SPE-209127-MS

Efficiency and Effectiveness - A Fine Balance: An Integrated System to Improve Decisions in Real-Time Hydraulic Fracturing Operations

Somnath Mondal, Shell Exploration & Production Co.; Ashan Garusinghe, Shell Canada Ltd.; Sebastian Ziman, Muhammed Abdul-Hameed, Rakesh Paleja, Matthew Jones, Jan Limbeck, and Bryce Bartmann, Shell Global Solutions International B.V.; Jeremy Young, Kent Shanley, Bonner Cardwell, and Humphrey Klobodu, Ely & Associates; Paul Huckabee, Gustavo Ugueto, and Christopher Ledet, Shell Exploration & Production Co.

Copyright 2022, Society of Petroleum Engineers DOI 10.2118/209127-MS

This paper was prepared for presentation at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, 1 - 3 February 2022.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of SPE copyright.

Abstract

Hydraulic fracturing is a key driver of well productivity and field development planning, in addition to being the most significant portion of capex in shales. Recent breakthroughs in connectivity and digital technologies have enabled the monitoring and analyses of frac operations in real-time. However, most of the digitalization effort to date has been focused on increasing operational efficiency to reduce cost. Without an equal consideration for creating effective fracture geometries, this may lead to poor resource recovery and leave significant value behind. In this paper, we - 1) demonstrate the need to balance between optimizing fracture efficiency and effectiveness; 2) present an integrated system for frac optimization using real-time, historical data along with organizational knowledge; and 3) discuss the challenges of setting up such a system and key considerations, along with examples of large, untapped potential that can be unlocked with data science to deliver real value.

Currently, several service providers exist to stream frac data with interactive analytics dashboards. While they offer some customizability, most do not provide a true frac optimization platform that goes beyond frac monitoring and analytics geared towards efficiency and cost indicators. We are still dependent on an individual operator's experience and rules of thumb to make job decisions during a frac stage. In this paper, a real-time optimization workflow is presented that uses advanced data science and statistical techniques to interpret and predict time-series treatment data, integrate historical and contextual information, and honor basin-specific knowledge that has been gathered and tested over the years.

Examples are presented from diagnostic pads that highlight the need for balancing stimulation effectiveness with efficiency. We demonstrate a platform to host and execute an ensemble of models and visualizations that communicate actionable insights to an operator within minutes of identifying an event, gather feedback, and learn. Results from field testing show that our system accelerates the learning curve, enables consistent decision making by operators, and can generate significant cost savings. Finally, we share learnings from our digitalization journey.

Completion and stimulation expenses account for approximately half of an unconventional well cost. Automated decision making for real-time fracture treatment is the holy grail of digital completions in

shales. However, a blind pursuit of efficiency may lead to sub-par fracture treatments and significant value erosion for shale assets. We present an integrated framework that connects real-time data and organizational knowledge to guide an operator to pump the best frac stage while reacting to formation response within a set of constraints. To the best of our knowledge, this is the first paper to describe the general architecture and demonstrate the viability of such a system that relies only on standard wellhead measurements during fracturing.

Introduction

The growth in US oil and gas production since the early-2000s has been driven by hydraulically fractured horizontal wells producing from source rocks, also known as the 'Shale Revolution'. Hydraulic fracturing is a key driver of well productivity and field development planning, in addition to being the most significant portion of capex in shales (Figure 1). Several market research reports also forecast that the global hydraulic fracturing market size will grow significantly to over USD 38 billion by 2027 from USD 28 billion in 2019, at a compounded annual growth rate of over 8%, driven primarily by adoption of this technology in new markets outside North America and offshore (Business Insights Report, 2020). As such, any opportunity to improve or optimize fracturing may have wide reach and high value impact.

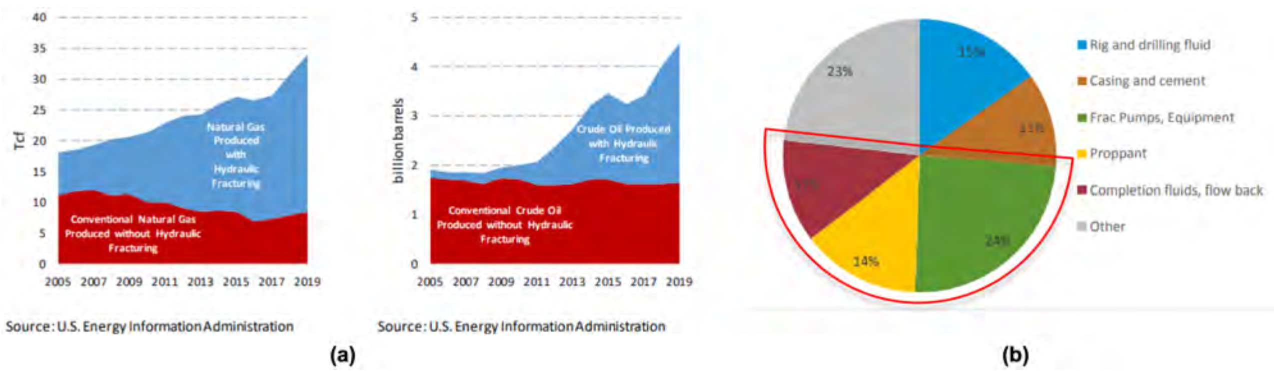

Figure 1—(a) US Dry Gas and US Oil production (2005-2019) showing hydraulically fractured wells accounted for the entire increase in hydrocarbon production in the US (DOE, 2021) (b) Breakdown of cost for US onshore oil and natural gas drilling and completion showing hydraulic fracturing related costs (frac pumps, equipment, proppant, completion fluids, and flowback) comprise approximately 50% of the capital expenditure in shales (EIA, 2016).

Recent breakthroughs in connectivity and digital technologies have enabled the monitoring and analyses of hydraulic fracturing (frac) operations in real-time. The introduction of 'edge' computing devices coupled with low latency data streaming to cloud servers has spurred a host of 3rd party data and analytics providers to offer real-time remote frac monitoring solutions. Most of these solutions offer real-time data streaming from the frac van, wireline cab, and pumpdown truck, etc. using a physical 'edge' device(s) on location that connect to the cloud through a cellular or on-site internet connection. The main customers of these solutions are operations staff and management, completions engineers, frac consultants, and company management.

The 'frac digitalization' effort by the industry to date has been focused on increasing operational efficiency to reduce cost. Currently, the 3rd party frac monitoring solutions are focused on real-time data streaming and analytics where most of them offer a real-time frac treatment data chart, post-stage frac analytics, and key operational efficiency metrics. Most do not provide a true frac optimization platform that goes beyond frac monitoring and efficiency/cost analytics.

Without consideration for creating effective fracture geometries and good stimulation distribution, this may lead to poor resource recovery and leave significant value behind. Below are two field examples of stages that demonstrate the need to balance between optimizing fracture efficiency and effectiveness.

Figure 2 below shows a stage where there was room to increase rate and sand concentration earlier in the stage and about 5 to 10 mins of pump time could have been saved by reacting early. Historical analyses of stages showed that such missed opportunities due to late or inconsistent reaction are common.

Figure 2—A typical frac treatment stage showing an opportunity where rate and/or sand concentration may have been increased earlier in the stage by responding quickly as pressure stabilized and fell below upper limit.

Figure 3 shows a stage with 6 perforation clusters from an optic fiber diagnostic well. Shortly after proppant slurry is pumped through the clusters, there is a decreasing surface treating pressure trend likely caused by perf erosion and reduction in initial tortuosity. Slurry rate however is not increased during this time to utilize the opening pressure headroom. The optic fiber data from distributed acoustic sensing (DAS) measurements show that the heel cluster is lost due to screenout, likely due to insufficient rate to maintain limited entry pressures. Similar behavior was observed in other stages especially extended stages where maintaining rate was critical to keeping perf clusters active throughout a stage. In this example, increasing rate appropriately as soon as the treatment pressure dropped may have prevented the early screenout and improved stimulation distribution effectiveness.

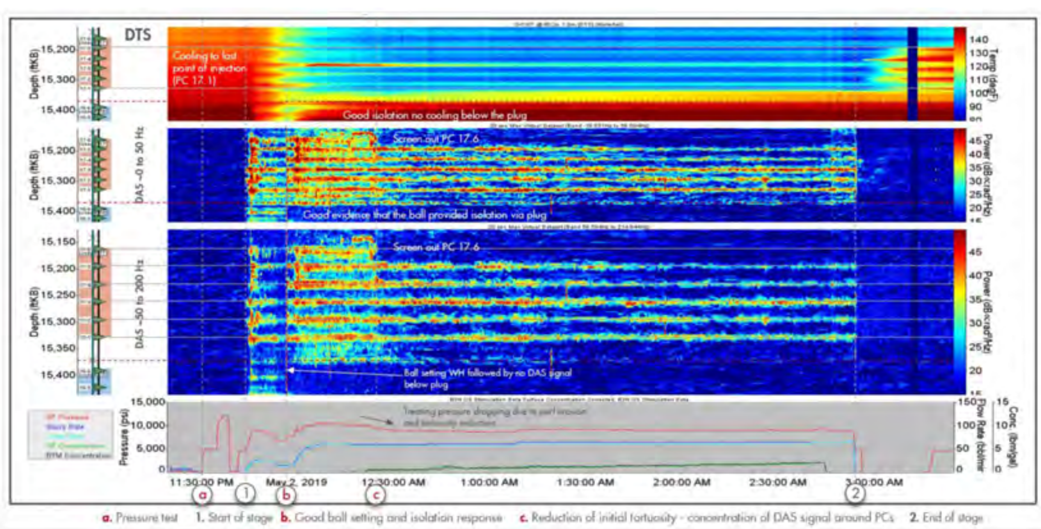

Figure 3—A typical frac treatment stage from a well with Distributed Fiber Optic measurements, showing early heel cluster screenout likely because of insufficient rate to preserve limited entry as perf cluster erodes from proppant slurry. Increasing rate appropriately as soon as the treatment pressure dropped may have prevented the early screenout and improved stimulation distribution effectiveness.

In other cases, integrated optic fiber diagnostics data have also shown that perf cluster screenout signatures can sometimes be observed in the surface treatment data. Historical analysis of surface treatment data has revealed that perf cluster screenout signatures are widely observed in other basins (Figure 4). Cluster screenouts, especially early ones, lead to severe non-uniformity in stimulation distribution. These examples demonstrate the need for including effectiveness considerations in any real-time fracture optimization algorithm.

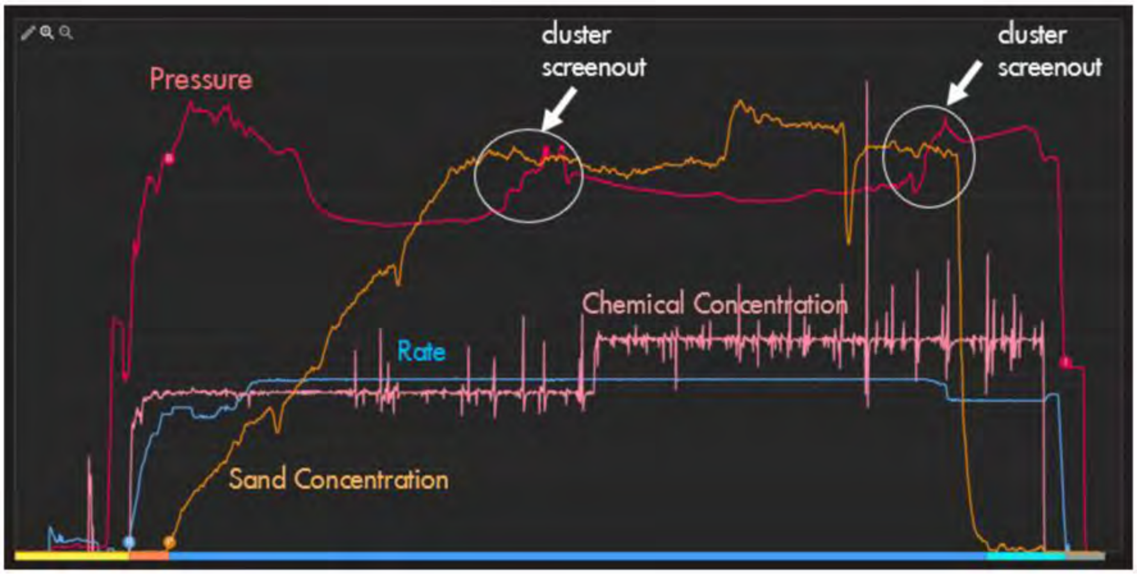

Figure 4—Example of cluster screenout signature from surface treatment data in a well. Integrated optic fiber diagnostic data from several basins have confirmed that such patterns are associated with perf cluster screenout.

The CART (Completions Automation and Remote Technology) Center was set up in 2017 to remotely monitor and optimize all Shell Shales fracturing operations across multiple basins. The CART is staffed by experienced frac consultants that have decades of experience in frac pumping, treatment design and deep basin knowledge and their main role is to provide input directly to the frac engineer on-site to optimize each frac treatment. Currently, this remote frac monitoring and optimization model relies on the experience of the CART frac consultant to know when and where to provide the right input. There's inherent variability in how each frac consultant and on-site frac engineer view frac optimization opportunities and what actions they take. Also, since CART is monitoring multiple frac operations at once, it relies heavily on on-site frac engineers to identify and respond to real-time frac signals while CART consultants focus more on post-stage analysis and recommendations.

Literature Review

While multiple tools exist for real time monitoring of fracturing data, the landscape of optimization is sparse. What work has been done is often for the purpose of improving frac design and auto summarization for historical analysis. For instance, Paryani et al., (2018) shows how subsurface data gathered during the drilling process can be applied to generate completion designs in real time, assisting engineers in decision making ahead of time. Similarly, there has been significant investigation into how data analysis can be used to improve frac efficiency. Morozov et al. (2020) details how machine learning can be used to perform historical analysis to improve frac design to boost production.

Computational intelligence methods have also been used to solve prediction and classification problems in completions and production. Xiong and Holditch (1995), proposed the use of fuzzy logic systems to improve stimulation design and productivity in oil and gas wells. Artificial Neural network models were developed by Chen, Rahman and Sarma (2014) to predict the interaction between hydraulic fractures and natural fractures to improve treatment design parameters in a gas reservoir. Artificial neural networks were also used to estimate production in hydraulic fractured wells (Hassan et al., 2020). In today's fracturing landscape, completions engineers most often interact with frac data during a job through cloud-based platforms, that perform automatic analysis and visualization of real-time data. For instance, the work of Shen et al. (2020) shows how event recognition and classification can be used to identify key parts of a fracturing job; demonstrating how this can be integrated into a cloud-based platform and used to speed up analysis. They also describe how machine learning techniques combat challenges of real time data, like automatic detection of the start and end of a fracturing stage in real time, a key step when using any model that is sensitive to its initialization point. However, Shen et al. (2020) aims to complete these challenges for the purpose of improving and speeding up historical analysis and deviates away from the idea of real time optimization.

Despite this, work has been done to extend these automated analyses to include predictive analytics. For example, the work of Ben et al. (2020a) identifies a workflow to model the wellhead pressure using control variables via a deep learning model that is trained on real-time data, providing a forecasting window of a few minutes. They later extend this by using pre-defined pumping schedules to predict the wellhead pressure and implement it into a cloud-based platform. Ben et al. (2020b) later continued this work by demonstrating a workflow for real-time optimization using Model Predictive Control (MPC). By modelling the pressure using a dynamical system, they use MPC to vary the control variables in real time to minimize the calculated cost of the stage. This demonstrates a significant advancement by integrating pressure prediction into an optimization routine; frequently pumping costs are dependent upon the average pressure for the stage, thus making it necessary to understand pressure response to stimuli.

Methods, Procedures and Processes

In this section we lay out our model framework, illustrating the methods involved and some of the models implemented.

Data Pre-processing

Modularity is a key motivator of our framework. We aim to have a plug-and-play system where different models can be inserted and removed from the routine with no interdependency (Figure 5). In this vein the first step after retrieving new data is to process the data, calculating important statistics used in historical analysis, such as first order derivatives of control variables. During this step we also include a step for quantifying important statistics such as time the sand was started, and the first-time sand hits the formation. We also perform calculations to generate statistics used within the models, for instance the injectivity or derivatives of control variables. We can then sequentially apply our models within this framework with no problems with interdependency.

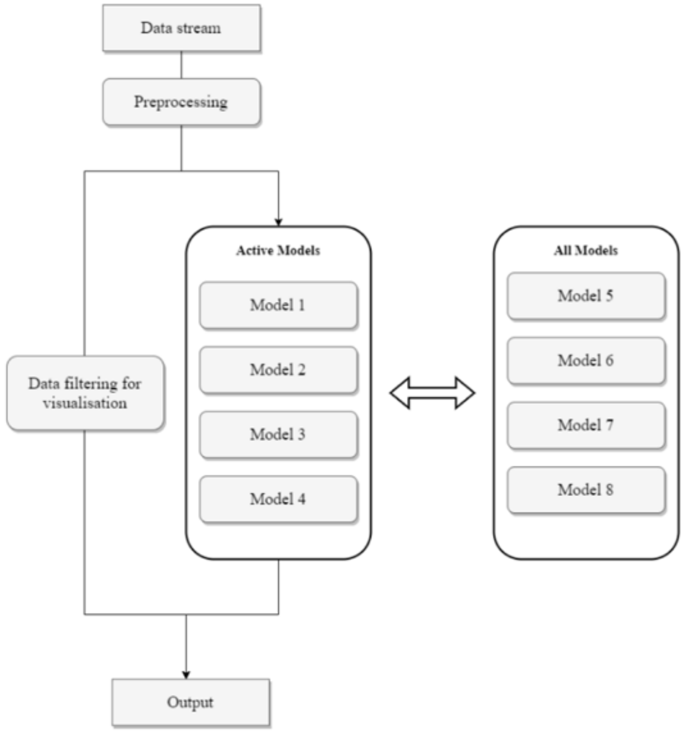

Figure 5—A diagram of the model framework implemented, demonstrating the plug and play nature of the structure. All models currently active are executed serially, but independently so no compatibility issues arise.

As the data is fed into the framework, we also perform automated analysis to detect the start and end of a stage in real-time, similar to Shen et al. (2020). This is crucial since we introduce models that are sensitive to the time they are instantiated, making it vital to identify the start accurately. However, instead of opting for a complex deep learning algorithm like Shen et al. (2020), we opted to implement a simpler rule-based model to improve operational efficiency of the tool. Therefore, at each new timestamp we look to see whether our injection rate and pressure are high enough to suspect that fracturing operations have commenced; should the flow rate fall back to near zero the framework will reset itself if this was simply a testing period of the stage, or counts the stage as ended if the total sand pumped is more than the minimum required for the stage.

Easy or Hard Stage

A fracturing stage can be 'Easy' or 'Hard' depending on how easily injectivity can be achieved and maintained. Engineers try to classify a stage as early as possible to decide how readily a stage will accept sand. We present a workflow to assist engineers in the early classification of these stages. This 'Easy vs Hard' problem is not a true classification problem, as frequently stages can be 'in between' and a well that is initially 'Hard' can become 'Easy' etc. To account for this, we introduce a score function p_t that tracks the probability of a stage being 'Easy' at time t. We define our score function as follows

$$p_0 = \frac{1}{2},$$
$$p_t = p_{t-1}^{\exp(-f(x_t))} \quad for\, t > 0.$$

where, X_t is the current state of the well at time t.

This construction allows us to define f naturally such that when f is positive then p increases and when f is negative then p decreases. Following this f is described by

$$f(X_t) = \alpha I_t - \beta\left(1 - \hat{Q}_t\right)$$

where I_t is the injectivity, defined as the ratio between the injection rate and wellhead pressure and $\hat{Q}_t$ is the normalized injection rate with α and β being scalar parameters.

These two parameters can be stage specific however default values were chosen based on experimental analysis of historical data. This means we reward for achieving injectivity earlier and punish when the injection rate is kept low. As shown in Figure 6, this model allows an engineer to get a consistent, unbiased, early indication of stage performance enabling less variability in operator reaction and rate or sand ramp-up decisions.

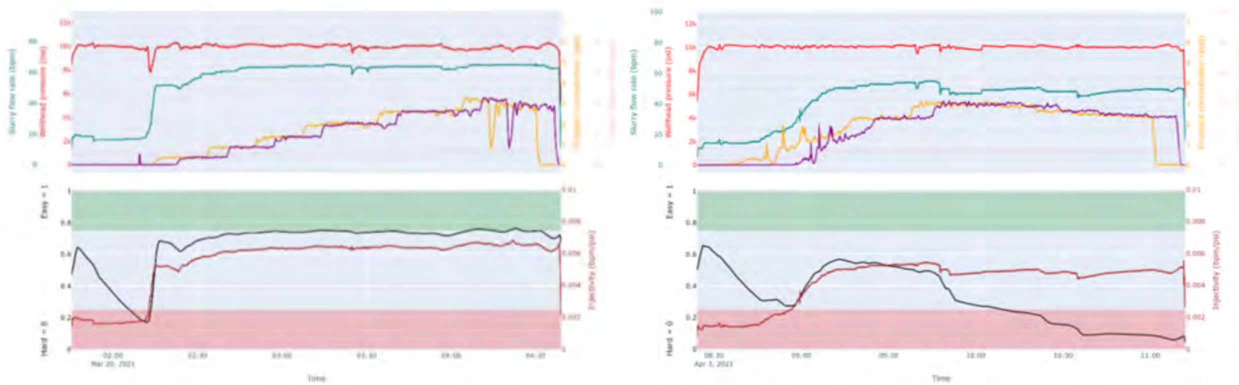

Figure 6—Examples demonstrating how the Easy vs Hard model can adapt with time. The top panel shows the treatment data, and the bottom panel shows the injectivity (brown) and a probability of easiness on a 0 to 1 scale (black) for two example stages (left and right), respectively. Here we can see that initially the left-hand stage looks hard as the pressure holds steady and rate cannot be increased, driving down the probability, demonstrating the penalty system. However, once breakdown occurs the rate can be increased easily, rewarding the model and increasing the probability, eventually settling to a relatively easy score throughout the stage. The right-hand stage on the other hand starts poorly, and then looks like it might become easier, but the inability to increase the rate means the stage continues being hard, and the model continues to penalise.

Breakdown detection

Breakdown identification is a key task for the on-site frac engineer and CART, as they try to spot signs of the subsurface rock fracturing, and quickly respond. We developed a method for identifying potential breakdowns while they occur, based on a physics-based algorithm in collaboration with fracturing SMEs and completions engineers. We designed the algorithm to suspect that a breakdown might be starting when the pressure begins to decrease while the injection rate remains constant or is increasing, and then conclude the breakdown when the pressure levels off. There is then a post processing step where the breakdown is checked to ensure whether it was a true breakdown or not, by ensuring that there was sufficient change in injectivity which lasted for a sufficient duration. Figure 7 below shows a typical stage treatment data with breakdown periods identified and displayed using vertical shaded bands.

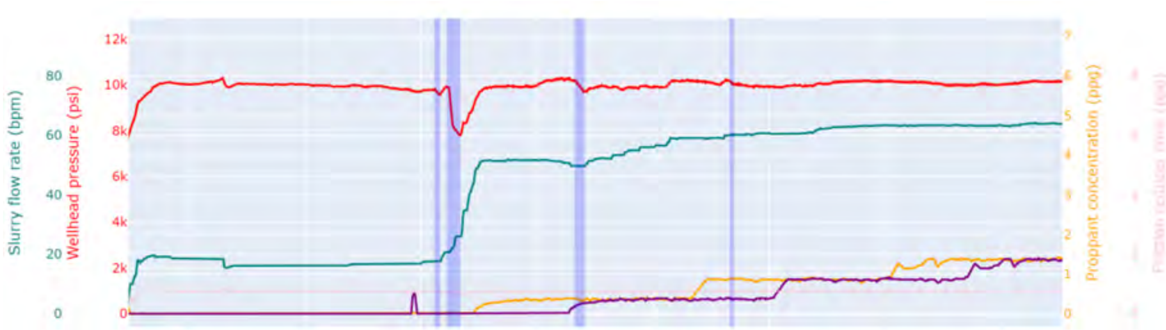

Figure 7—Demonstration of the breakdown identification displayed on treatment charts using shaded vertical bands.

Optimization

Finally, we present a full optimization routine to minimize the cost structure (Figure 8). Due to practical limitations, all optimizations must be implemented manually by the engineers, thus restricting us to reasonable recommendations (for instance suggesting adjusting the injection rate by <1 bpm would be impractical for an engineer to implement). This effectively reduces our problem to a discrete optimization problem. Therefore, using previous methods such as Model Predictive Control (MPC) are inappropriate as we cannot implement changes rapidly enough or fine enough to leverage the power of MPC. Thus, we instead approach the problem by attempting to identify potential optimization opportunities and then identifying valid optimizations at these opportunity times. The potential operating states are constrained by a set of hierarchical rules defined by subject matter experts, learnings from integrated diagnostics pilots, basin specific frac engineer knowledge, etc.

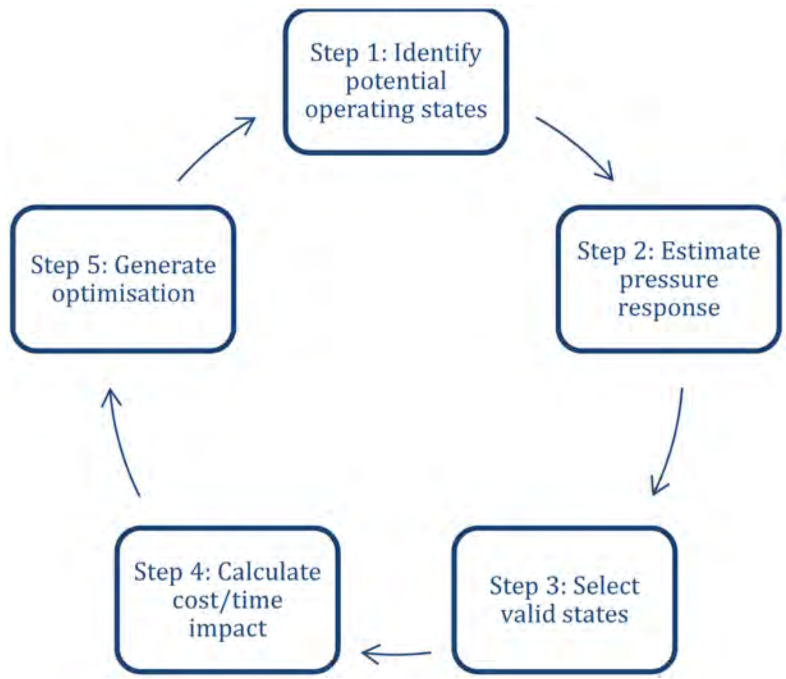

Figure 8—A flowchart of the real-time frac optimisation routine.

Pumping costs for fracturing stages are often calculated retroactively using the average pressure and average injection rate for the stage; to perform effective cost analysis we must understand the impact any optimization will have on our pressure and injection rate. What's more, a fracturing well has other constraints that must be adhered to, for instance the pressure and injection rate must not exceed certain values to preserve well integrity etc. Collectively these constraints describe our state space for optimization. Note that 'effectiveness' considerations are also embedded in Step 1 and Step 3.

Pressure Prediction. To estimate the cost impact and understand the viability of potential optimizations we need to understand the pressure response. We investigated two main methods for pressure prediction, one using a dynamic linear model and another one using a supervised learning model.

Dynamic Linear Model

We opted to implement a Dynamic Linear Model (DLM) (West & Harrison, 2006). These models are appropriate for modelling scenarios where the underlying system is constantly changing, precisely what is happening during a fracturing stage; as fractures form and propagate, conditions in the wellbore, near-wellbore, and far-field changes, changing the dynamics of the system. The model is essentially a standard linear model whose parameters update every timestep. Because of this, the model is extremely lightweight, improving efficiency which is crucial in a real-time environment. This also broadens the applicability of the model; it can be applied to any well of any design with any subsurface geometry as the only input required is real-time data. Unlike a supervised learning model such as a neural network, DLMs do not require training, meaning a training set does not need to be curated and maintained to ensure validity of the model. A full derivation and explanation of DLM can be found in West & Harrison, 2006 but hinges around the equations

$$y_t = F_t x_t + w_t,\ w_t \sim N(0, W_t)$$
$$x_t = G_t x_{t-1} + v_t,\ v_t \sim N(0, V_t)$$

which are the observation and system equations respectively, where y is a vector of our dependent variable (in this case a scalar value of pressure), x is a vector of the independent variables, F and G are matrices and V, W are covariance matrices describing the noise. Thus, these equations describe a linear model that evolves with time.

This approach performed well on the latter part of the stage after breakdown has been established, when the system change is more gradual. However, in the earlier parts of the stage during the ramp the linearity of the model is its weakness, heavily prone to overfitting and overcompensating for the sudden changes associated with breakdown as seen in Figure 9.

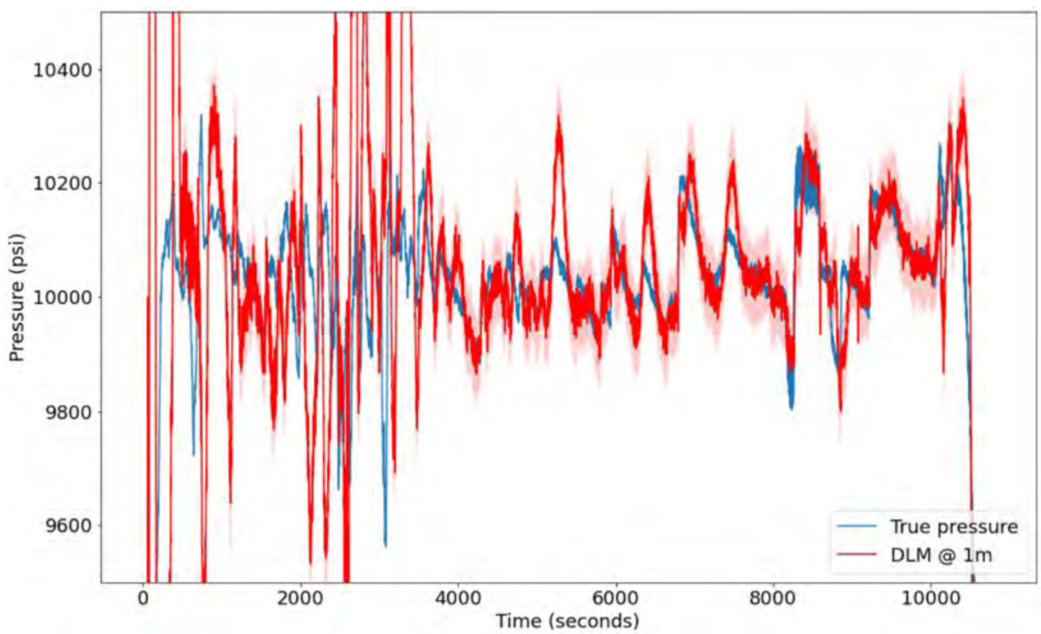

Figure 9—Dynamic Linear Model (DLM) on a sample stage predicting 1 minute ahead, with confidence intervals in shaded band.

This model has its limitations, as it is only a linear model. Frequently we see scenarios where the pressure becomes negatively correlated to the proppant or shows non-linear dynamics due to complex wellbore and near-wellbore physics (wellbore proppant transport, perforation erosion, near-wellbore tortuosity, and cluster-level heterogeneities in rock properties). Table 1 demonstrates this, showing how when we increase the prediction horizon from 1 minute to 3 minutes, the model completely breaks down; the linear model becomes insufficient to model the complexities of the system. Even restricting the prediction to post-breakdown phase of the frac stage is not much better, showing limited predictive power. If we want to push the pressure prediction further than this, an alternative approach is required.

Table 1—A comparison of the performance of the DLM across the whole stage, and performance on the stage after 1 hour (post breakdown).

	DLM @ 1min	DLM @ 3min	DLM @ 1min after 1hr	DLM @ 3min after 1hr
R^2	0.635	-3.026	0.834	0.531
MSE	645,121.65	7,123,314.49	412,154.39	1,197,577.67
rMSE	803.19	2668.95	641.99	1094.34
90% error	552.42	1675.55	338.9	939.38

Table 1 further demonstrates how the performance improves drastically in the post-breakdown (or stable injectivity phase of the stage), after the rate ramp. The R^2 for the model improves significantly post breakdown, showing increased performance. We also see that pushing the prediction window 3 minutes ahead causes the model to completely breakdown, even when considering the 'stable' part of the stage; showing the limitations of a linear model as non-linear effects begin to dominate the behaviour. Looking at the 90th percentile error, calculated by calculating the error between True Pressure and Predicted Pressure at each timestamp and calculating the 90th percentile, we see this is considerably lower than the rMSE, indicating that there are a few exceptionally large errors deviating the rMSE. When the majority of these are removed by considering only the latter sections of the stage, we see improved performance.

Gradient Boosting Model

The second approach we considered was using a Gradient Boosting model (Friedman, 2001), to leverage large quantities of historical data available and allow us to use contextual data like the stage depth. We structured the model to take as input the value of all variables at the current time, the proposed change in the variables and the contextual information about the well and stage, such as the depth or formation type.

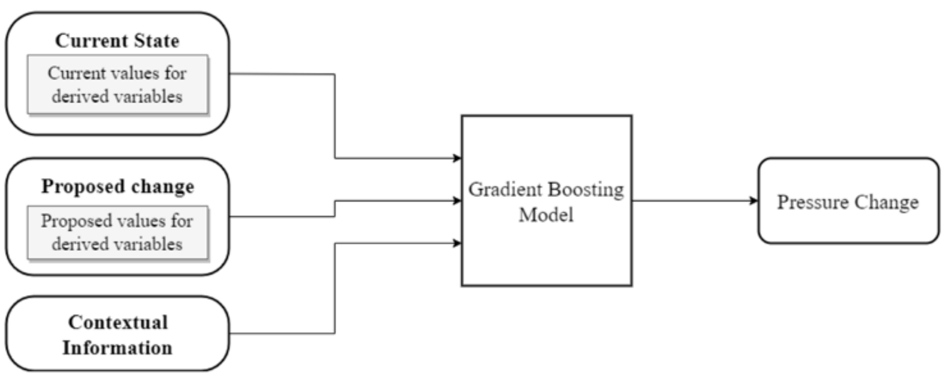

Figure 10—Structure of the Gradient Boosting Model.

The model showed impressive predictive power, able to predict the pressure response up to 3 minutes ahead of the event with reasonable accuracy (Figure 11), whereas the DLM struggled in this forecasting window, as can be seen in Figure 12. However, much like the DLM there is too much variability in the early ramp phase of the stage to accurately predict. While the model only achieves an R^2 of 0.82, we can see the performance on a sample stage is good during the plateau phase but also struggles during the ramp phase of the well much like the DLM.

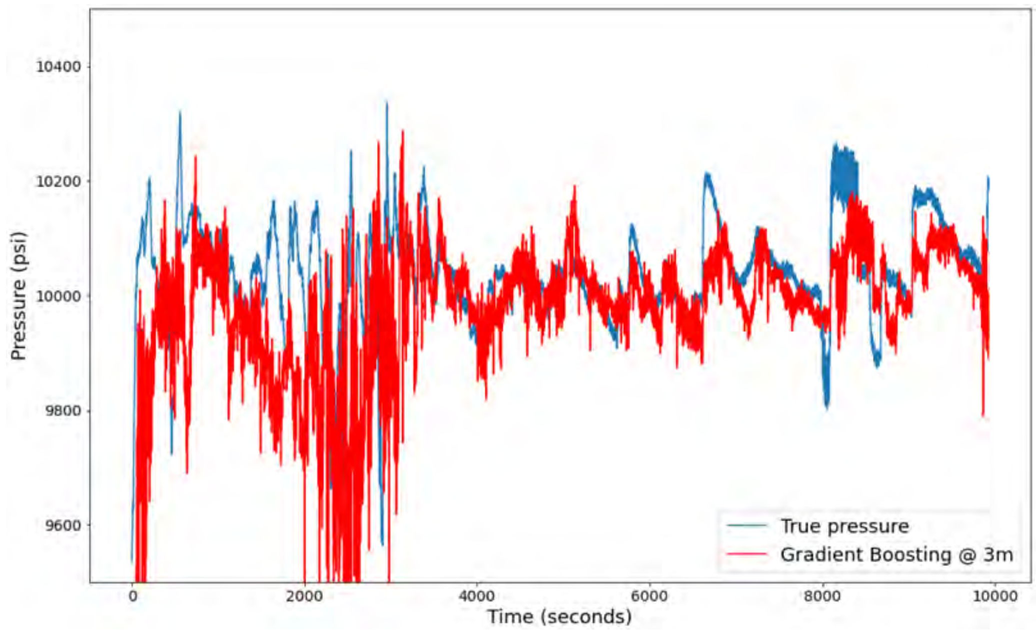

Figure 11—Gradient boosting model predicting 3 minutes ahead.

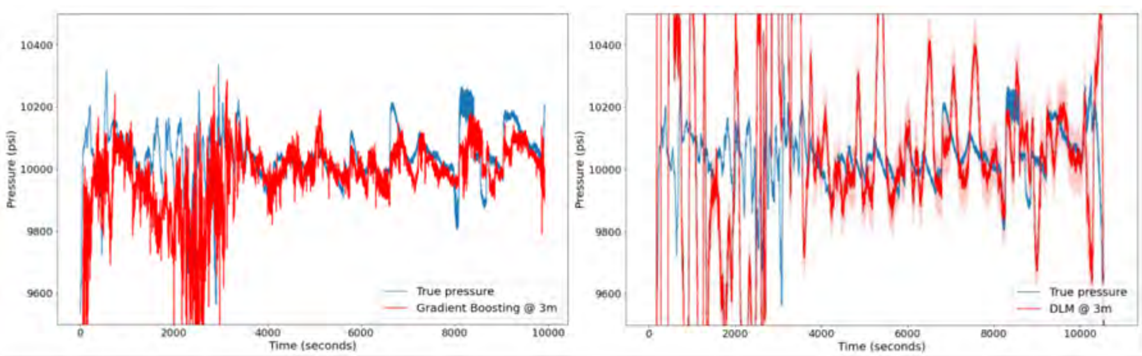

Figure 12—A comparison between the Gradient Boosting models (left) and DLM (right), predicting 3 minutes ahead against the true pressure. This illustrates the increased predictive power of the Gradient Boosting model.

Table 2 demonstrates how the gradient boosting model has significantly increased predictive power over the DLM, accurately predicting 3 minutes ahead, benefiting from its ability to model non-linearity. To establish a baseline, we also compare against a naïve model that predicts zero pressure change at any given time and see that the gradient boosting model has both fewer extremal values and better consistency.

Table 2—A comparison between the DLM (predicting at 1 minute ahead and 3 minutes ahead), Gradient Boosting model and a Naïve model that predicts zero pressure change at all time

	DLM @ 1min	DLM @ 3min	Gradient Boosting @ 3min	Naïve
R^2	0.635	-3.026	0.821	-0.019
MSE	645,121.65	7,123,314.49	61,423.66	349,739.32
rMSE	803.19	2668.95	247.84	591.4
90% error	552.42	1675.55	313.57	376.0

We have so far not included this model into the framework due to implementation challenges. A large gradient boosting model has a significant performance consequence and requires a method to host the sizeable number of parameters associated with it. Also, as we already have mentioned, the curation of a representative training sample to cover all possible well types to fit such a model is a time-consuming process. However, we are investigating the possibility of integrating this model into the live framework to improve predictive power.

Optimization Routine. We first must identify opportunities where an optimization might occur. We do this by looking for points where sufficient pressure headroom is available, pressure trend is favorable, and is sufficiently stable to be modelled. We do this as another measure to ensure optimizations are only given to engineers at times where they are likely to implement them. For instance, if the pressure was already increasing engineers may be unlikely to adopt a change until this subsides.

At a given time step we identify all potential states based on operating constraints using an exhaustive search method (Figure 13). To do this we discretize our state space; identifying the minimum change in a variable that is reasonably achievable within mechanical limitations and the maximum implementable change. We use these to discretize the state space into values within our maximum change that differ by the minimum change. For example, if our current value for a variable is 1.0 and the minimum possible change is 0.1 and maximum change of 0.3, we would generate possible values of (0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3).

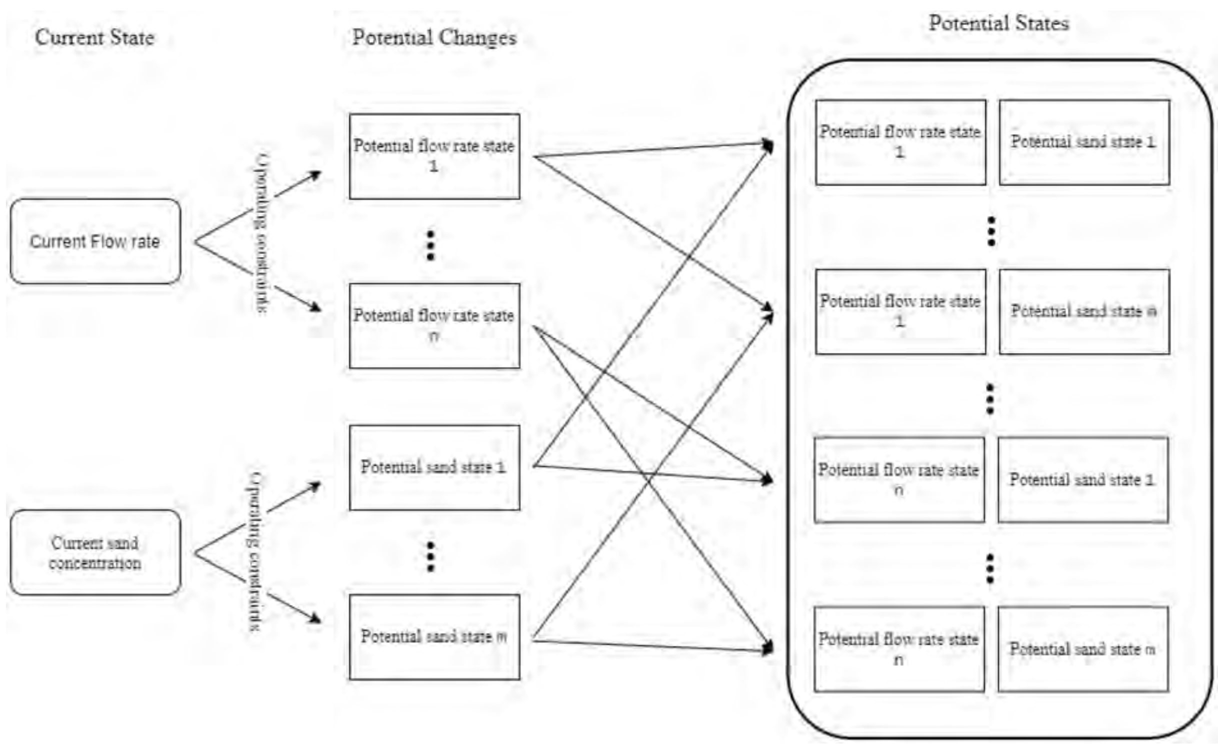

Figure 13—An illustration of the state space search.

For our implementation we only consider optimizations of the injection rate and sand concentration, however this is readily extendable to any set of independent variables. The pressure response is predicted for all these states. Those with a predicted pressure outside the acceptable limit for the well are discarded.

From here we exploit prior knowledge of the required total sand and fluid for the stage to estimate the time to completion; the stage is then 'simulated' under the assumption the control variables (and thus the pressure) do not change further for the duration of the stage. It is now clear that the estimated cost for each possible state can be readily calculated and the most optimal can be selected, completing the optimization routine.

Further work could be done to improve this assumption of constant behaviour until the end of the stage. In particular, the pressure will often vary with constant control variables, for reasons we have already mentioned. We implicitly assume that the pressure does not vary enough to sufficiently affect the cost calculations. We acknowledge this is a limitation of the routine and further work could be done to improve this 'simulation' stage.

Advanced visualisations

Alongside our optimization routine and analytics, we also provide support for a series of advanced visualizations, developed in conjunction with the CART engineers, to provide historical context to current stage performance (Figure 14). This allows engineers to directly compare performance across similar wells and incorporate historical context into their decision making. We do this by automatically assigning each upcoming stage with a list of 'similar' historical stages, which we denote the *Historical Subset*. The user is allowed some flexibility in defining 'similar' historical stages. From there we can display a series of plots to follow performance in comparison to the selected historical stages (Figure 15).

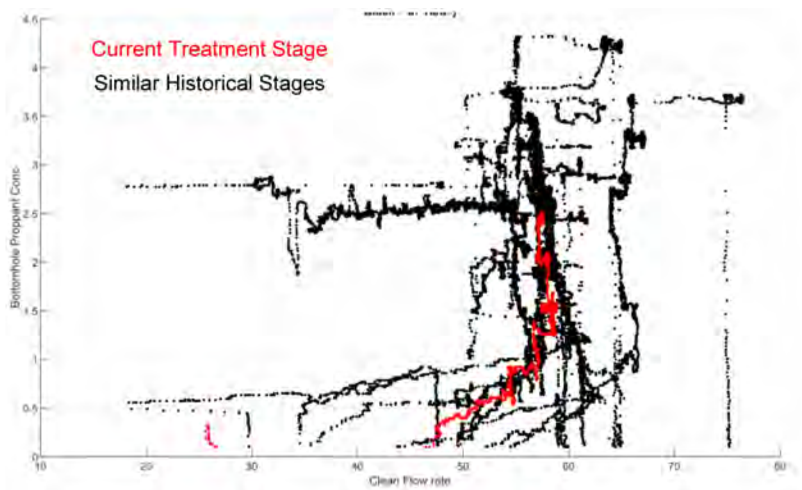

Figure 14—A sample cross plot of injection rate against sand concentration illustrating how historical data can be leveraged to contextualize performance. Current stage marked in red.

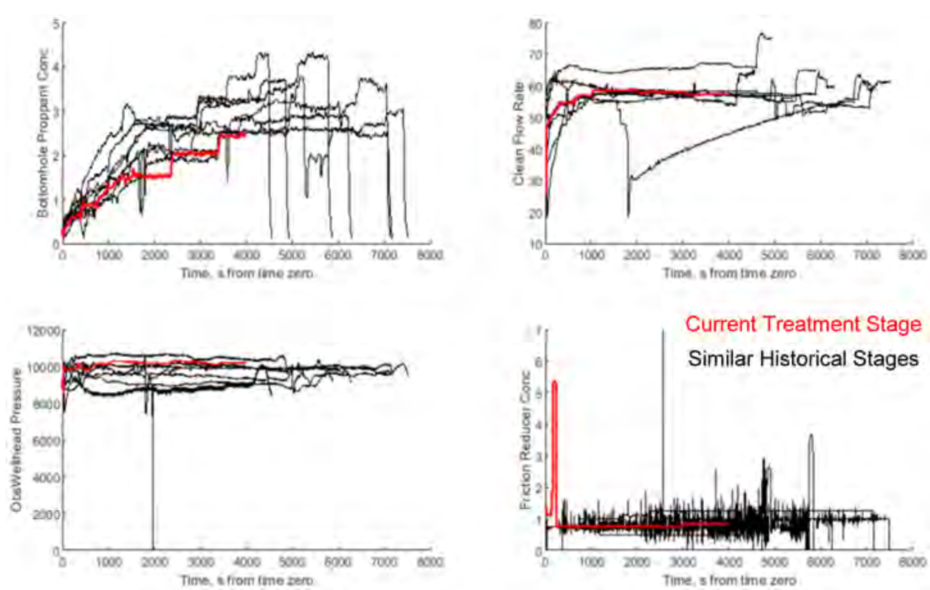

Figure 15—Some further examples of showing the trajectory of dependent variables.

Model Implementation

We have illustrated a procedure to perform automated cost analysis that outputs optimizations and specific actions; in practice however, this may not be feasible to implement due to system limitations. Since automated optimization through closed-loop control is a relatively new technique to the Shales space, the pumping equipments involved are often not automated and any optimization requires manual intervention.

Recommendation Cards. To compensate for this, a recommendation 'card' is constructed for the user, which can be displayed in the app dashboard. This card includes the suggested change as well as contextual information to assist the engineer in decision making, such as average pressure, predicted pressure, and estimated savings should the recommendation be adopted. A sample card is shown in Figure 16. The estimated savings in this case was removed for publication but is typically in thousands of dollars per recommendation.

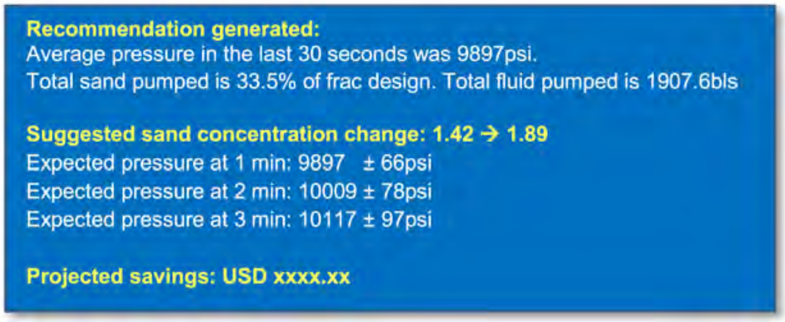

Figure 16—A sample recommendation illustrating the expected output for the user.

In this vein, optimization performed on every timestep is both impossible and irksome for an engineer. Instead, the frequency of recommendations is restricted to allow time for an engineer to implement and see the response to any optimization before a new one is generated.

Furthermore, since any optimization requires confident knowledge of pressure behavior, we also impose that for any optimization to be performed, we must be post-breakdown, in the plateau phase of the well. This is to maximize the likelihood that the pressure is stable and can be adequately modelled to make a prediction. For now, our implementation enforces that recommendations suggest either a sand OR an injection rate change and not both. This is due to operator uncertainty, and unwillingness to perform a change in multiple variables simultaneously due to concerns of screenouts and can be lifted in the future as model performance and user confidence improves. In principle, with this framework, multiple models tackling various controls or decisions could be executed simultaneously and a hierarchical optimization performed on the whole system ranked by priority, for e.g., operation safety > stimulation effectiveness > cost efficiency.

Data Engineering Model and Challenges

Wellsite Information Transfer Specification (WITS) is an industry standard communications format used to transfer a wide variety of wellsite data. The high-level model is implemented in the following process: WITS Data Stream → Model Application → End User Interface + Alert Notifications

WITS Data Stream - extracting data from 3rd party real-time data acquisition platform as frequently as possible. The 3rd party real-time data acquisition platform provides an interface for extracting WITS sensor data from multiple assets. This enables us to extract measures for a particular well between two specified start and end timestamps. It was not designed to provide data in a streaming fashion (i.e., a source which provides new data as it becomes available), so to achieve this a separate process was built to specify the parameters that should be used for each API call. Spark (Apache Spark, 2021) was leveraged via Databricks (Azure Databricks, 2021) in order to parallelize the API calls for each active treatment well. This allowed us to make multiple calls to the 3rd party real-time data acquisition platform concurrently and extract 10 seconds worth of data for all the 60+ measures captured by the platform. During typical zipper frac operations, there are multiple active wells, which means every 10 seconds several thousands of records are being called. We then wait 10 seconds before making the next set of calls. This gives the program enough time to receive and process the response before moving on to the next set of calls. It also stops us from overburdening the 3rd party real-time data acquisition platform servers with too many requests within a short timeframe.

The WITS data is enriched with contextual details such as the field name, pad name, and frac design via a connection to our internal hydraulic fracturing database. This information is combined with the data stream in real time, as new records are processed.

Model Application – running model against incoming WITS data. The model is designed to adjust parameters as more data becomes available for a stage. Each well and stage combination will have its own specific version of the model. This meant that a process was needed to use a default model at the start of any new stage and ensure that only new data for that specific well/stage was considered when adjusting the parameters.

MLflow (MLflow, 2021) is a library used to track different model experiments and control model versioning. This was used to store a different model for each well/stage and update its parameters accordingly.

The same parallelization technique used to extract WITS data was leveraged here to process the models more efficiently. A processing time of 10 seconds is used again for the model calculations to complete before the next set of data is fed in.

End User Interface (UI). The model outputs are stored in a Data Lake (Azure Data Lake Storage, 2021) for scalability, using a file format called Delta (Delta Lake, 2021). Delta can be optimized for repeated queries against specific fields, in a similar way to database optimizations.

Using Plotly Dash and Pyspark, we can query the Delta tables on demand and express operations against Pandas Data Frames for use in the visualizations. This allowed us to use the Data Scientist's existing visualization code in a more dynamic dashboard for users to interact with.

In the UI, there is a model parameters page which enables end users to adjust the default parameters used by the model. There is also a reset button to reverse all the changes made back to the default values.

Alert Notifications. A separate process is used to send notifications to Microsoft Teams in near real time as the model outputs are written to the Data Lake. This was done by using the pymsteams library available in Python and setting up a webhook in Microsoft Teams with a key that can be used by Python to access the Teams channel securely. The same notifications are also displayed in the UI, but the user can also see the alerts in Teams without having to access the UI.

Conclusions and Recommendations

We demonstrate a platform to host and execute an ensemble of models and visualizations that communicate actionable insights to an operator within seconds of identifying an event and potential to gather feedback.

Specifically, we make the following conclusions:

- Hydraulic fracturing stages should be both efficient (minimize cost) and effective (ultimate recovery). Currently this multi-objective optimization is mostly relying on the experience of the frac engineers/operators and not fully automated leading to potentially sub-optimal frac jobs.
- This paper presents an approach to real-time optimization using a combination of data science models that are constrained by physics-based rules, basin specific organizational knowledge, and/or learnings from integrated diagnostics trial of 'effective' fracturing practices to maintain stimulation distribution.
- We have demonstrated a practical, implementable solution while real-time frac models or more complicated machine learning models may not be technically or computationally feasible today.

Field testing has shown that the presented approach can consistently identify opportunities and augment decision making to save cost while preserving fracture effectiveness.

- However, the authors acknowledge the gap that fracture effectiveness has neither been demonstrated in production, nor shared any results that demonstrate that this approach has led to decline in cluster screenouts. This may be a topic of future publication once more data is gathered from field testing.
- We have demonstrated that established statistical techniques can be used to sufficiently model fracturing pressures for the purpose of optimization, having the potential to realise substantial cost benefits. We acknowledge the limitations of this optimization, both in the limited predictive power and scope of the optimization, and the imperfect assumptions made on frac behaviour.
- We present novel models and visualizations that can be readily constructed using only real-time streaming data that assist completions engineers in their decision making.
- We demonstrate that a real-time streaming pipeline can be implemented on a 3rd party source that doesn't provide streaming endpoints, and that we can use this process to apply multiple functions at scale within a magnitude of seconds.

Finally, we share learnings and recommendations from our digitalization journey:

- Get the right (capable and competent) resources and onboard them properly.
- Outsource the labor-intensive real-time data collection.
- Involve end users very early.
- True agile - adapt to user feedback and needs on the fly.

Acknowledgments

We would like to thank the managements of Shell Exploration and Production Company and Shell Global Solutions International B.V., for their support on this project as well as for giving permission to publish this paper.

References

Apache Spark. 2021. https://spark.apache.org/ (accessed 2nd December 2021)

Azure Databricks. 2021. https://docs.microsoft.com/en-us/azure/databricks/ (accessed 2nd December 2021)

Azure Data Lake Storage. 2021. https://azure.microsoft.com/en-us/services/storage/data-lake-storage/ (accessed 2nd December 2021)

Ben, Y., Perrotte, M., Ezzatabadipour, M., Ali, I., Sankaran, S., Harlin, C., and Cao, D. 2020a. Real-Time Hydraulic Fracturing Pressure Prediction with Machine Learning." Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, February 2020. doi: https://doi.org/10.2118/199699-MS

Ben, Y., Sankaran, S. Harlin, C., and Perrotte, M., 2020. Real-Time Completion Cost Optimization Using Model Predictive Control. Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, February 2020. doi: https://doi.org/10.2118/199688-MS

Business Insights Report. 2020. Hydraulic Fracturing Market Size, Share & COVID-19 Impact Analysis. Available online (Accessed 2nd December 2021): https://www.fortunebusinessinsights.com/industry-reports/hydraulic-fracturing-market-100419

Chen, P., Rahman, M.M., and Sarma, H.K., 2014. Interaction Between Hydraulic Fracture and Natural Fracture - A New Prediction Model by Adaptive Neuro-Fuzzy Inference System (ANFIS). Abu Dhabi International Petroleum Exhibition and Conference. Abu Dhabi, UAE, Society of Petroleum Engineers: 16.

Delta Lake. 2021. https://docs.delta.io/latest/index.html (accessed 2nd December 2021)

Department of Energy. 2021. Economic and National Security Impacts under a Hydraulic Fracturing Ban. Available online (Accessed 2nd December 2021): https://www.energy.gov/sites/prod/files/2021/01/f82/economic-and-national-security-impacts-under-a-hydraulic-fracturing-ban.pdf

Energy Information Administration. 2016. Trends in U.S. Oil and Natural Gas Upstream Costs. Available online (Accessed 2nd December 2021): https://www.eia.gov/analysis/studies/drilling/pdf/upstream.pdf

Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, **29**(5), 1189–1232. http://www.jstor.org/stable/2699986

Hassan, A., Mahmoud, M., Al-Majed, A., and Abdulraheem, A., 2020. A New Technique to Quantify the Productivity of Complex Wells Using Artificial Intelligence Tools. Paper presented at the International Petroleum Technology Conference, Dhahran, Kingdom of Saudi Arabia, January 2020. doi: https://doi.org/10.2523/IPTC-19706-Abstract

Morozov, A., Popkov, D., Duplyakov, V. & Mutalova, R., 2020. Machine Learning on Field Data for Hydraulic Fracturing Design Optimization: Digital Database and Production Forecast Model. Conference Proceedings, First EAGE Digitalization Conference and Exhibition, Nov 2020, Volume **2020**, p.1 - 5. European Association of Geoscientists and Engineers. doi: https://doi.org/10.3997/2214-4609.202032068

MLFlow. 2021. https://mlflow.org/docs/latest/index.html (accessed 2nd December 2021)

Paryani, M., Sia, D., Mistry, B., Fairchild, D., and Ouenes, A., 2018. Real-Time Completion Optimization of Fracture Treatment Using Commonly Available Surface Drilling and Fracing Data. Paper presented at the SPE Canada Unconventional Resources Conference, Calgary, Alberta, Canada, March 2018. doi: https://doi.org/10.2118/189810-MS

Shen, Y., Cao, D., and Ruddy, K., 2020. Deep Learning Based Hydraulic Fracture Event Recognition Enables Real-Time Automated Stage-Wise Analysis. Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, February 2020. doi: https://doi.org/10.2118/199738-MS

West, M. & Harrison, J., 2006. Bayesian Forecasting and Dynamic Models. Springer Science & Business Media.

Xiong, H. and Holditch, S.A., 1995. An Investigation into the Application of Fuzzy Logic to Well Stimulation Treatment Design. *SPE Comp App* **1**(4): 18-24; Trans., AIME, 299. SPE-27672-PA. doi: https://doi.org/10.2118/27672-PA