black rectangle i) as criterion, only segmentation e) is correctly detected. If we use *Minimum Delta Tangent*(black rectangle ii), all segmentations except d) are correctly detected. For *Minimum Derivative*(black rectangle iii), all segmentations except e) are correctly detected. Thus, we could draw an conclusion that: no matter which criterion we choose, the deviation is inevitable.

Through a number of experiments with the whole real field data set, *Minimum Derivative* shows itself as a best criterion.


i)
ii)
iii)
a)
b)
c)
d)
e)
400
1360
1940
1800
2900

	point_index	tangent_left	tangent_right	deltaTangent	first_order_derivative	Elapsed time
0	2257	-342.20375	-373.69272	-31.48897	-510.962664	396.558355
1	2349	-46.42982	-66.58831	-20.15849	-91.721819	398.745855

	point_index	tangent_left	tangent_right	deltaTangent	first_order_derivative	Elapsed time
0	9040	-438.76951	-500.53325	-61.76374	-642.180323	1358.831293
1	9125	-62.96539	-86.41051	-23.44512	-120.186696	1360.464626

	point_index	tangent_left	tangent_right	deltaTangent	first_order_derivative	Elapsed time
0	13852	-46.28772	-90.20230	-43.91458	-475.801199	1943.529161
1	13907	-372.41913	-521.34189	-148.92276	-963.700346	1946.983328
2	13909	-346.24296	-643.36076	-297.11780	-963.876781	1947.012495
3	13987	-92.98003	-113.17233	-20.19230	-124.970432	1948.316661

	point_index	tangent_left	tangent_right	deltaTangent	first_order_derivative	Elapsed time
0	12391	-672.25666	-699.59180	-27.33514	-933.082084	1819.814631
1	12463	-60.69203	-100.95288	-40.26085	-80.978376	1820.964631

	point_index	tangent_left	tangent_right	deltaTangent	first_order_derivative	Elapsed time
0	24216	-2.79996	-547.15789	-544.35793	-701.517689	2900.999937
1	24263	-276.60590	-500.81762	-224.21172	-1008.874025	2901.712437

Figure 6.11: a) ∼ e): segmentations of draw-down points detected by the third step of workflow. i) Use criterion *Minimum of absolute value of Left Tangent*; ii) Use criterion *Minimum Delta Tangent*; iii) Use criterion *Minimum Derivative*. Green rectangles represent the point that should be chosen. The ellipses in yellow, purple and red color indicate that the actual points selected by corresponding criterion. *The overlap of rectangle and ellipse means a correct identification.*

6.4 Discussion of methods

In this section, we will discuss the methods in the following five aspects.

- Algorithms
- Accuracy

- Tolerance
- User input/parameters
- Model running time

6.4.1 Two underlying principles for transients identification's algorithms developed in this study

- To observe data points in windows

  In this study, we experimented various algorithms. At first, only two adjacent points are analysed, the experiment results deviated a great deal from the manual interpreted break points. After that, we try to analyze a certain point in a window. We observed that the identification will be more accurate when both left window and right window of a certain point are taken into consideration. A sudden jump of a certain metrics (e.g. derivative, tangent, etc) between two adjacent points may be caused by noise. The sudden jump of two windows will be more reliable to indicate a real break point.

- To take the average of the windows

  The average algorithm have a good performance to resist the changes caused by noise. We have ever implemented two algorithms for tangent methods, one is to use the left tangent and right tangent of a certain point, the other is to use the average left tangent and average right tangent of a certain point. The former detected many false points, thus was discarded. The latter achieve a good result, see Algorithm 3.

  The demerit is that the positions of the detected points might drift from the manual interpreted break points. The reason is that the neighbour points will have similar values as a consequence of the average algorithm, thus will be elected as break points. In the *Cluster representing* sub-module of the workflow, one point out of these neighbour points should be selected as a representative. The drifts are derived if the selection is not correct.

  For our testing data set, the detected positions of violet cell and green cell(see Table 6.2) drift least from the manual interpreted break points, thus be considered as the best results.

> In Table 6.2, we experiment with *Point Window*. The left n points and right n points are extracted. However, for a severely uneven distributed data set, it might be better to use *Time Window*: points in left time interval and right time interval will be analysed. Both point window and time window are implemented in our deployed web app which will be introduced in Chapter 7.

6.4.2 Accuracy

With the 5-steps workflow, all the methods that we implemented detected these 31 "shut-in"s with reasonable parameters and thresholds, and also performed good for identifying the multi-rate transients.

For *Task 1*, **DeltaFOD** obtains the best result(violet cell in Table 6.2) : the positions of the detected break points have smallest deviation from the manual interpreted break points among these methods.

For *Task 2*, **DeltaTangent** obtains the best result(green cell in Table 6.2) : detected most multi-rate transients in flowing periods among these methods.

The definition of "best result" are demonstrated in Chapter 6.4.1.

6.4.3 Tolerance

This 5-steps workflow tolerates the threshold setting, especially when they are set at low level. The low threshold will increase the number of break points candidate, however the minor transients will be removed by *Task 1* and *Task 2* modules. For instance, the **DeltaFOD** detected 755 buildUps, *Task 1* only selected 31 of them. *Task 2* only selected roughly 1/5 of the them.

Many trials and errors will be required to determine a proper threshold when using the methods alone. Only a couple of times iterations needed when using this workflow.

When using the **PatternRecognition** method, the *learned pattern* normally have a significant influence to the detected results. With this 5-steps workflow, the imperfect patterns still obtain relatively good results.

6.4.4 User input

Different methods require different user inputs. **PatternRecognition** requires least user inputs. For a certain method, if we increase the number of user inputs/criterion, we may get more precise results at the cost of reducing the automation of the model and increase the complexity of the implementation. Thus trade-offs should be made.

Among these user inputs, *learned pattern*, *Point Window* and *Polynomial Order* are universal ones that the values normally do not need to be changed for different data sets.

Point Window is 10 or 15 will not make big difference to the results for our test data set. As a rule of thumb, when the size of point window decrease, the sensitive to minor changes rise; when the size of point window increase, the deviation of the position for the detected points from the manual points rise, as well as the resistance to the noise.

Polynomial Order is user input for **DeltaTangent**, which defines the highest order for polynomial fitting, see Equation 4.1. In our user case, a reasonable setting for *Polynomial Order* is an positive integer no larger than 3. *Polynomial Order* 1 is the same as the linear regression which has a good resistance to the noise, while *Polynomial Order* 3 is more sensitive to minor changes of the pressure measures, as well as the threshold. As shown in the Table 6.2, for the same *Tangent Threshold* 30, *Polynomial Order* 1 still identified the correct shut-ins. We would recommend using *Polynomial Order* 1 for most cases.

Tuning Parameter is used to define the threshold for **DeltaFOD**, the smaller theshold indicates the more break points candidates detected.

Shut-in Threshold & *Multi-rate Threshold* are parameters used in all the methods. The former is used to screen out the shut-ins. The latter is used to remove minor transients in flowing period. The module could detect more multi-rate breakpoints by reducing this threshold. A example is shown in

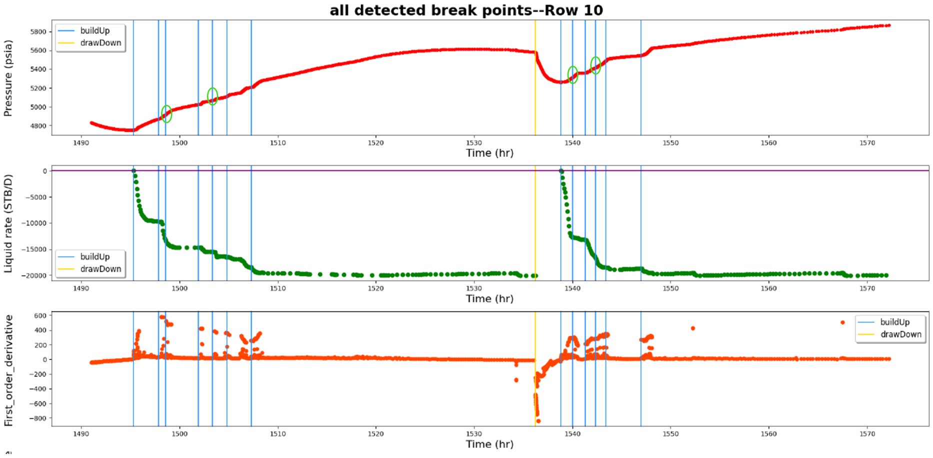


Figure 6.12: *Multi-rate Threshold*=0.01, all blue lines will be detected. *Multi-rate Threshold*=0.02, all blue lines except those in green ellipses will be detected.

Figure 6.12. When we set 0.01 and 0.02 as *Multi-rate Threshold*, the former one will detect 4 more points than the latter.

6.4.5 Model running time

The algorithms that we use require the extractions of the pressure measures in the left window and right window, which is a good approach to resist noise. The disadvantage is that the amount of computation has been greatly increased. For instance, when *Point Window* is 10, the amount of computation will be increased $(10 + 10 - 1)$ times.

PatternRecognition comsumes more time than other two methods, while **DeltaFOD** consumes the least. The running time for **DeltaTangent** increases when the *Polynomial Order* rises.

6.5 Discussion of patterns for break points

To design a good algorithm for our task, we need to have a good understanding about which kind of "patterns" represent a break point. Through

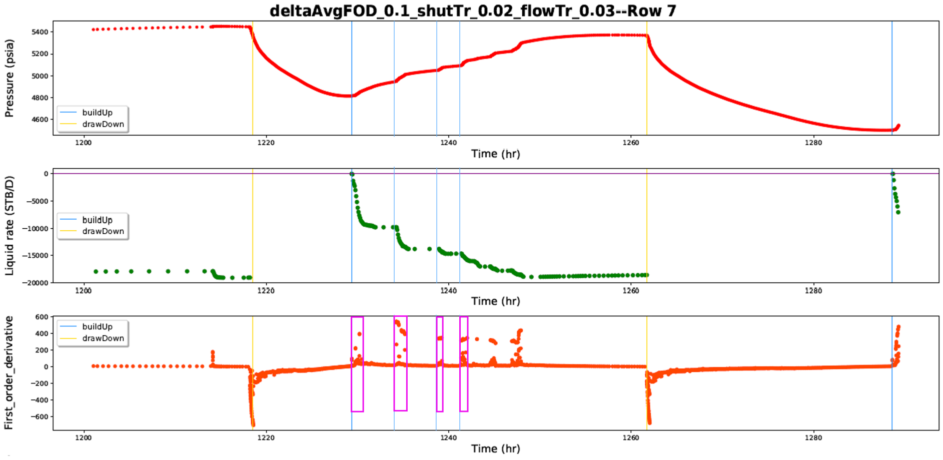


Figure 6.13: A segmentation of testing data set with detected breakpoints.

errors and trials, we have more and more insights about these "patterns", thus enable us keep improving our algorithms and workflow. We feel it is worthy to discuss them using an independent section.

It should be noted that the word "pattern" here does not have same meaning with the word "pattern" in Chapter 5 about the method of **PatternRecognition**, in which the "pattern" is similar to "shape". The "pattern" here is more similar to "feature". The features discussed in this section could provide some hints for a feature-based machine learning methods which is not able to be implemented in this study due to time limitation.

In this section, we try to use statistics approaches to reveal some "facts" about the "patterns" of the true break points based on derivative and tangent. The value of tangent and derivative in this section are based on our real field testing data. The tangent calculation use *Polynomial Order* 1 and *Point Window* 10.

6.5.1 Pattern for first order derivative

Hints from derivative plot

By observing the derivative plot, we could get some hints about the derivative patterns. As shown in Figure 6.13, the pattern for build-up break points is different from for draw-down break points.

For draw-down, the pattern is more similar to synthetic data. It is fairly likely that a narrow sharp peak indicate the beginning for a draw-down transient (see yellow lines in Figure 6.13), although exceptions still exist. Among 30 manual interpreted start points for shut-in periods, 23 points have minimum negative first order derivative comparing with their neighbour points.

For build-up, it becomes more complex. A cluster of points with relatively large positive first order derivative shows the start of build-up transient. The build-up break point usually locates at the first position of this cluster(see purple rectangles in Figure 6.13). It is not necessarily true that the first order derivative of the first point is the maximum one in this cluster. For start points of flowing periods (build-up transients), it is normal to have a smaller positive first order derivative values than their following neighbour points. Among 30 manual points, only 5 points have a larger first order derivatives comparing with their neighbour points.

Hints from statistics plots of manual interpreted points

The statistics for first order derivative values of the manual interpreted points could give us a quantitative view about the characteristics demonstrated above. Shown as the third row of figure 6.14, we could see first order derivatives of start points for flowing periods are more concentrated, most points are in the range of (-50,75), first order derivatives of start points for shut-in periods are roughly distributed over range (-900,-50).

Comparing two figures in the third row, we could conclude that: i) most of the absolute values of first order derivative for start points of flowing are much smaller than them of shut-in; ii) the distribution for the values of first order derivative for start points of flowings are much concentrated than them of shut-ins.

Combing these knowledge with the statistics of the whole data set (see Figure 6.15), we conclude that 40% of negative first order derivative values could be safely dropped while most of positive first order derivative values have to been retained during the coarse filtering step of the workflow. Thus the percentile range for the coarse filtering are set to be (10,50) in Chapter 6.2.2.

6.5.2 Pattern for tangent

From the histogram of left tangent and delta tangent, as shown in the first and second row of Figure 6.14, we observe similar distribution pattern as derivative. The distribution of the left tangent values and delta tangent values for start points of flowing periods are more concentrated than them of shut-in periods.

Start points of flowing periods.

- For delta tangent, approximately 90% points are in the range of (0,70).
- For left tangent, approximately 90% points are in the range of (-20,50).

Start points of shut-in periods.

- For delta tangent, as shown in first row, the values are distributed over a wide range (-700,200), approximately 75% points are in the range of (-250,0).
- For left tangent, half of the points are in the range (-50,0). Nevertheless, quite some points have relatively large absolute left tangent values. It proves the conclusion that we draw in Chapter 6.3.3, when choosing the criterion for selecting start points of shut-in periods, *Minimum of absolute value of Left Tangent* has worst performance among three criteria, since the manual interpreted points often have large *absolute value of Left Tangent.*

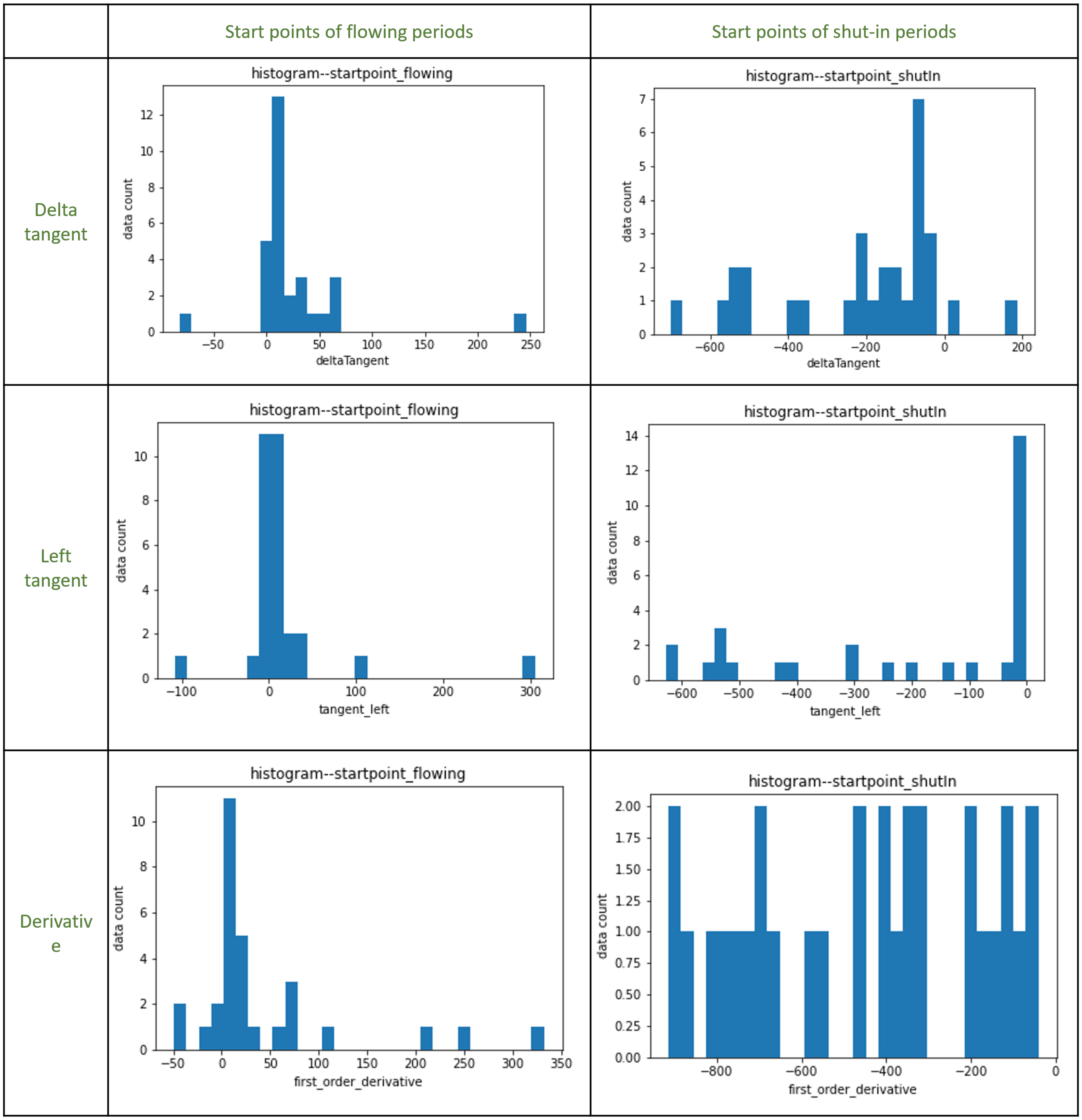
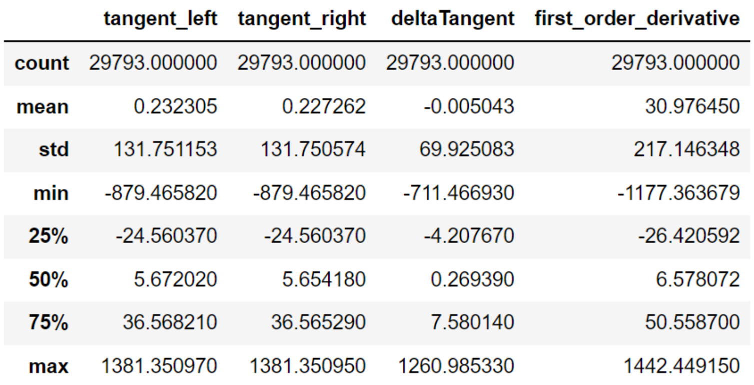


Figure 6.14: The histogram of delta tangent, left tangent, first order derivatives for start points of flowing periods & shut-in periods.



	tangent_left	tangent_right	deltaTangent	first_order_derivative
count	29793.000000	29793.000000	29793.000000	29793.000000
mean	0.232305	0.227262	-0.005043	30.976450
std	131.751153	131.750574	69.925083	217.146348
min	-879.465820	-879.465820	-711.466930	-1177.363679
25%	-24.560370	-24.560370	-4.207670	-26.420592
50%	5.672020	5.654180	0.269390	6.578072
75%	36.568210	36.565290	7.580140	50.558700
max	1381.350970	1381.350950	1260.985330	1442.449150

Figure 6.15: The statistics of first order derivatives, left tangent, right tangent, delta tangent based on whole data set.

6.5.3 Summarization

The discussion about patterns of derivative and tangent could lead to the following conclusions.

1. The distribution of these three metrics(delta tangent, left tangent, derivative) for start points of flowing periods are different from them of shut-in periods.

   According to this conclusion, two different separating thresholds should be applied to flowing and shut-in periods for a better detection. However to reduce the number of input parameters, we only use single threshold for both two periods, as we did in **DeltaTangent** method. This leads to a large number of detected break points candidate. A selection step will be required further to deal with these candidates, thus initiating us to design the forth and fifth step of the work flow.

2. A combination of different metrics is a must to identify start points for two periods.

   Take delta tangent for example, as shown in 6.14, the values of delta tangent for start points of shut-in periods are distributed in a relatively wide range which is approximately (-700,200). And if we combine the knowledge of delta tangent statistics of the whole data set, we could see there are a great number of points have delta tangent in this range. Thus, another metric will be needed to pick up the true

> points from the candidates produced by **DeltaTangent** method. As we did in the forth step of workflow, we use the maximum derivative to choose the start points for shut-in.

These insight to the metrics(derivative & tangent) patterns really helps when we design our algorithm, workflow, as well as choosing the parameters and thresholds.

Chapter 7

Model deployment

7.1 Web App scheme

For our web app, our initial scheme is to use Python Flask (back-end), Javascript (front-end) and Rest API. However we finally gave up this scheme and turn to a python library *Streamlit*[1].

Streamlit is an open-source app framework that could easily turn data scripts into web apps. Streamlit makes life much easier and is much more light weighted than a typical web app solution. All scripts are in Python. It is extremely suitable for web app in machine learning and data science.

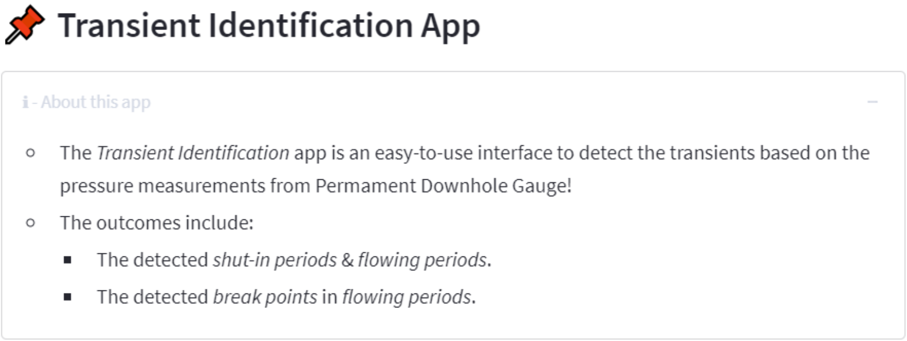
The app was deployed in *Streamlit Cloud* and can be visited **here**[2].

The scripts were pushed to a **GitHub repository**[3].

[1] https://streamlit.io/

[2] https://share.streamlit.io/juneciel510/transient_identification/main/app.py

[3] https://github.com/juneciel510/transient_identification



Transient Identification App

i - About this app

- The *Transient Identification* app is an easy-to-use interface to detect the transients based on the pressure measurements from Permament Downhole Gauge!
- The outcomes include:
  - The detected *shut-in periods* & *flowing periods*.
  - The detected *break points* in *flowing periods*.

Figure 7.1: User interface: introduction.

7.2 User interface

The user interface use one-page frame consisting of four parts: i)intro, ii) data upload & preview, iii) select method and parameters, iv) results.

7.2.1 Intro

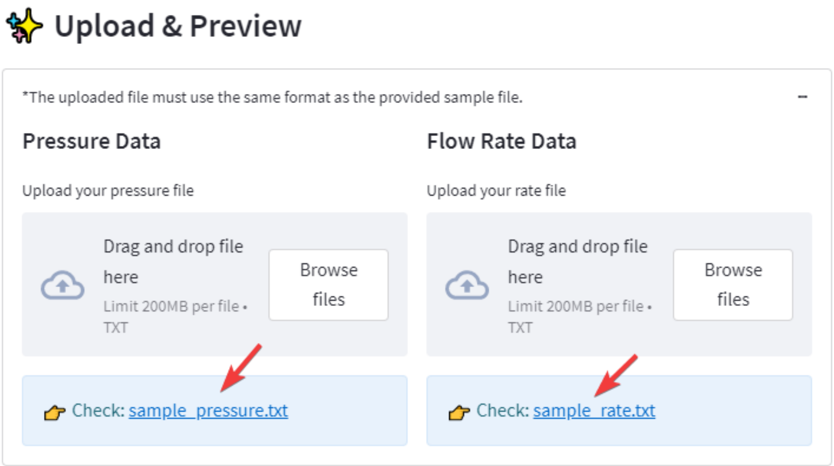
This part briefly introduces the web app and the outcomes that will be produced, see Figure 7.1.

7.2.2 Upload & preview

Sample pressure file and rate file are provided to show the format for the files to be uploaded, see red arrows in Figure 7.2. The head of the uploaded files can be previewed in the preview field, see red rectangle in Figure 7.3.

7.2.3 Select method & parameters

The users can experiment with different methods and adjust the parameters according to their data sets. The parameter field (see rectangle in



Upload & Preview

*The uploaded file must use the same format as the provided sample file.

Pressure Data

Upload your pressure file

Drag and drop file here

Limit 200MB per file • TXT

Check: sample_pressure.txt

Flow Rate Data

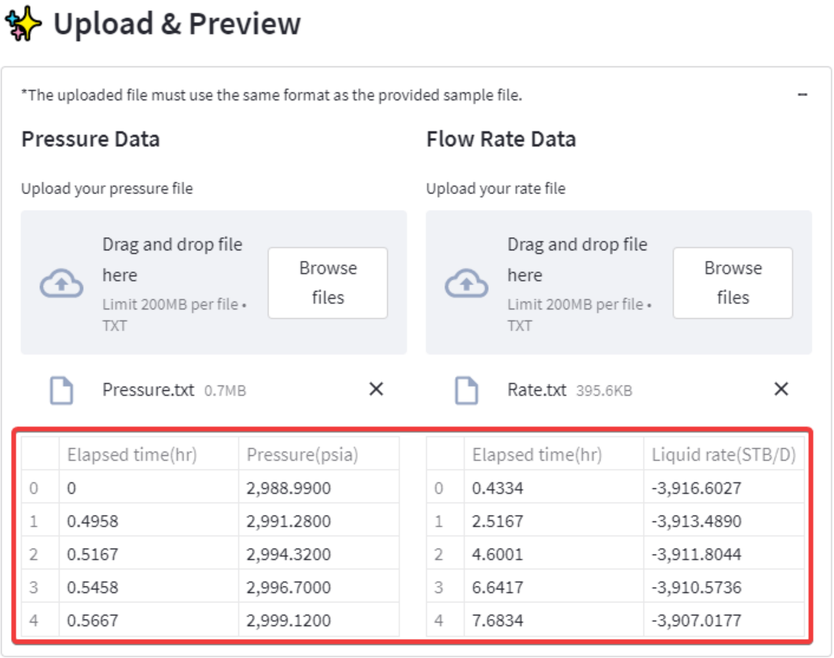
Upload your rate file

Drag and drop file here

Limit 200MB per file • TXT

Check: sample_rate.txt

Figure 7.2: User interface: Upload & Preview. Red arrows: sample files show the format for uploaded files.



Upload & Preview

*The uploaded file must use the same format as the provided sample file.

Pressure Data

Upload your pressure file

Drag and drop file here

Limit 200MB per file • TXT

Pressure.txt 0.7MB

	Elapsed time(hr)	Pressure(psia)
0	0	2,988.9900
1	0.4958	2,991.2800
2	0.5167	2,994.3200
3	0.5458	2,996.7000
4	0.5667	2,999.1200

Flow Rate Data

Upload your rate file

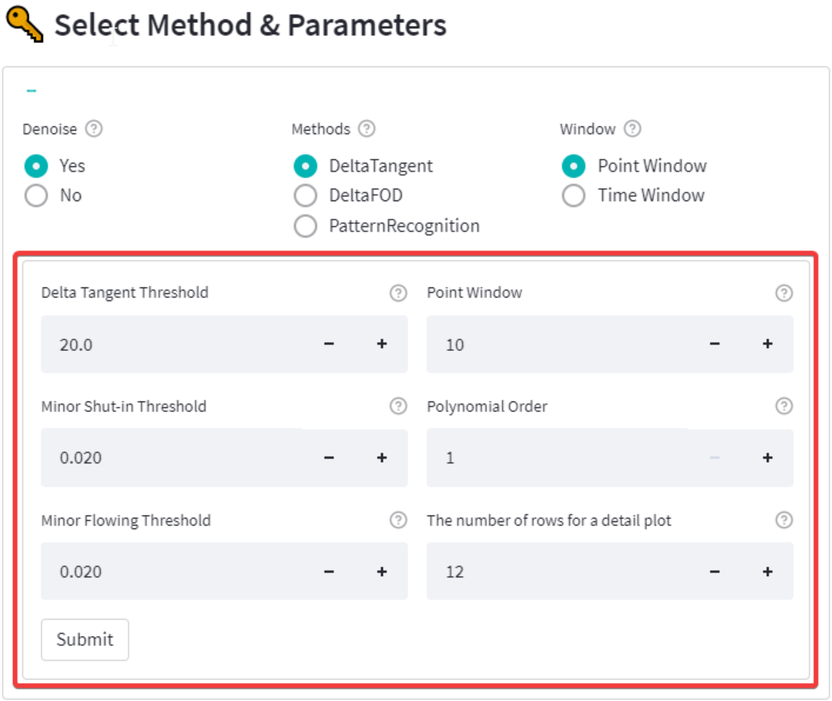
Drag and drop file here

Limit 200MB per file • TXT

Rate.txt 395.6KB

	Elapsed time(hr)	Liquid rate(STB/D)
0	0.4334	-3,916.6027
1	2.5167	-3,913.4890
2	4.6001	-3,911.8044
3	6.6417	-3,910.5736
4	7.6834	-3,907.0177

Figure 7.3: User interface: Upload & Preview. Red rectangle: preview field shows the head of the uploaded file.



Select Method & Parameters

Denoise

● Yes

○ No

Methods

● DeltaTangent

○ DeltaFOD

○ PatternRecognition

Window

● Point Window

○ Time Window

Delta Tangent Threshold: 20.0

Point Window: 10

Minor Shut-in Threshold: 0.020

Polynomial Order: 1

Minor Flowing Threshold: 0.020

The number of rows for a detail plot: 12

Figure 7.4: User interface: Select method & Parameters. Red rectangle: parameter field will vary corresponding to different methods.

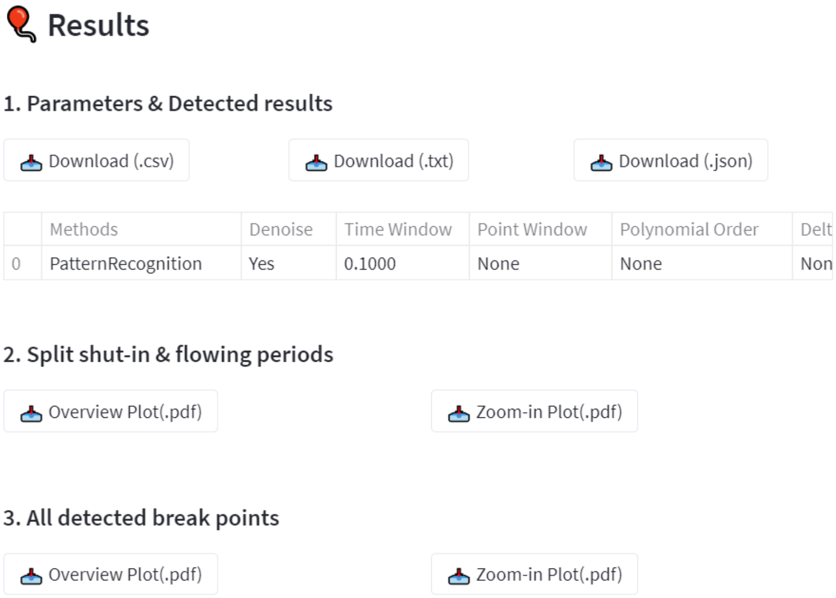
Figure 7.4) will vary according to different methods. The learned patterns for **PatternRecognition** method are embedded in web app and are used for predictions. The window can switch between point window and time window. The corresponding fields will change according to users' choices.

7.2.4 Show results

This web app generates two types of results and could be downloaded by clicking the downloading button, see Figure 7.5.

1. The parameters chosen by user and detected points indices for *Task 1* and *Task 2* could be downloaded as csv, txt and json files. A json example is Appendix E.

   Line 2 $\sim$ 8 corresponds to the chosen method and parameters.



Results

1. Parameters & Detected results

	Methods	Denoise	Time Window	Point Window	Polynomial Order	Delt
0	PatternRecognition	Yes	0.1000	None	None	Non

2. Split shut-in & flowing periods

3. All detected break points

Figure 7.5: User interface: results.

Line 9 $\sim$ 10 shows the number of detected shut-in Periods and flowing periods respectively.

Line 11 $\sim$ 12 shows the number of all detected build-up break points and draw-down break points.

Line 13 $\sim$ 45 specifies the start point and end point for every shut-in period.

Line 46 $\sim$ 197 specifies the start point and end point for every flowing period and all the build-up points included in corresponding period.

2. The visualizations for *Task 1* and *Task 2* are saved as PDF files for downloading. The whole data set is plotted in one row in *Overview Plot*. The whole data set are divided into multiple rows to show details of the testing data set in *Zoom-in Plot*. The number of rows could be specified by user with the inputted parameter: *The number of rows for a detail plot*. Examples for *Zoom-in Plot* of *Task 1* and *Task 2* are Appendix C and Appendix D respectively.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

The novelty of this study lies in its newly developed methodologies for transient identification and a 5-steps workflow. And a web app is also deployed to integrate all these methods and workflow.

Three methods In this study, we developed three methods based on derivative, tangent and pattern recognition, named as **DeltaFOD**, **DeltaTangent** and **PatternRecognition**. Two new algorithms were designed: *DeltaFOD* (Algorithm 2) and *DeltaTangent*(Algorithm 3). For **PatternRecognition**, a new learning process to learn patterns was suggested. The conclusions drawn from these 3 methods are listed below:

1. The study brings an new idea of the transient identification methodology, which is taking average of a certain metric (tangent, derivative, etc) of a window. This methodology is effective for a noisy data set.
2. The average algorithm sometimes will cause small position deviation of the detected position when comparing with the ground truth. However the results are still satisfactory.

3. For a severely uneven-distributed data set, it would recommend to use a time window rather than a point window. Both of these two ways are implemented in web app.

A five-step workflow In this study, we consider the transient identification problem as two-stage pipeline of sequence identification tasks: i) to split the data set into shut-in and flowing periods; ii) to detect multi-rate break points in flowing periods.

To address this problem, a new workflow was designed. It consists of 5 steps: i) Denoising; ii) Coarse Filtering; iii) Methods pipeline & Cluster representing; iv) *Task 1* processing; v) *Task 2* processing. The three methods(**DeltaFOD**, **DeltaTangent** and **PatternRecognition**) will be embedded in *methods pipeline* sub-module individually or in a combination. The conclusions drawn from this 5-steps workflow are listed below:

1. This 5-steps workflow achieves a satisfactory results for the two-stages transients identification task.
2. All the methods have a good performance in *Task 1*. Nevertheless, **DeltaFOD** obtains best result for *Task 1*. **DeltaTangent** performs the best for *Task 2*.
3. This workflow has a good tolerance to the noise and parameters setting.
4. More multi-rate break points could be identified by decreasing the multi-rate threshold.

User-friendly web app We use the Streamlit library to build a web app to integrate the three methods and the workflow that we designed. The web app is easy to use, and provides the downloading for the detection results.

8.2 Future work

1. Experiment with different denoising techniques discussed in this study(e.g. Moving average, Hampel filter, etc), to see if a better performance will

be achieved.

2. Experiment with more possible combinations of our designed methods.
3. To improve the learned patterns for **PatternRecognition** method.
4. To optimize the scripts to decrease the model running time.
5. To develop an un-supervised machine learning approach for the transients identfication tasks.
6. To test the designed methods and workflow when more data sets are available.

Bibliography

[1] Ramey Jr., H.J. (1975). Pressure transient analysis for geothermal wells. In: Proceedings of the Second UN Symposium on the Development and Use of Geothermal Resources, pp. 1749-1757.

[2] Allain, O. F. (1988). Use of artificial intelligence for model identification and parameter estimation in well test interpretation. Engineer's thesis, Stanford University, Stanford, Califonia.

[3] Ershaghi, I., Li, X., Hassibi, M. and Shikari, Y. (1993).A robust neural network model for pattern recognition of pressure transient test data. Presented at SPE Annual Technical Conference and Exhibition, 3-6 October, Houston, Texas, SPE-26427-MS. doi:10.2118/26427-MS.

[4] Sinha, S., & Panda, M. N. (1996). Well-test model identification with self-organizing feature map. SPE Computer Applications, 8(04), pp. 106-110.

[5] AlMaraghi, A. M., & El-Banbi, A. H. (2015, September). Automatic reservoir model identification using artificial neural networks in pressure transient analysis. In SPE North Africa Technical Conference and Exhibition. OnePetro.

[6] Shchipanov, A. A., Berenblyum, R. A., and L.. Kollbotn. (2014). Pressure Transient Analysis as an Element of Permanent Reservoir Monitoring. Paper presented at the SPE Annual Technical Conference and Exhibition, Amsterdam, The Netherlands. doi: https://doi.org/10.2118/170740-MS.

[7] Athichanagorn, S., Horne, R., and Kikani, J. (2002). Processing and interpretation of long-term data acquired from permanent pressure gauges. SPE Reservoir Evaluation & Engineering, 3(3):384–391. SPE-80287-PA.

[8] Athichanagorn, S. (1999). Development of an interpretation methodology for long-term pressure data from permanent downhole gauges.

[9] Ribeiro, Priscila Pires, Adolfo Ferroni, Jose Puntel, Eduardo. (2006). Use of Wavelet Transform in PDG Data Treatment. Proceedings - SPE Annual Technical Conference and Exhibition. 1. 10.2118/100719-MS.

[10] Olsen, S., Nordtvedt, J.E. (2005). Automatic Filtering and Monitoring of Real-Time Reservoir and Production Data.

[11] Donoho, D. L., & Johnstone, I. M. (1994). Ideal Spatial Adaptation by Wavelet Shrinkage. Biometrika, 81(3), 425–455. https://doi.org/10.2307/2337118

[12] Kikani, Jitendra, and Meiqing He. (1998). Multi-resolution Analysis of Long-Term Pressure Transient Data Using Wavelet Methods. Paper presented at the SPE Annual Technical Conference and Exhibition. doi: https://doi.org/10.2118/48966-MS

[13] McCracken, Michael, and David M. Chorneyko.(2006) Rate Allocation Using Permanent Downhole Pressures.Paper presented at the SPE Annual Technical Conference and Exhibition. doi: https://doi.org/10.2118/103222-MS

[14] Chuan Tian and Roland N. Horne.(2015) Machine Learning Applied to Multiwell Test Analysis and Flow Rate Reconstruction. . ISBN 978-1-61399-376-7.

[15] Zarrouk, Sadiq & Mclean, Katie. (2019). Geothermal Well Test Analysis: Fundamentals, Applications and Advanced Techniques, pp. 63-64.

[16] H. Rai. (2005). Analyzing rate data from permanent downhole gauges. MS report, Stanford University.

[17] Khong, C.K. (2001). Permanent Downhole Gauge Data Interpretation, MS report, Stanford University.

[18] Nomura, M. (2006). Processing and interpretation of pressure transient data from permanent downhole gauges.

[19] Suzuki, Satomi & Chorneyko, David. (2009). Automatic Detection of Pressure-Buildup Intervals From Permanent Downhole Pressure Data Using Filter Convolution. 10.2118/125240-MS.

[20] Hampel, F. (1971). A General Qualitative Definition of Robustness. Annals of Mathematical Statistics, 42, 1887-1896.

[21] Hampel, F. R. (1974). The Influence Curve and Its Role in Robust Estimation. Journal of the American Statistical Association, 69(346), 383–393. https://doi.org/10.2307/2285666

[22] Suzuki, Satomi. (2018). Using Similarity-Based Pattern Detection to Automate Pressure Transient Analysis. 10.2118/193285-MS.

[23] A. Shchipanov, L. Kollbotn, R. Berenblyum. (2017). Integrating Pressure Transient Analysis into History Matching. 79th EAGE Conference Exhibition. Paris, France, 12-15 June. doi.org/10.3997/2214-4609.201700995

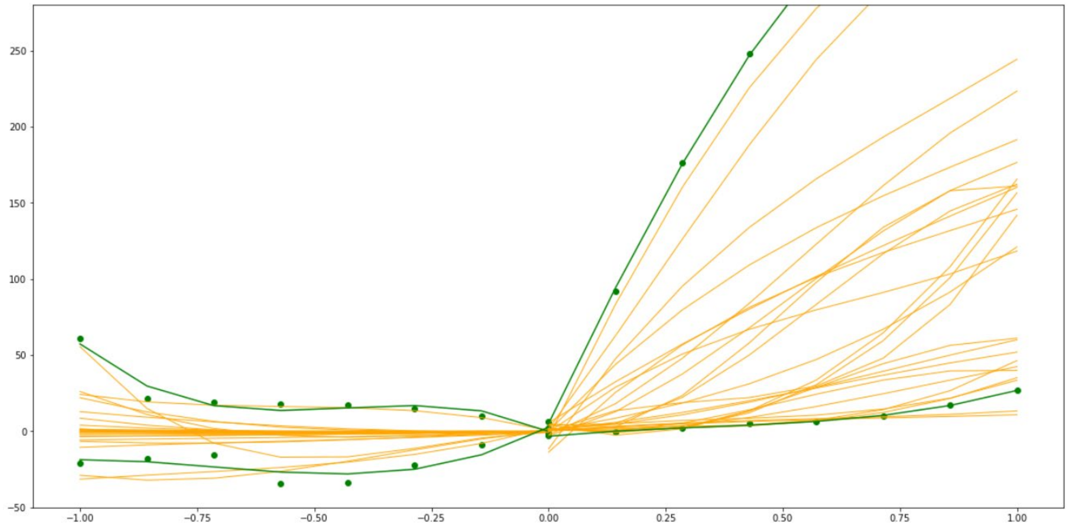
[24] Liu Junrong, Yao Jun, Yu Weiqiang. (2016). Study of Identification Method of Transient Flow from Permanent Downhole Pressure Data. Journal of Southwest Petroleum University(Science Technology Edition)

[25] Thomas,O. (2002). The data as the Model: Interpreting Permanent Downhole Gauge Data without Knowing the Reservoir Model. Master Thesis, Stanford University.

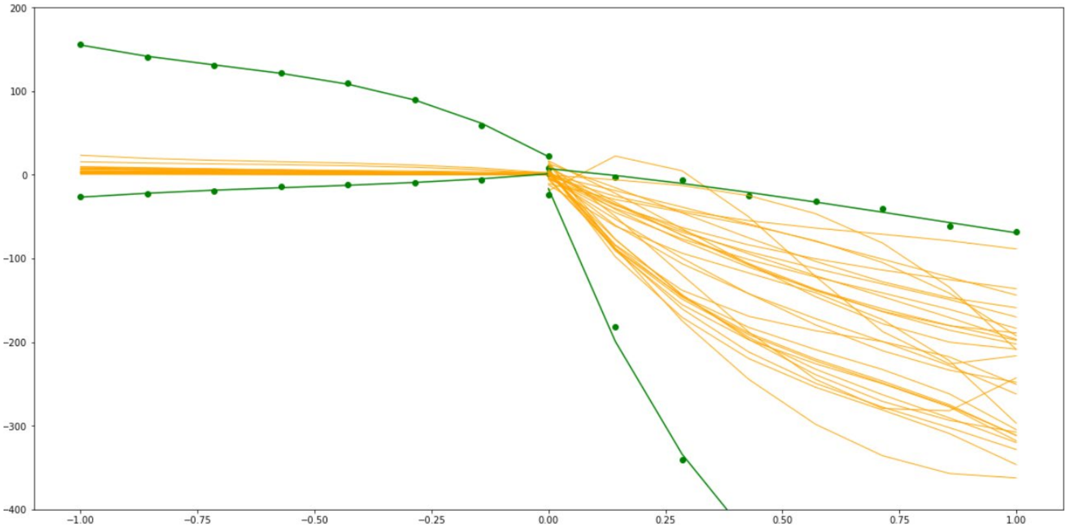
Appendix A

The learned patterns

build-up pattern



draw-down pattern



Note:

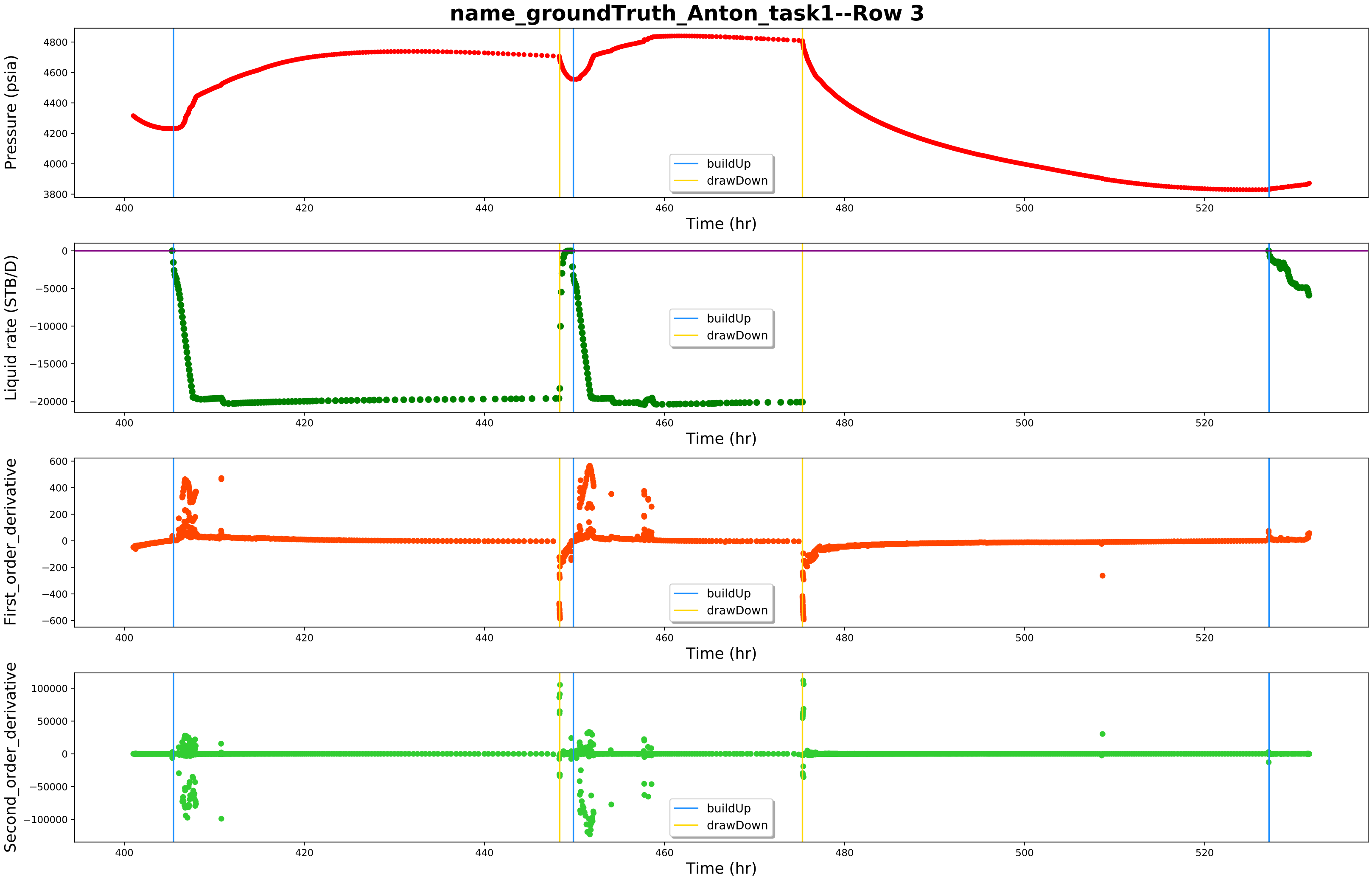
The green curves indicate the learned patters.

The orange curves indicate the learning process.

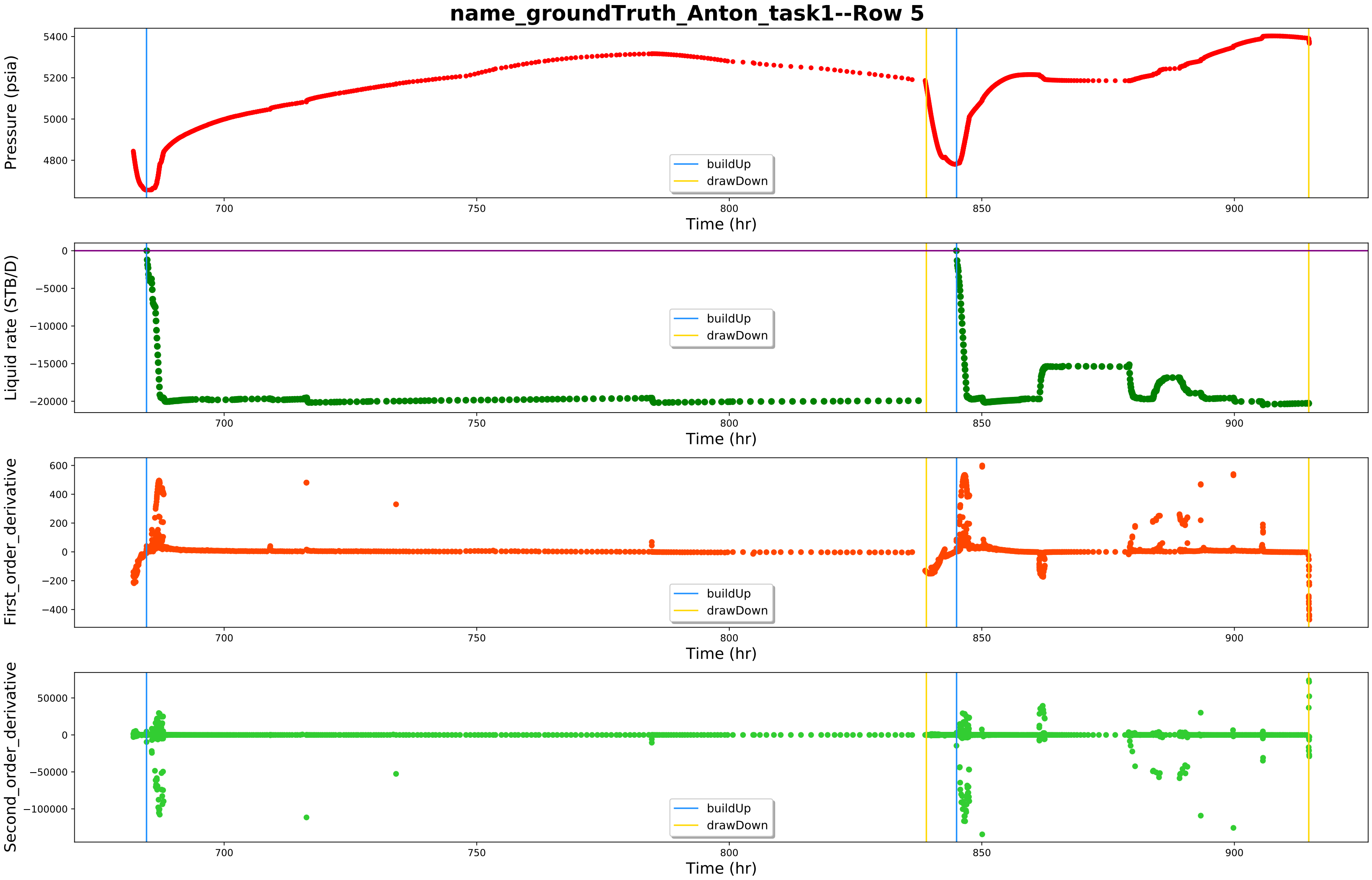
Appendix B

The manual interpreted result of ***Task 1*** (sliced version)

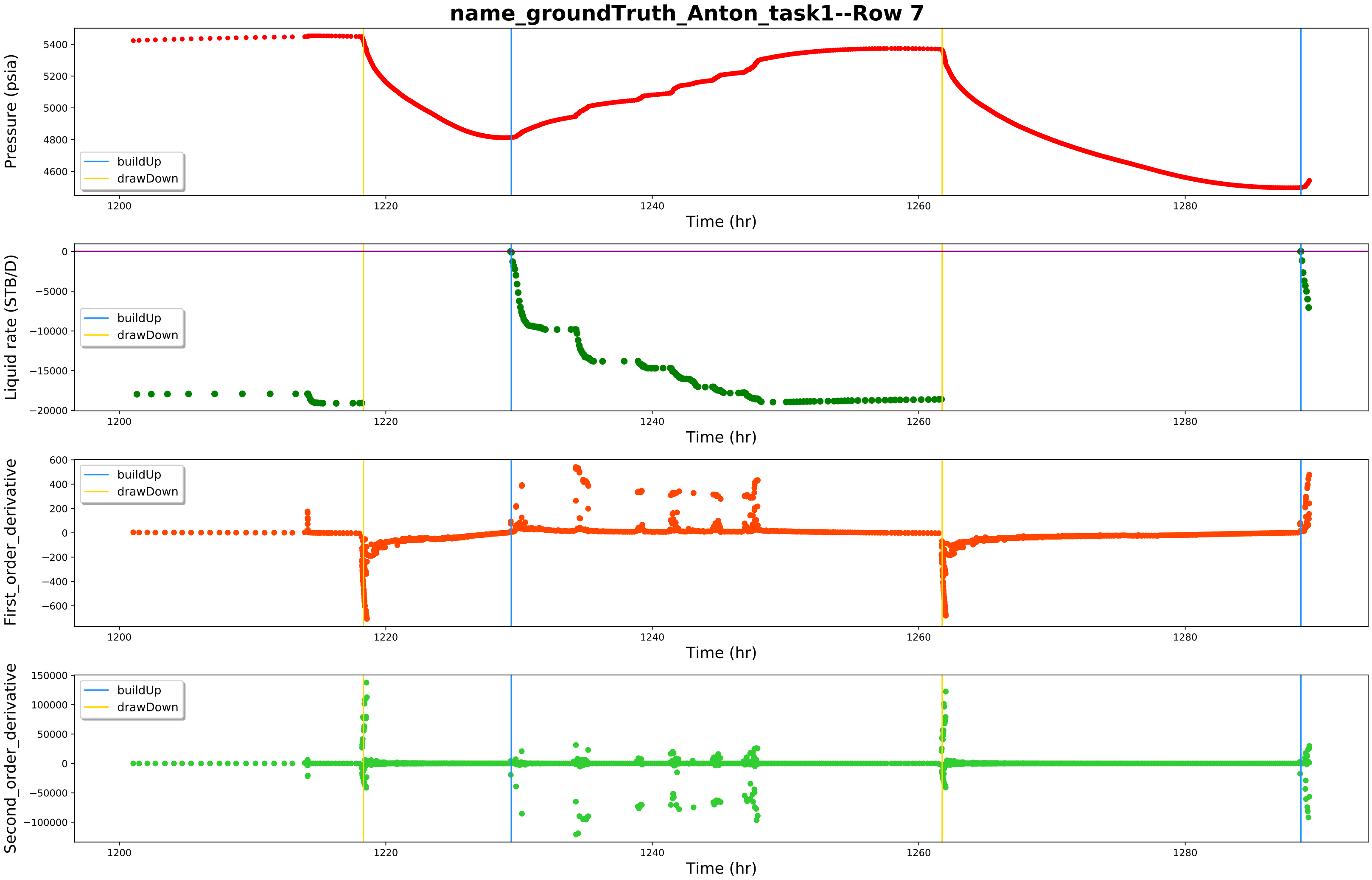
The manual interpreted result of *Task 1* (sliced version)



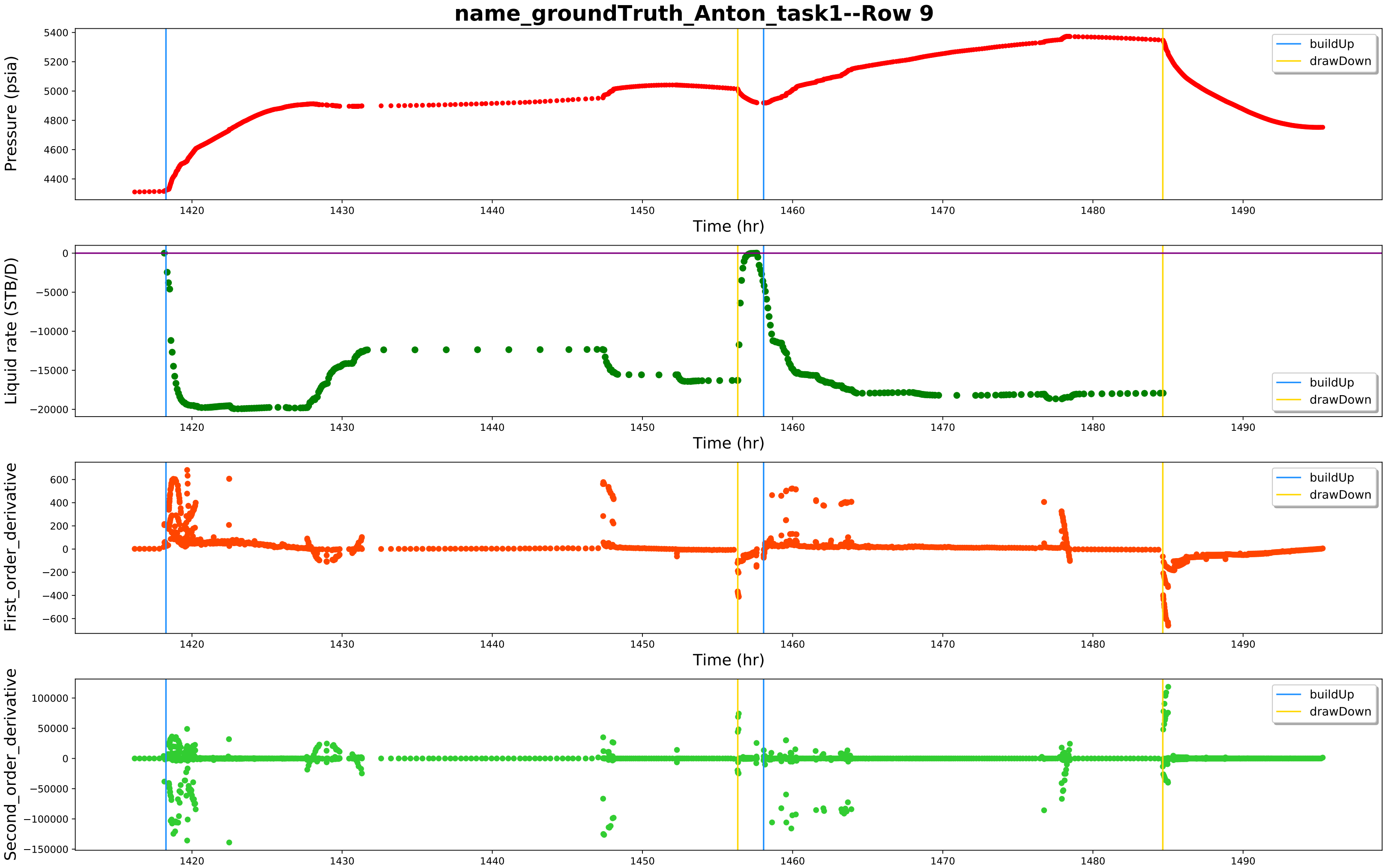
The manual interpreted result of *Task 1* (sliced version)



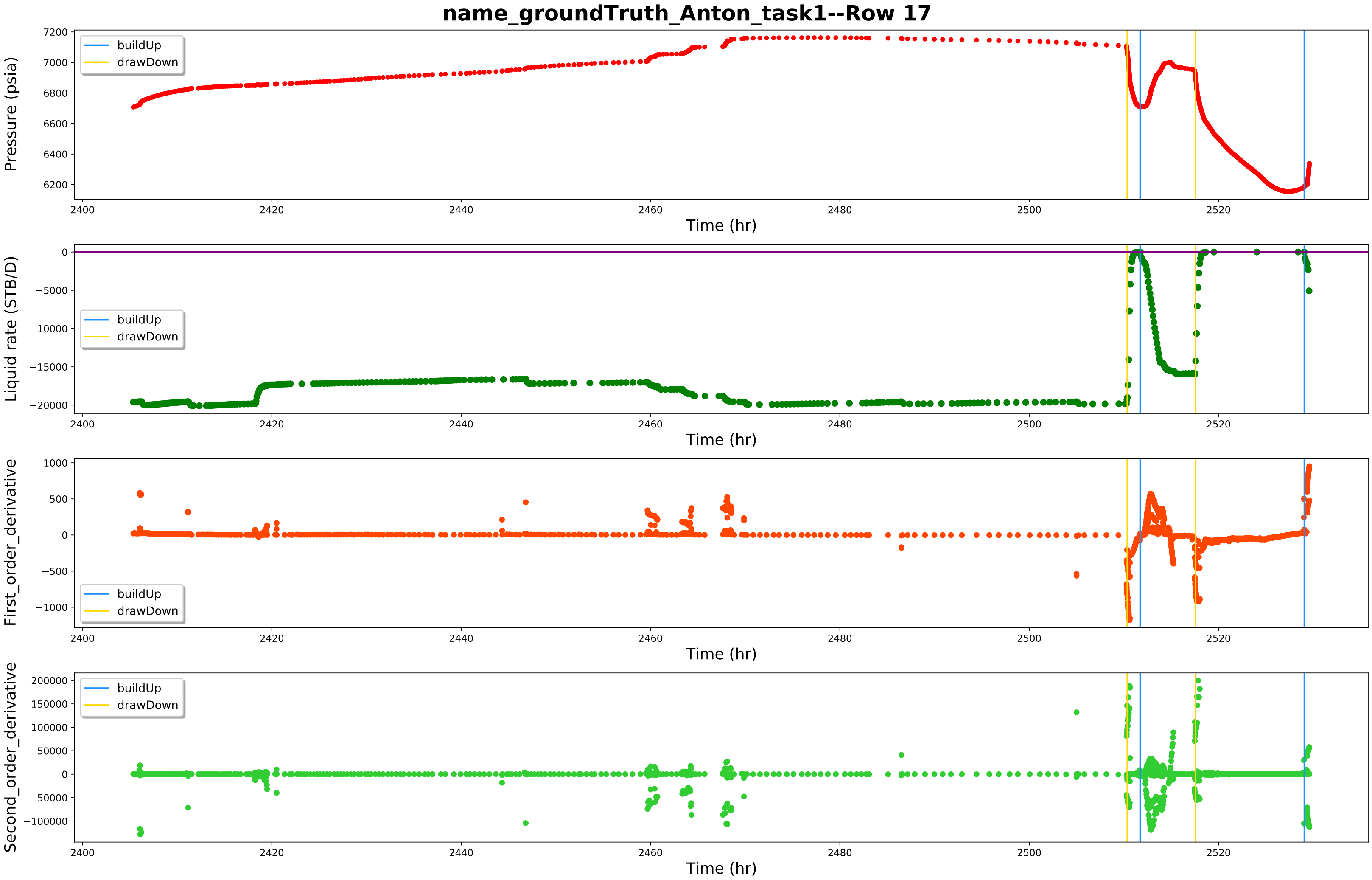
The manual interpreted result of *Task 1* (sliced version)



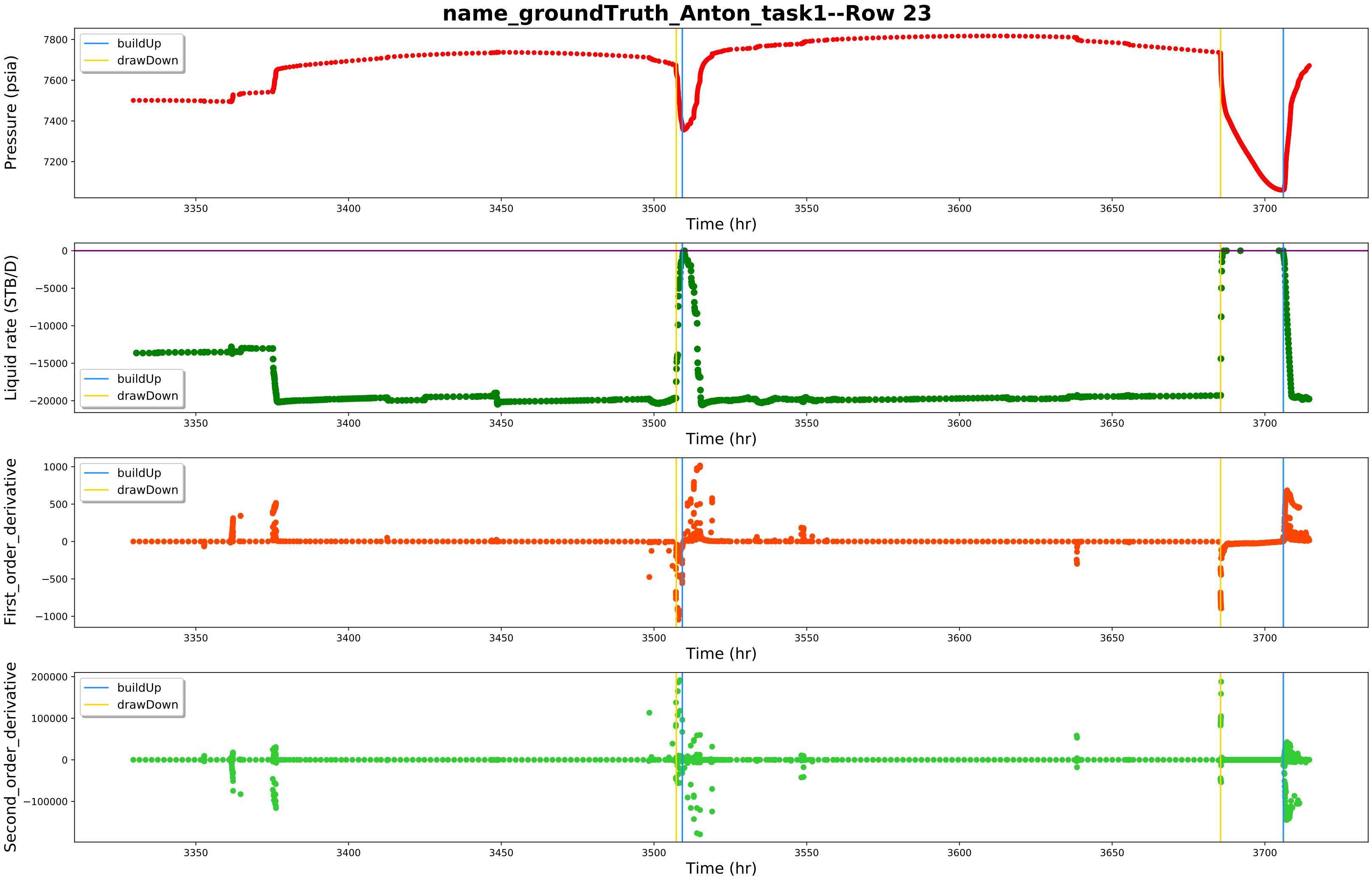
The manual interpreted result of *Task 1* (sliced version)



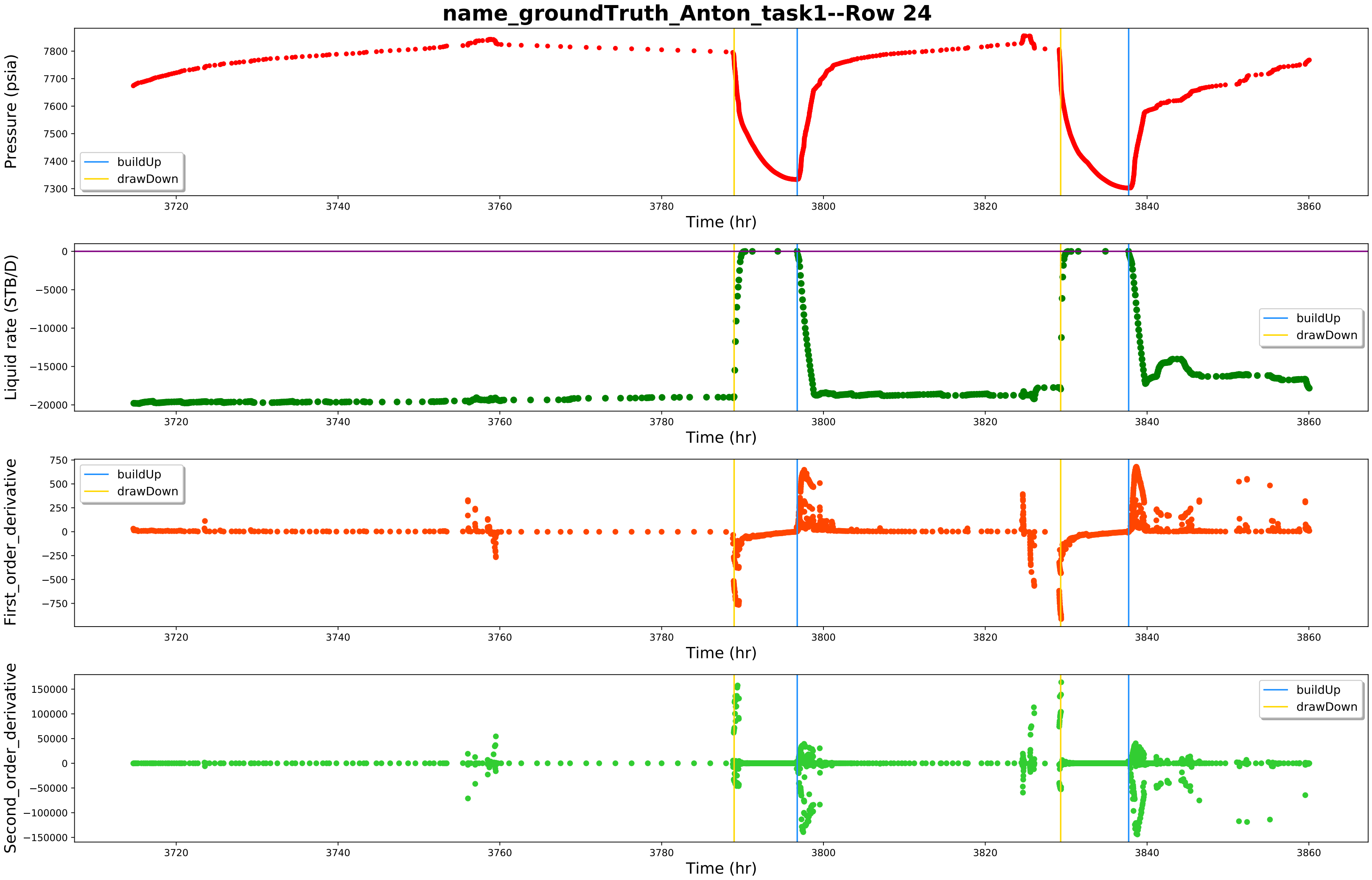
The manual interpreted result of *Task 1* (sliced version)



The manual interpreted result of *Task 1* (sliced version)



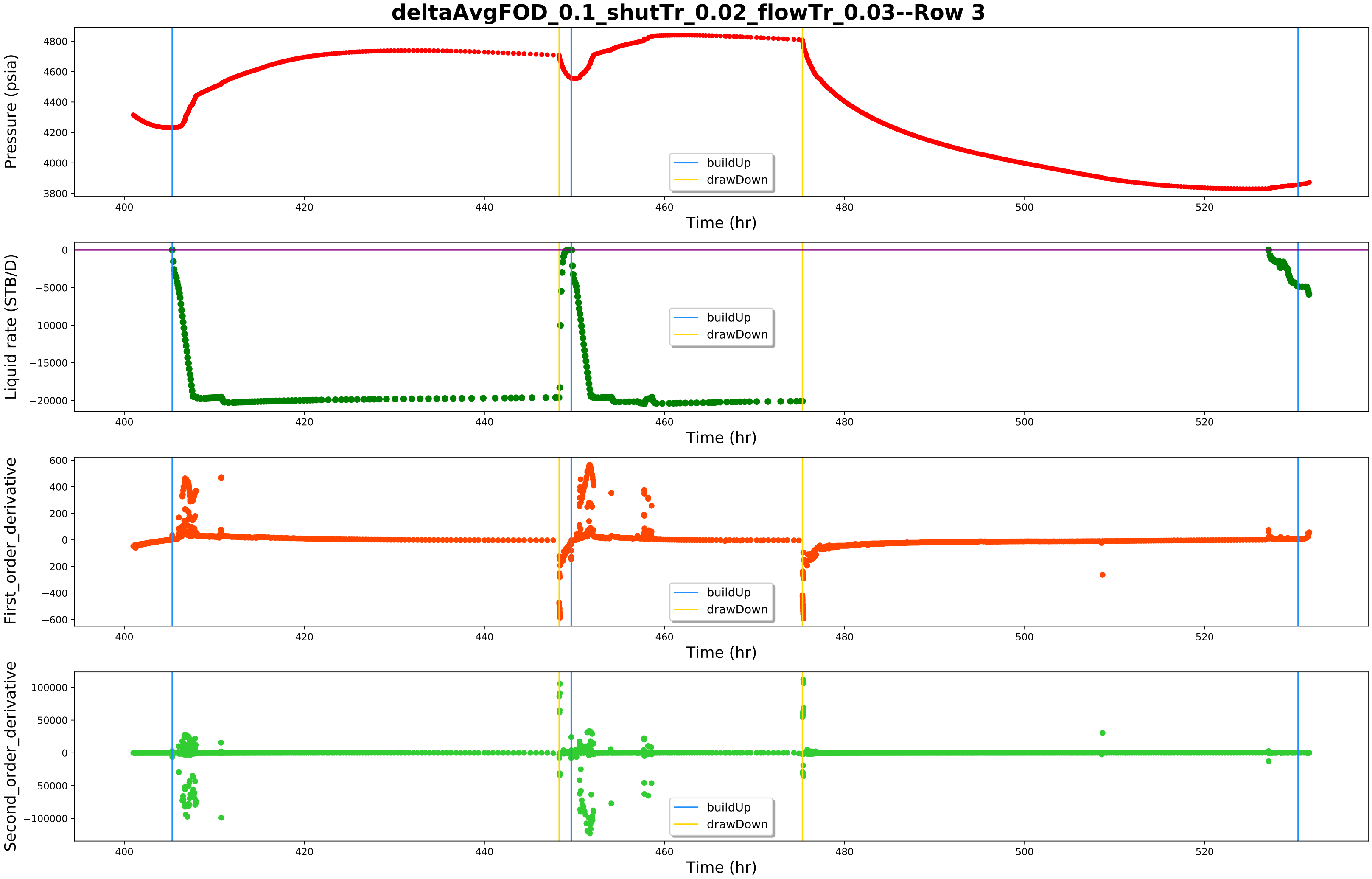
The manual interpreted result of *Task 1* (sliced version)

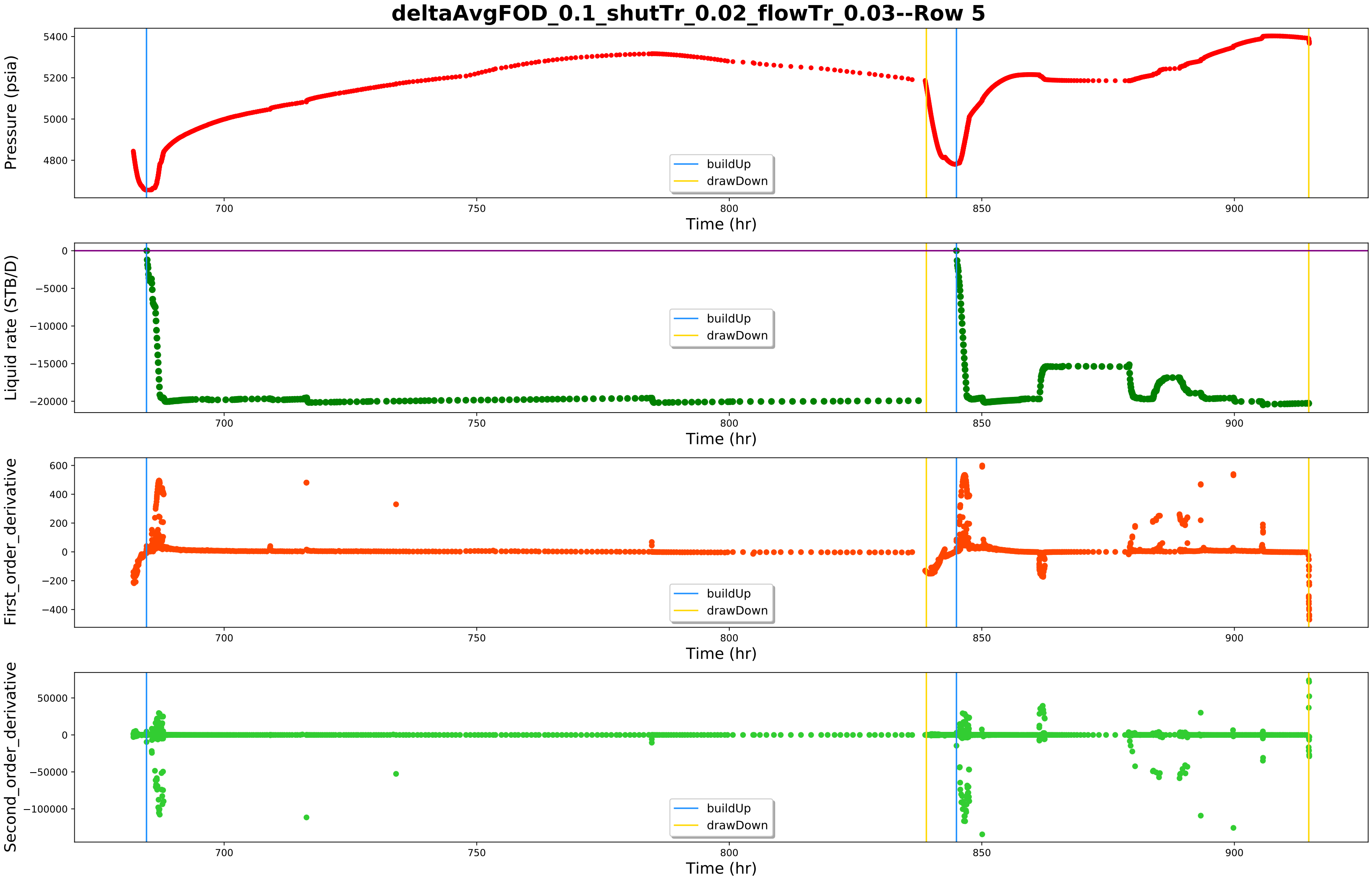


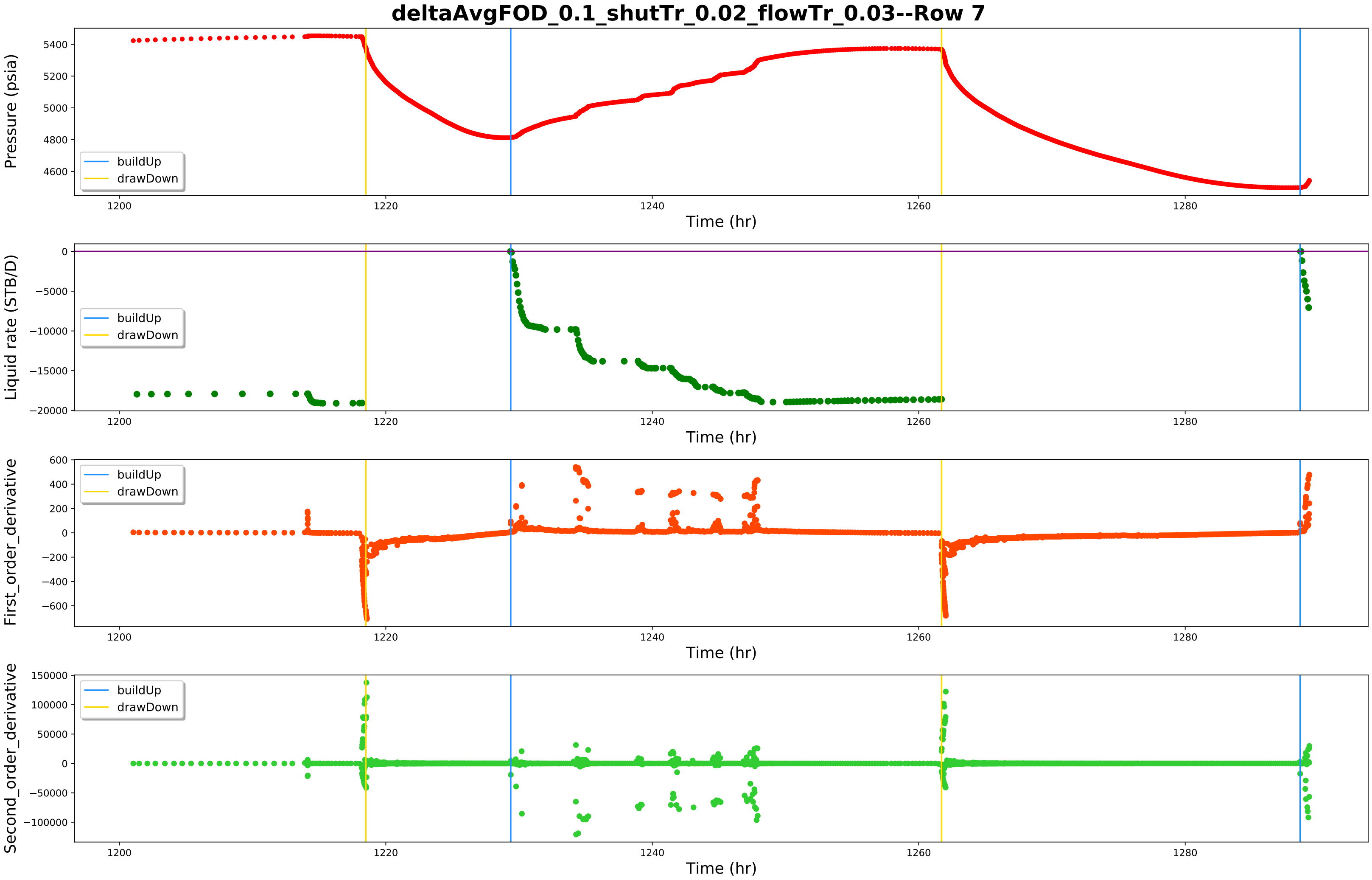
Appendix C

The best result of *Task 1* (sliced version)

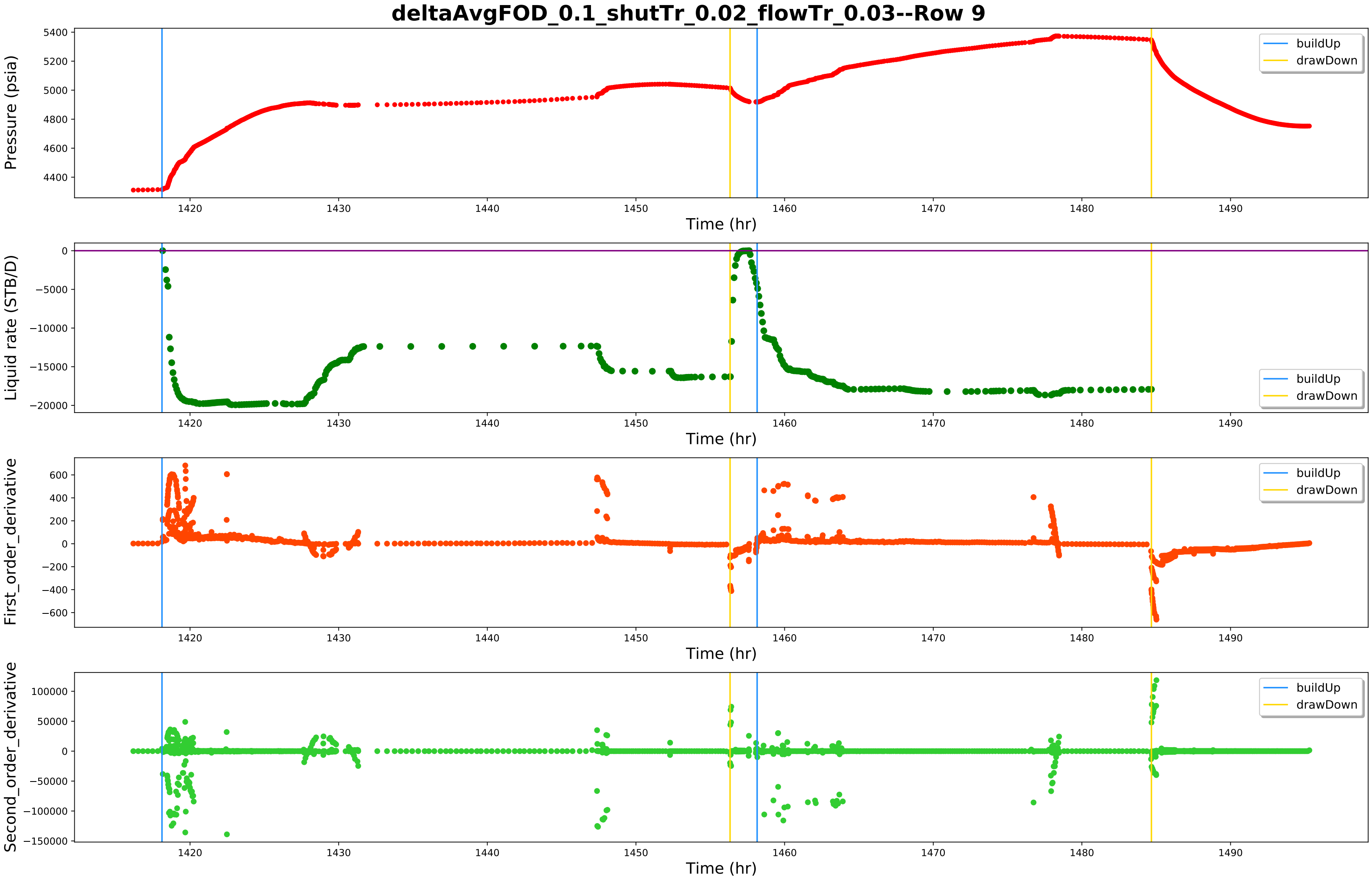
The best result of *Task 1* (sliced version)



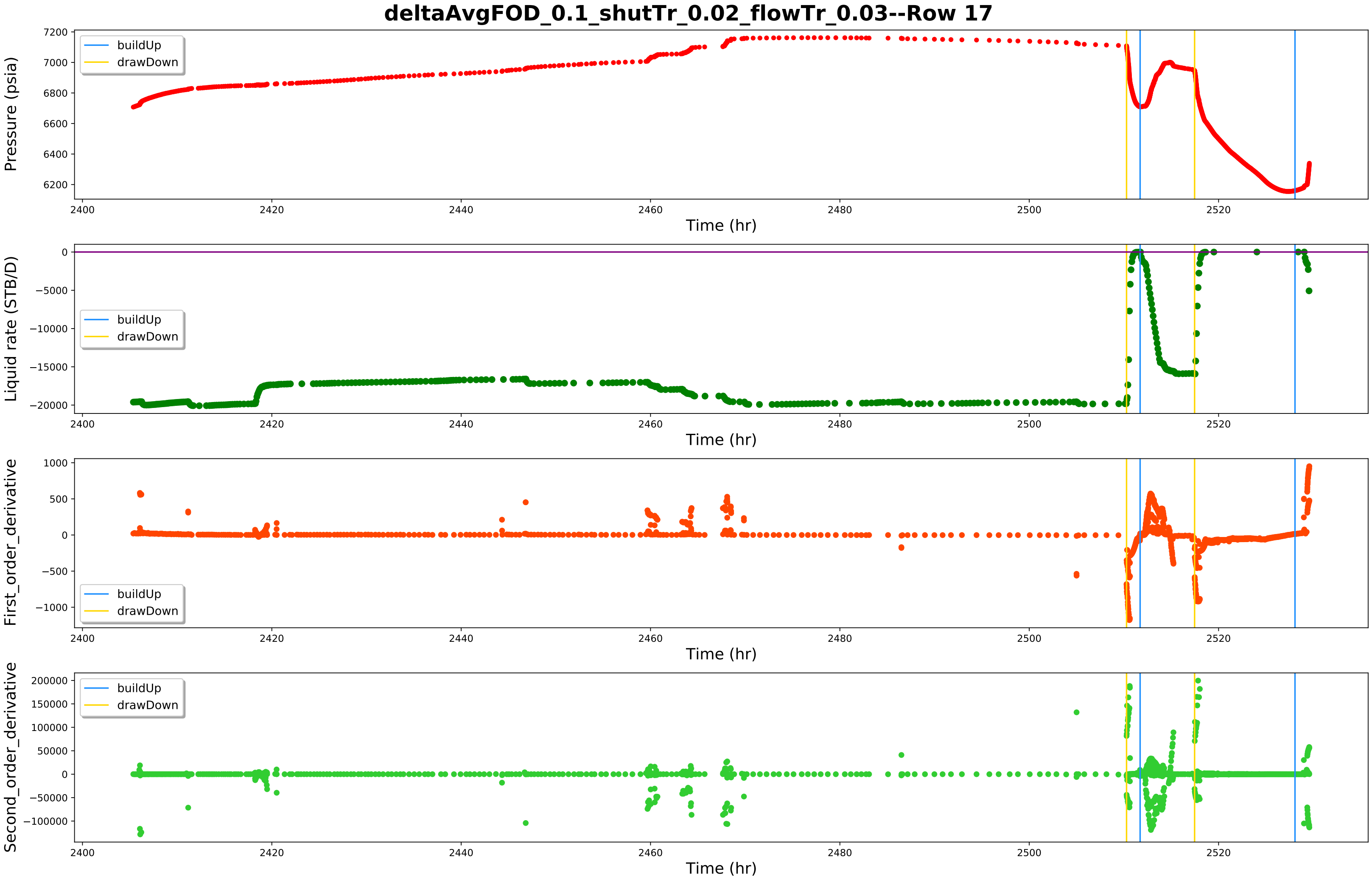

deltaAvgFOD_0.1_shutTr_0.02_flowTr_0.03--Row 5
Pressure (psia)
5400
5200
5000
4800
Liquid rate (STB/D)
0
−5000
−10000
−15000
−20000
First_order_derivative
600
400
200
0
−200
−400
Second_order_derivative
50000
0
−50000
−100000
700
750
800
850
900
Time (hr)
buildUp
drawDown


deltaAvgFOD_0.1_shutTr_0.02_flowTr_0.03--Row 7
Pressure (psia)
5400
5200
5000
4800
4600
buildUp
drawDown
Time (hr)
1200
1220
1240
1260
1280
Liquid rate (STB/D)
0
−5000
−10000
−15000
−20000
First_order_derivative
600
400
200
0
−200
−400
−600
Second_order_derivative
150000
100000
50000
0
−50000
−100000

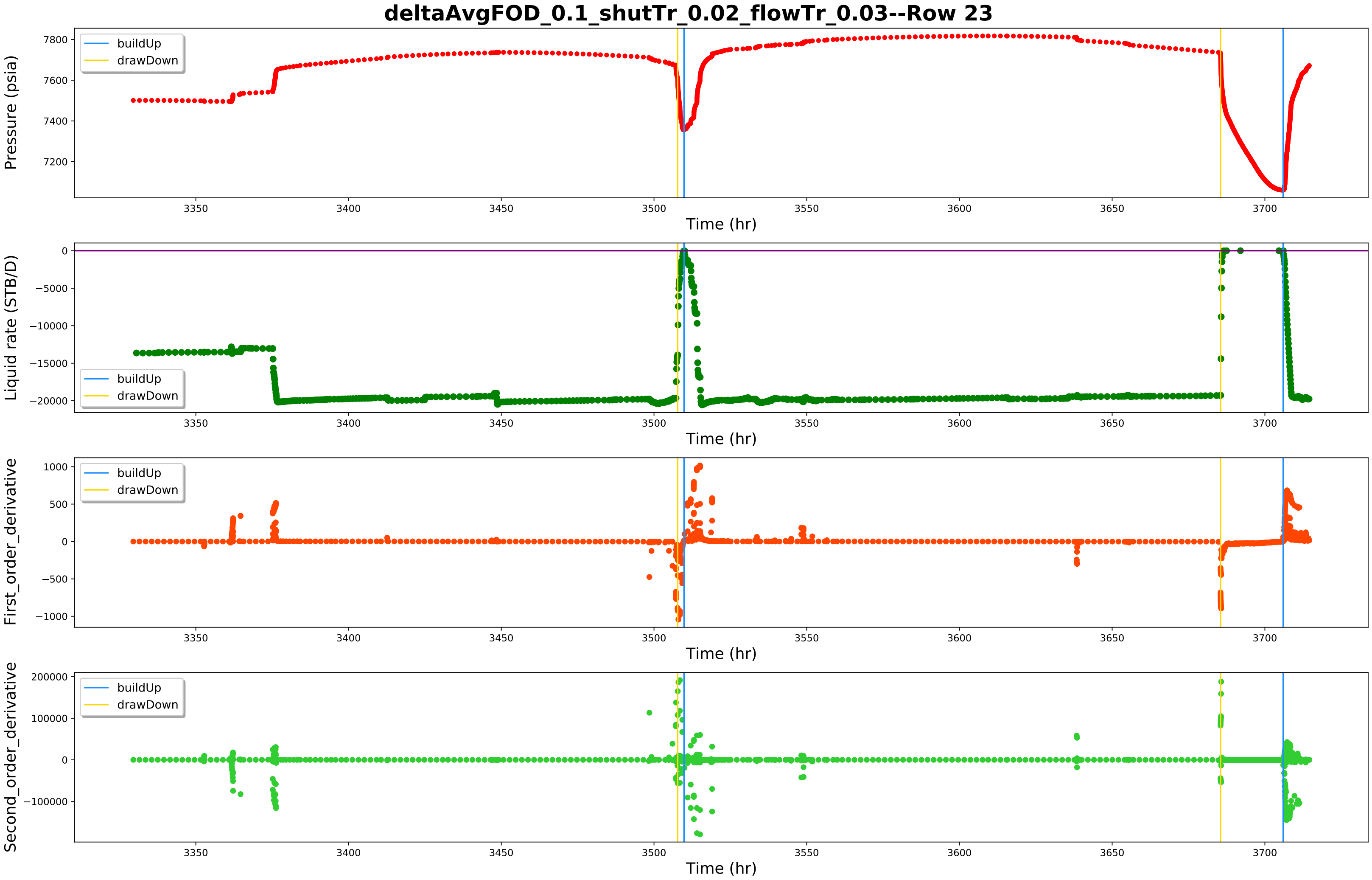
The best result of *Task 1* (sliced version)



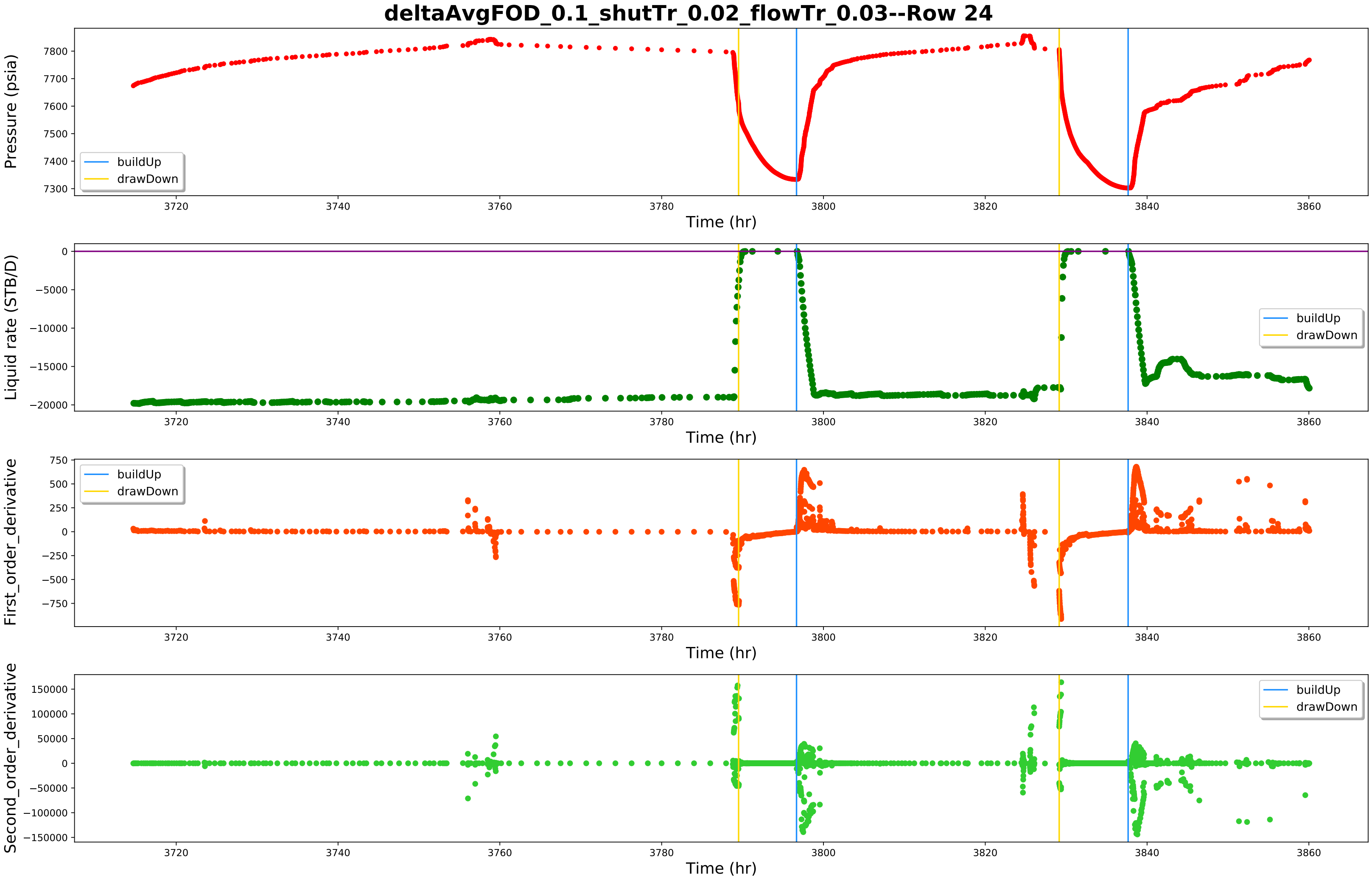
The best result of *Task 1* (sliced version)



The best result of *Task 1* (sliced version)



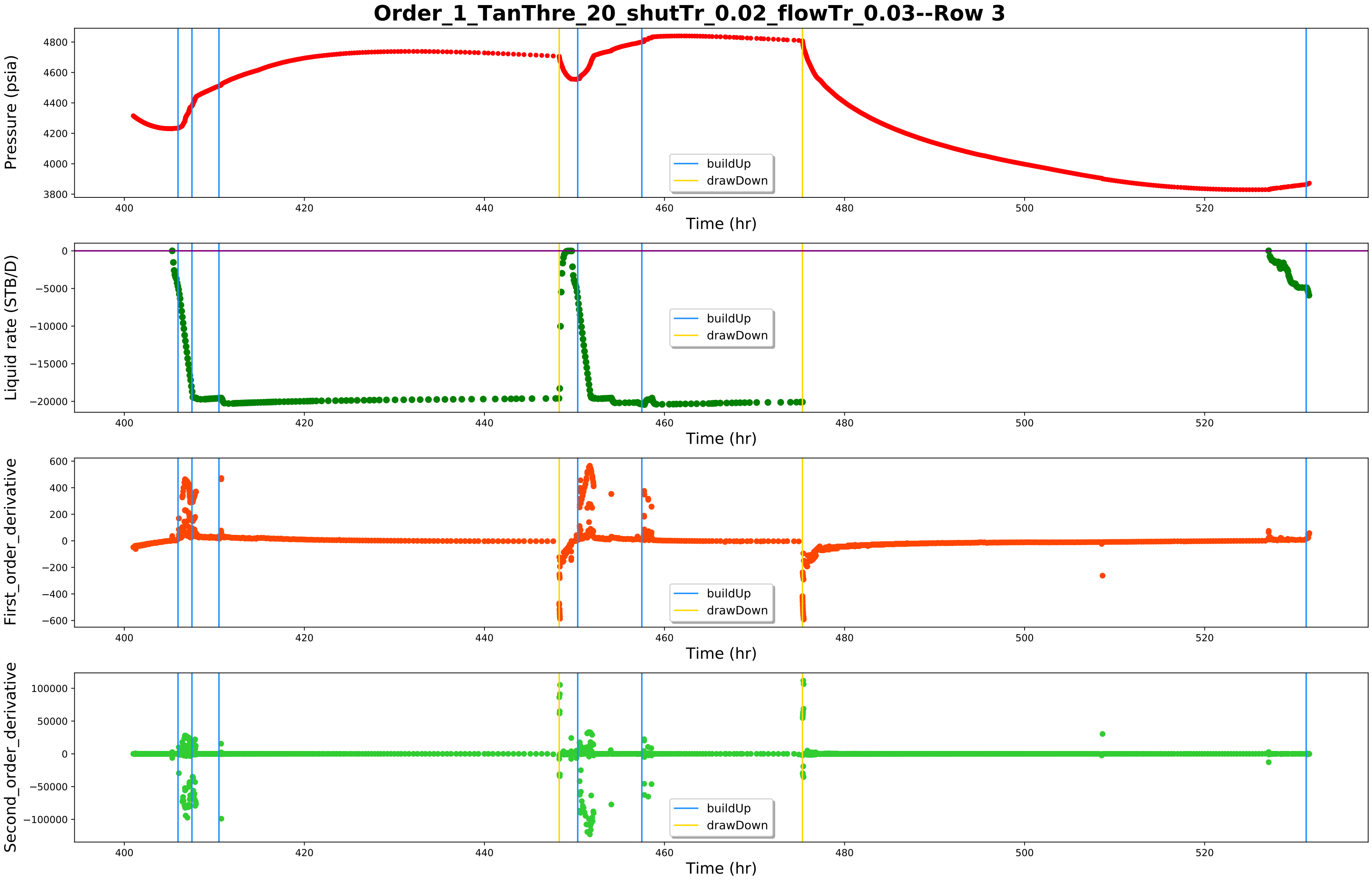
The best result of *Task 1* (sliced version)



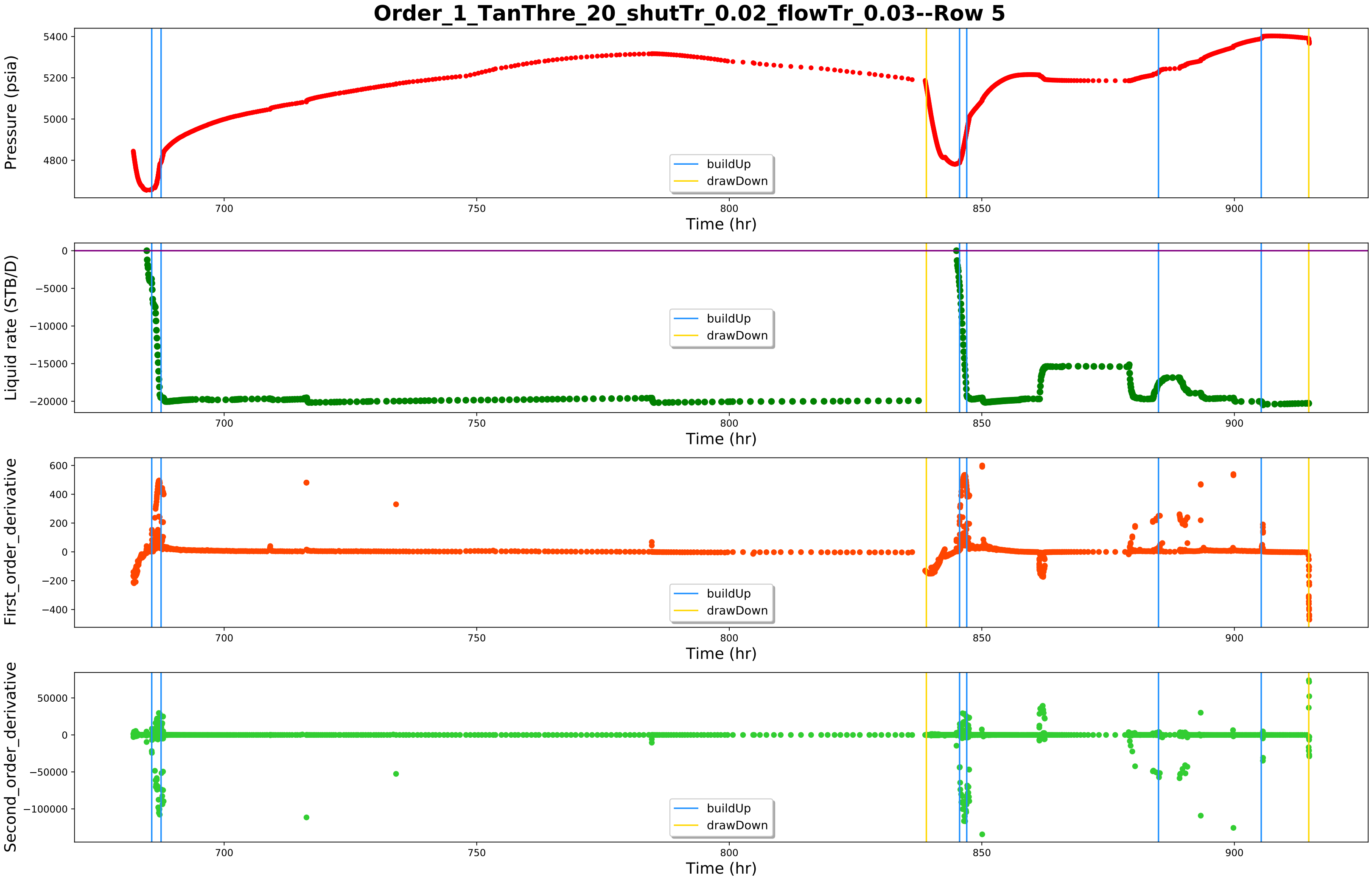
Appendix D

The best result of ***Task 2*** (sliced version)

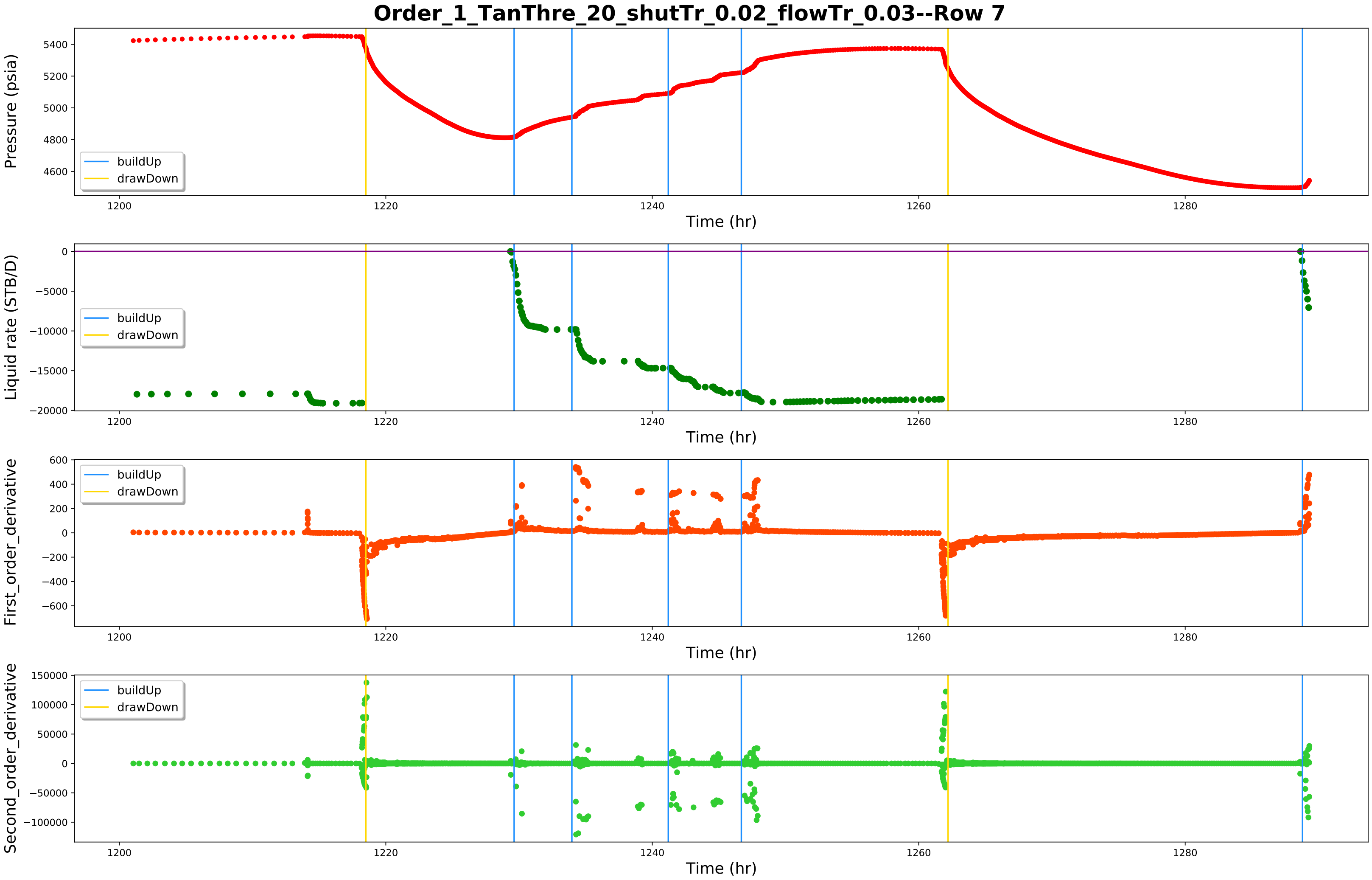
The best result of *Task 2* (sliced version)



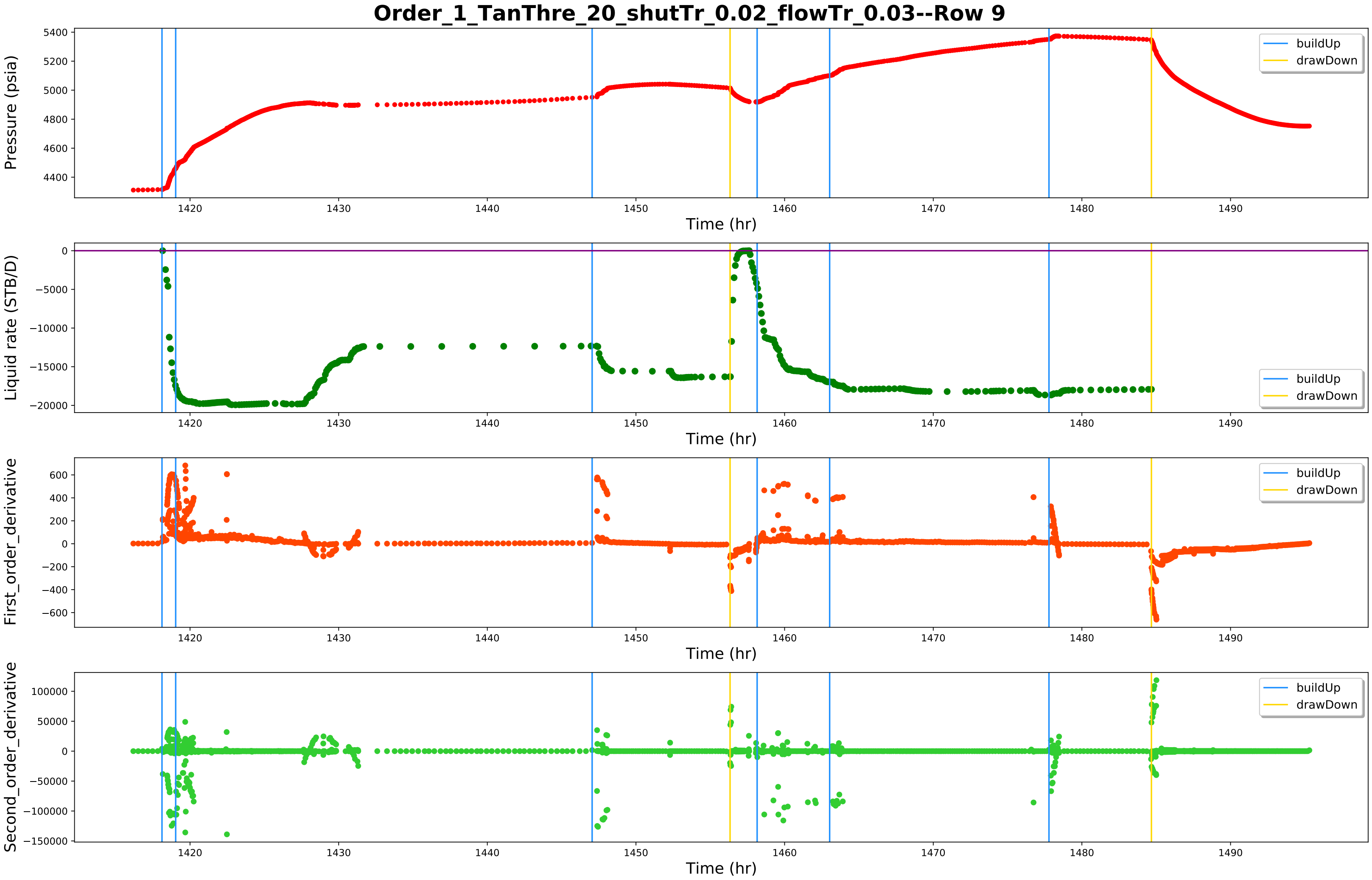
The best result of *Task 2* (sliced version)



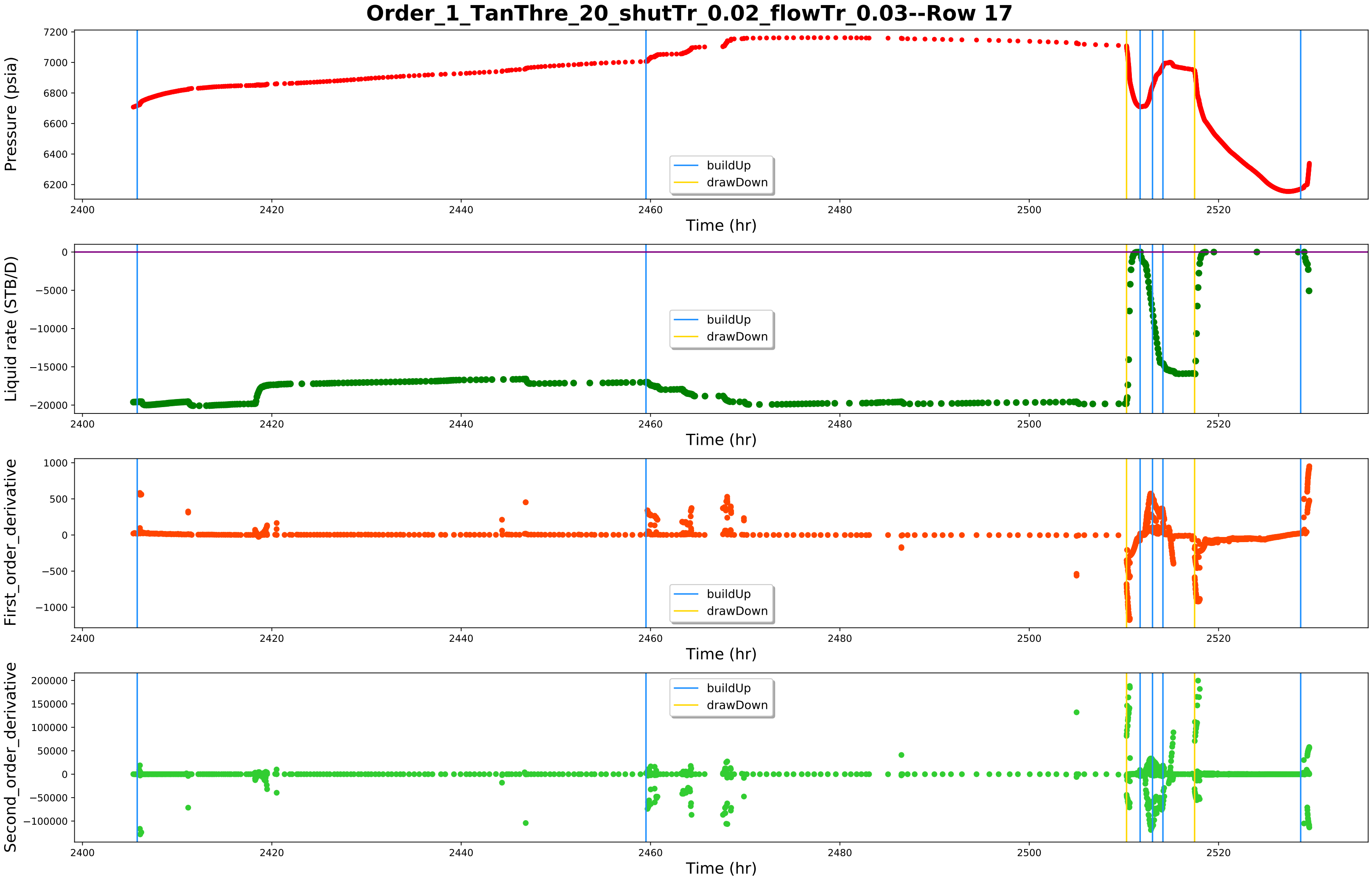
The best result of *Task 2* (sliced version)



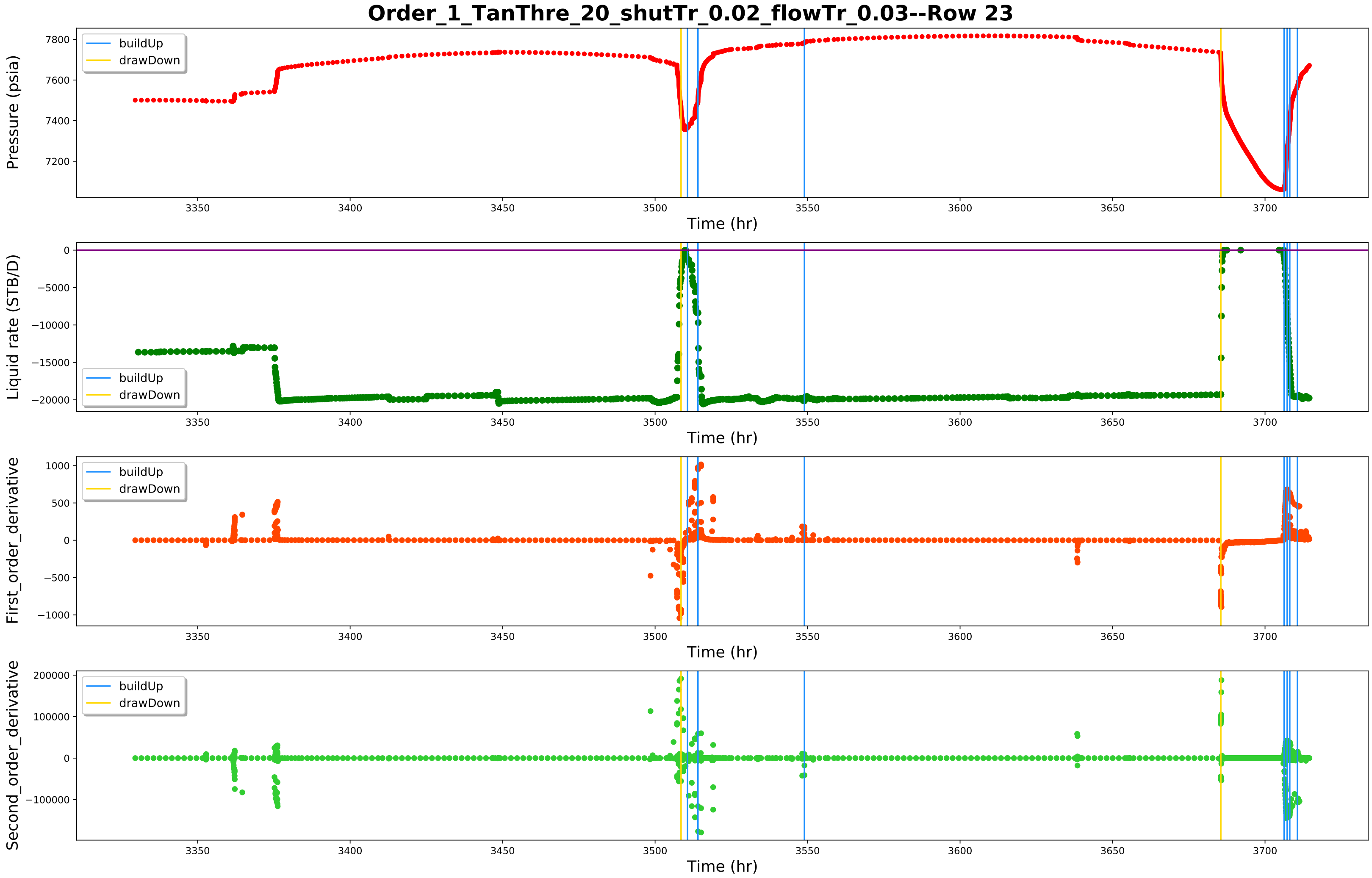
The best result of *Task 2* (sliced version)



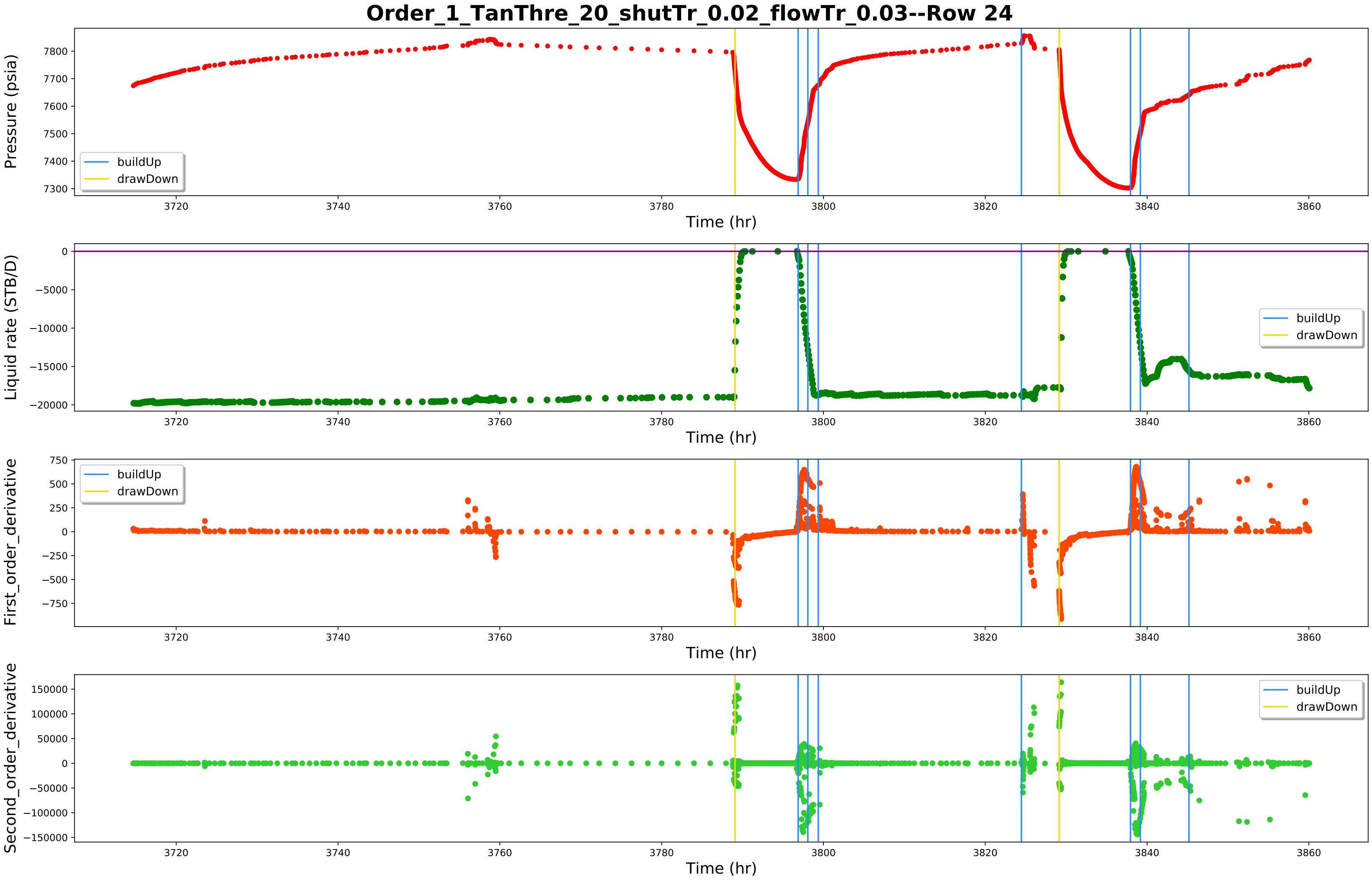
The best result of *Task 2* (sliced version)



The best result of *Task 2* (sliced version)



The best result of *Task 2* (sliced version)



Appendix E

A example of json file for the parameters and detected points indices

```json
{
  "Methods": "DeltaTangent",
  "Denoise": "Yes",
  "Point Window": 10,
  "Polynomial Order": 1,
  "DeltaTangent Threshold": 20.0,
  "Minor Shut-in Threshold": 0.02,
  "Minor Flowing Threshold": 0.02,
  "Number of Shut-in": 31,
  "Number of Flowing": 32,
  "Number of All Build-up Points": 191,
  "Number of All Draw-down Points": 31,
  "Shut-in Periods": [
    [2248, 2477],
    [2804, 2891],
    [3095, 3594],
    [4009, 4228],
    [4750, 4899],
    [5304, 5445],
    [5985, 6543],
    [7284, 7551],
    [8003, 8368],
    [9024, 9606],
```

A example of json file for the parameters and detected points indices

```
    [10149, 10224],
    [10527, 10800],
    [11340, 11480],
    [12362, 13038],
    [13909, 14089],
    [14477, 14575],
    [14774, 15345],
    [16096, 16583],
    [17589, 18433],
    [19642, 19761],
    [20035, 20344],
    [20700, 21034],
    [21974, 22189],
    [22728, 23230],
    [24215, 24679],
    [25750, 26000],
    [26708, 26767],
    [27043, 27324],
    [27799, 27969],
    [28290, 28493],
    [28876, 29112]
  ],
  "Flowing Period & Breakpoints in Flowing": [
    {
      "Flowing Period": [0, 2248],
      "Breakpoints in Flowing Period": [
        55, 434, 604, 820, 1033, 1358, 1811, 2111
      ]
    },
    {
      "Flowing Period": [2477, 2804],
      "Breakpoints in Flowing Period": [2539, 2601, 2658]
    },
    {
      "Flowing Period": [2891, 3095],
      "Breakpoints in Flowing Period": [2950, 3022]
    },
    {
      "Flowing Period": [3594, 4009],
      "Breakpoints in Flowing Period": [3646, 3689, 3901]
    },
    {
      "Flowing Period": [4228, 4750],
      "Breakpoints in Flowing Period": [4305, 4392, 4491]
    },
    { "Flowing Period": [4899, 5304], "Breakpoints in ...
        Flowing Period": [4981] },
    {
      "Flowing Period": [5445, 5985],
```

A example of json file for the parameters and detected points indices

```
     "Breakpoints in Flowing Period": [5498, 5555, 5833, ...
       5943]
    },
    {
     "Flowing Period": [6543, 7284],
     "Breakpoints in Flowing Period": [6771, 6865, 7038, ...
       7137]
    },
    {
     "Flowing Period": [7551, 8003],
     "Breakpoints in Flowing Period": [7621, 7682, 7773, ...
       7845]
    },
    {
     "Flowing Period": [8368, 9024],
     "Breakpoints in Flowing Period": [8426, 8599, 8660, ...
       8774, 8867]
    },
    {
     "Flowing Period": [9606, 10149],
     "Breakpoints in Flowing Period": [9663, 9709, 10064]
    },
    {
     "Flowing Period": [10224, 10527],
     "Breakpoints in Flowing Period": [10330, 10464]
    },
    {
     "Flowing Period": [10800, 11340],
     "Breakpoints in Flowing Period": [10870, 10946, ...
       10998, 11081]
    },
    {
     "Flowing Period": [11480, 12362],
     "Breakpoints in Flowing Period": [11589, 11668, ...
       11744, 12240]
    },
    {
     "Flowing Period": [13038, 13909],
     "Breakpoints in Flowing Period": [
      13100, 13147, 13190, 13268, 13331, 13395, 13526, ...
        13600, 13666, 13774
     ]
    },
    {
     "Flowing Period": [14089, 14477],
     "Breakpoints in Flowing Period": [14143, 14184, 14236]
    },
    {
     "Flowing Period": [14575, 14774],
```

A example of json file for the parameters and detected points indices

```
      "Breakpoints in Flowing Period": [14662]
    },
    {
      "Flowing Period": [15345, 16096],
      "Breakpoints in Flowing Period": [
        15401, 15460, 15573, 15738, 15865, 15938, 16007
      ]
    },
    {
      "Flowing Period": [16583, 17589],
      "Breakpoints in Flowing Period": [
        16638, 16697, 16750, 16804, 16901, 17063, 17127, ...
            17244, 17341, 17462
      ]
    },
    {
      "Flowing Period": [18433, 19642],
      "Breakpoints in Flowing Period": [
        18480, 18523, 18588, 18642, 18715, 18775, 18809, ...
            18858, 18892, 18944,
        19028, 19071, 19147, 19204, 19457, 19556
      ]
    },
    {
      "Flowing Period": [19761, 20035],
      "Breakpoints in Flowing Period": [19854, 19904]
    },
    {
      "Flowing Period": [20344, 20700],
      "Breakpoints in Flowing Period": [20410, 20457, 20524]
    },
    {
      "Flowing Period": [21034, 21974],
      "Breakpoints in Flowing Period": [
        21085, 21114, 21171, 21213, 21420, 21513, 21649
      ]
    },
    {
      "Flowing Period": [22189, 22728],
      "Breakpoints in Flowing Period": [
        22273, 22329, 22406, 22488, 22555, 22620
      ]
    },
    {
      "Flowing Period": [23230, 24215],
      "Breakpoints in Flowing Period": [
        23283, 23312, 23354, 23393, 23564, 23638, 23771, ...
            23923, 23979, 24068,
        24161
```

A example of json file for the parameters and detected points indices

```
      ]
    },
    {
      "Flowing Period": [24679, 25750],
      "Breakpoints in Flowing Period": [
        24754, 24792, 24872, 24956, 25099, 25196, 25272, 25342
      ]
    },
    {
      "Flowing Period": [26000, 26708],
      "Breakpoints in Flowing Period": [
        26052, 26078, 26121, 26177, 26289, 26540
      ]
    },
    {
      "Flowing Period": [26767, 27043],
      "Breakpoints in Flowing Period": [26810, 26834, 26921]
    },
    {
      "Flowing Period": [27324, 27799],
      "Breakpoints in Flowing Period": [27376, 27414, ...
          27462, 27546]
    },
    {
      "Flowing Period": [27969, 28290],
      "Breakpoints in Flowing Period": [28033, 28073, ...
          28138, 28227]
    },
    {
      "Flowing Period": [28493, 28876],
      "Breakpoints in Flowing Period": [28594, 28653, 28784]
    },
    {
      "Flowing Period": [29112, 29784],
      "Breakpoints in Flowing Period": [
        29162, 29222, 29299, 29364, 29477, 29715
      ]
    }
  ]
}
```

Appendix F

The links for the outcomes produced but not included in this thesis

The deployed app: click **here**.

The GitHub repository of the scripts: click **here**.

Full version of manual interpreted result of Task 1: click **here**.

Full version of best result of Task 1: click **here**.

Full version of best result of Task 2: click **here**.