A Machine Learning Emulator for Hydrogen Embrittlement

Minh Triet Chau*, João Lucas de Sousa Almeida[1], Alberto Costa Nogueira[1]

[1]IBM Research Brazil

Abstract

We propose a Physics-informed Machine learning to emulate hydrogen embrittlement (HE) in pipelines by predicting the gas pressure on the pipe's inner wall. Despite its high-fidelity results, the current PDE-based simulators are time and computationally demanding. Using simulation data, we train a machine learning model as a fast and cheap emulation. It can predict the time and location of a failure much faster with controllable loss in accuracy compared to the current methods.

Blending hydrogen with natural gas is a viable strategy to reduce emissions and by-products associated with the burning of natural gas, and can displace conventional gasoline or diesel (Melaina, Antonia, and Penev 2013). Blends with less than 20% H_2 by volume can be transmitted using existing gas pipelines. However, this hydrogen compressed natural gas blend (or HCNG) is not yet mainstream. In Germany, where this technology is most widely adopted, it only accounts for 10% of demand per capita (Dolci et al. 2019). The main reason is during daily consumption, gas pressure may reach excessive values that lead to hydrogen diffusion through the pipeline wall for most natural gas pipeline materials. This phenomenon is known as hydrogen embrittlement (HE). An embrittled pipeline is prone to leaking hydrogen which can cause catastrophic events since even small amounts of hydrogen gas can ignite in the presence of air.

While the cause of HE is still in debate, it is generally accepted that the high concentration of atomic hydrogen in regions of high stress in the metallic lattice is a significant contributor (Andrews et al. 2021). Current pipeline integrity assessment methods employed in the Oil and Gas industry use tensile and notched tensile, slow strain rate, or disk pressure tests (European Industrial Gases Association 2004). Those tests are slow and costly, especially for complex pipeline networks. Since H_2 diffusion strongly correlates with pressure (Hafsi, Mishra, and Elaoud 2018), we hypothesize that by emulating the pressure of H_2 in the pipeline, we can predict where the leakage time and location occurs quickly and have more time to mitigate hazardous effects. Potentially, we can use the method on existing pipelines to anticipate possible hazardous events. Source code is available at `https://github.com/minhtriet/hydrogen_emb`.

*Corresponding author `s6michau@uni-bonn.de`

Problem statement

Given the parameters in Table 1, the Reynolds number of the flow in a pipeline is estimated as

$$Re = \frac{\rho u D}{\mu} = \frac{0.08988 \times 20 \times 7.62e^{-2}}{0.88e^{5}} \approx 15565.58$$

In a pipeline, when the Reynolds number is larger than 2900, the flow is turbulent (Schlichting and Gersten 2016) (See Figure 1). Such turbulence makes the system inherently chaotic, which is tricky from the perspective of conventional machine learning models. Usually, gas flow dynamics can be simulated using the PIMPLE algorithm, which solves an approximation of the Navier-Stokes equation. Classical simulators are time and computationally consuming. For instance, a simulation of 1000 time steps with fixed $\Delta t = 0.002s$ takes 30225 seconds. To make the matter worse, training a deep neural network with such a small amount of data is quite hard as it needs a lot of data to make the model generalize well. In this work, we keep the deep learning approach as simple as possible to reduce overfitting while emulating the physical system.

We will organize the paper as follows. Firstly, we present the specifications for our simulation. Then we discuss the choice of machine learning models to emulate our dynamic system. The details of the experiments follow, and we finish with discussions and conclusions.

Simulation setup

The setup for the simulation is in Table 1. The specs of the pipeline are from oil and natural gas pipeline standards (Mohitpour, Golshan, and Murray 2003).

The previous session establishes the fact that HE main culprit is H_2 molar concentration C. Two factors that govern C are the velocity v and pressure p of H_2, which affect the gas flow and the non-steady H_2 diffusion in steel (Hafsi, Mishra, and Elaoud 2018). With a simulation, we can construct snapshots for p and v through time. We can stack that information in the so-called snapshot matrix (cf. Figure 3b). The details of our simulation are in Figure 3. We use SimScale to simulate the physical system and collect the datasets to train our machine learning models.

After getting the simulator output, we use the first 50% data as the training set, and the next 10% as the validation

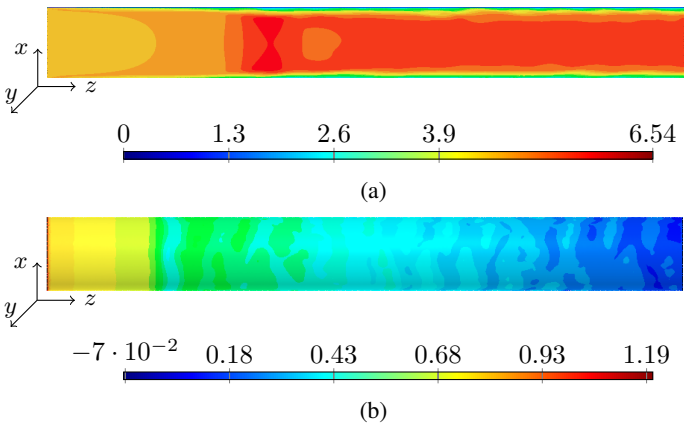


Figure 1: (1a): The velocity field profile of gas flow (m/s) in a pipeline. (1b): The pressure field (Pa) of gas flow in a pipeline to the inner wall. The reason of negative value is in Table 1.

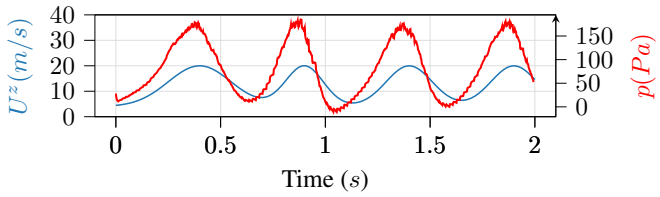


Figure 2: Time varying velocity U^z and mean pressure at the pipeline inlet.

set, which is used to find the best hyper-parameters for the machine learning models. After that, we retrain the model with the train and validation sets to predict the test set.

Machine learning model

The details of simulation and data transformation steps are in Figure 3.

To reduce the training time and computational cost, we propose building a surrogate model of the simulator using a Reduced Order Modeling technique (PCA or Dynamic Mode Decomposition) to generate a low-rank representation of the dynamic system preserving most of its original energy content. The left singular vectors V are the spatial eigenmodes, ordered by increasing energy content. We want to compute $\tilde{V}$ such that the MSE of $\|U - \tilde{V}\tilde{V}^T U\|_F^2$ is minimum, with U the original data). When projecting the data again onto the left singular vectors, we get the temporal coefficients or a set time series, on which machine learning can be employed.

In this work, we are going to compare Transformers and Temporal Fusion Transformers (Lim et al. 2021) with Operator Inference (OpInf) (Ghattas and Willcox 2021), a physics-aware method while fitting time series in the reduced order space.

Parameter	Value
Pipeline	
Diameter (cm)	7.62
Length (cm)	500
Orientation (cm)	Length is measured in z axis
Boundary condition (BC)	
Inlet velocity U^z_{inlet}	The gas velocity in a pipeline. To imitate the use of gas in real life, we create a periodic profile (Figure 2).
P_{outlet} (pa)	0. The setup is in relative pressure measure, 0 here means 1 atm or 101325 pa.
Wall	No slip
Simulation control	
Turbulence model	SST $k-\omega$ (Menter 1993)
Solver algorithm	PIMPLE
Δt	$0.002s$

Table 1: Simulation setup

Numerical experiments

For all methods, the prediction horizon was the test datasets length. No information about $\{p, U, nut, \omega, k\}_{\text{test}}$ is available for the models at the prediction stage.

Transformer and Temporal Fusion Transformers (TFT)

For the Transformers approach, we perform a grid search for the number of encoder and decoder layers, dropout rate, and the number of attention heads. Except for p, the other features are set as covariance series. The training of those models is powered by (Herzen et al. 2022). Details are in Table 2. Even though TFT performs better than Transformers and OpInf in most cases (See also table 3), we argue why it may not be a desirable model in the next session.

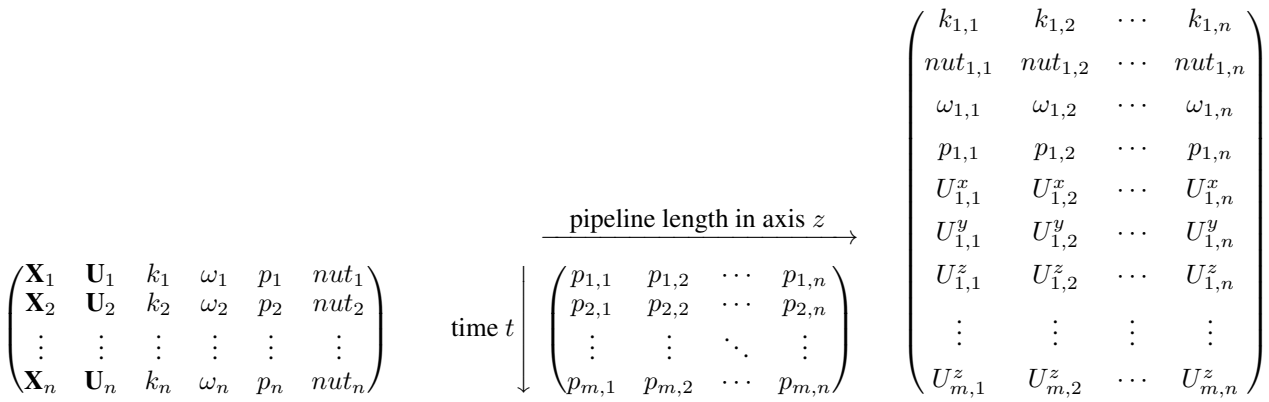
		RMSE	
Run	Input	Transformer	TFT
1	p	61.2973	**59.1081**
2	p, U^z	60.5701	58.3013
3	p, U^{xyz}	61.8641	**59.1939**
4	p, U^{xyz}, k	61.5241	**58.9044**
5	p, U^{xyz}, k, ω	61.5505	**58.8527**
6	$p, U^{xyz}, k, \omega, nut$	61.8284	**59.2410**

Table 2: Summary for Transformer and TFT models

Operator Inference (OpInf)

In its first step, OpInf reduces a full order model to a polynomial representation in latent space as

$$\frac{d}{dt}x_t = c + A(x_t) + H(x_t \otimes x_t) + G(x_t \otimes x_t \otimes x_t) + B(u_t)$$



(a) (b) (c)

Figure 3: The organization of features. Figure 3a: Raw simulation output field variables at grid point position $\mathbf{X} \in \mathbb{R}^3$ - gas velocity $\mathbf{U} \in \mathbb{R}^3$, pipeline internal static pressure p, rate of dissipation of kinematic turbulence k, kinematic turbulence energy ω, kinematic turbulent viscosity nut. Figure 3b: Snapshot matrix with the number of pipeline grid points n, and the total number of simulation time steps m. Figure 3c: For the Operator Inference method, we stack the field variables of multiple snapshots as input.

$\otimes$ operator denotes a column-wise Kronecker product. A, H and G represent linear, quadratic, and cubic state matrices respectively. $B(u_t)$ can be used as the reduced operator of a forcing term in reduced space. c is a constant term. Each of the terms may or may not be used by OpInf, depending on the experimenter's choice. In this work, we use A, H, B operators and c. In the second step, OpInf solves a least square problem to infer these operators.

Two key parameters to optimize for OpInf are the number of the basis for its dimension reduction, and the regularizer for the least square solver.

While choosing the number of basis in reduced space, we measure the quality of the reduced order model by computing the preserved energy with respect to the original dynamical system and calculating the RMSE between the original state variables and a PCA reconstruction of those variables. Based on these criteria, we chose 30 eigenmodes. Depending on what inputs $(p, \mathbf{U}, k, nut, \omega)$ to feed the algorithm, 30 basis capture at least 99.82% of the cumulative energy of the system. We also perform a PCA on train and test with the same number of bases to check if there is a distribution shift between these subsets.

Run	Input	Regularizer	RMSE
1	p	910.204	66.5808
2	p, U_z	918.367	**34.7737**
3	p, U^{xyz}	787.755	78.8812
4	p, U^{xyz}, k	983.673	104.3637
5	p, U^{xyz}, k, ω	885.714	103.5558
6	$p, U^{xyz}, k, \omega, nut$	2041734.694	85.7531

Table 3: Summary for Operator Inference approach. The highlighted result is when compared with TFT and Transformers models from Table 2. See Figure 4 for a visual comparison.

A crucial step for the OpInf approach is choosing the regularization penalties, since RMSE minimization alone may lead to overfitting. For the case OpInf run 6 (Figure 4f), even though its RMSE is smaller than run 4 (Figure 4d) and 5 (Figure 4e), it fails to capture the system dynamics induced by the BC (Compare with Figure 4f). The reason it does better metrically than runs 4 and 5 is that the latter runs miss the peak and valley of p, albeit they provide a generally similar wave-like prediction.

OpInf run 2 (Figure 4c) gives the best result as it uses the two most important features U^z and p. p is what we are trying to predict, and the flow is strictly dominant in the z axis (See Figure 1). For a vanilla OpInf, the rest of the features (ω, k, nut) are underutilized. To include them we need to include a closure mechanism of the simulator's algorithm in the latent space of OpInf, which is beyond the scope of this work.

Related work

Hydrogen embrittlement

The body of this research topic is for and by material scientists. The topics are new methods to obtain a quantitative measure of the embrittlement (Laliberté-Riverin et al. 2020). Other works focus on a database of experiments with different alloys for H_2 storage, not delivery (Thankachan et al. 2017, HyMARC and h2tools.org). (Hafsi, Mishra, and Elaoud 2018) uses simulations to predict the concentration (ppm) of H_2 in steel after long exposure.

To the best of our knowledge, there has been no work using the time-based approach for embrittlement. However, the effect of H_2 on pipelines can be derived through time-dependent PDEs.

Machine learning for dynamical systems

There is a spectrum from pure deep learning to pure physics method. On the deep learning's end (Sanchez-Gonzalez et al.

2020) benchmarks on RMSE to a ground truth simulation, without regard if any physics rules persist in the model.

On the other side of the spectrum is the pure numerical methods as seen in simulators. With their pros and cons discussed in previous sessions, we turn our interest to the works that lie somewhere in the middle. They enforce some physics constraints to existing deep learning architecture. (Wang, Walters, and Yu 2021) impose the relation of symmetric groups and conservation rule, which is given by Noether's principle, to convolution neural network.

Based on a result from the universal approximation theorem, which says one hidden layer neural network can approximate any nonlinear continuous operators, (Lu, Jin, and Karniadakis 2019) build a neural network architect to solve PDEs. Another ML approach for dynamical systems is (Li et al. 2020), where they propose a new type of neural operator by parameterizing its integral kernel to Fourier space.

Conclusion

We propose a machine learning model for emulating HE through H_2 concentration and pressure. Compared to simulations, it now takes only seconds to predict the pressure in a pipeline. This work proposes a fast and fairly accurate predictor of the location and time of the next pipeline failure.

The effect of our method is twofold. For the current pipeline, it predicts the next failure point and time faster than physical simulation, consequently giving more time for taking action.

For new material discovery, it is forbiddingly expensive to facilitate a high throughput experiment cycle, as they are only usable after the sample is sufficiently exposed to hydrogen. The ASTM F1624 test, for example, takes up to 30 hours. Instead, we hypothesize that our method could measure the pressure of H_2 during that period.

This method is a cost-effective pipeline material testing tool for HCNG or potentially a pure hydrogen transmission. It helps accelerate the hydrogen economy to mitigate climate change.

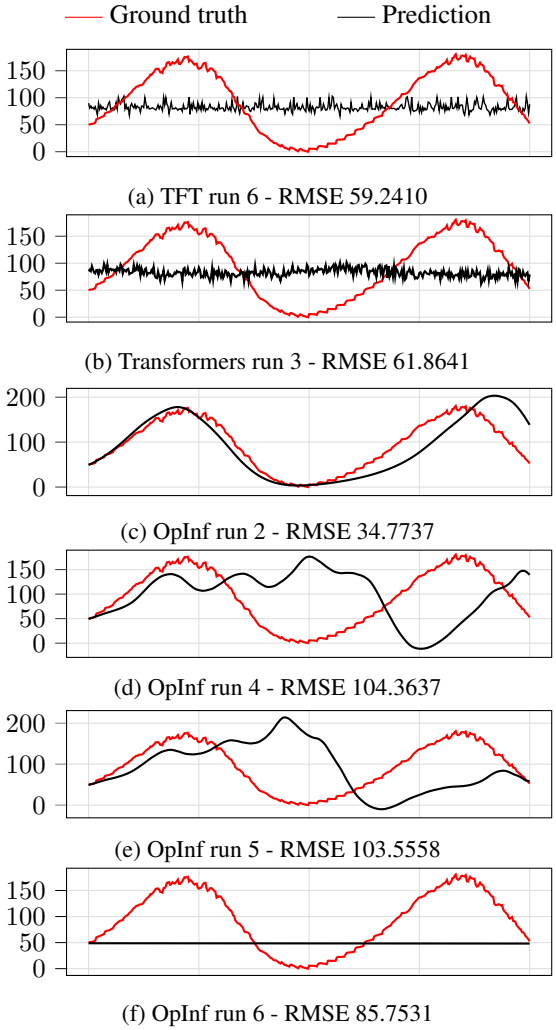


Figure 4: Comparison of prediction of average pressure through time. x axis is time, y axis is average pressure (Pa). Figure 4a and 4b shows a typical prediction of Transformers and TFT, regardless of future input and parameters tuning. For OpInf, a higher regularizer used in Figure 4f causes the insensitivity to changes from BC, even though it minimizes validation loss.

References

Andrews, R. M.; Gallon, N.; Huising, O. J.; and Lam-Thanh, L. 2021. Hydrogen Transport Pipelines - Design Challenges and Mitigations. *Pipeline Technology Conference 2021 Proceeding*.

Dolci, F.; Thomas, D.; Hilliard, S.; Guerra, C. F.; Hancke, R.; Ito, H.; Jegoux, M.; Kreeft, G.; Leaver, J.; Newborough, M.; Proost, J.; Robinius, M.; Weidner, E.; Mansilla, C.; and Lucchese, P. 2019. Incentives and legal barriers for power-to-hydrogen pathways: An international snapshot. *International Journal of Hydrogen Energy*, 44(23): 11394–11401.

European Industrial Gases Association. 2004. HYDROGEN TRANSPORTATION PIPELINES.

Ghattas, O.; and Willcox, K. 2021. Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30: 445–554.

Hafsi, Z.; Mishra, M.; and Elaoud, S. 2018. Hydrogen embrittlement of steel pipelines during transients. *Procedia Structural Integrity*, 13: 210–217. ECF22 - Loading and Environmental effects on Structural Integrity.

Herzen, J.; Lässig, F.; Piazzetta, S. G.; Neuer, T.; Tafti, L.; Raille, G.; Pottelbergh, T. V.; Pasieka, M.; Skrodzki, A.; Huguenin, N.; Dumonal, M.; Kościsz, J.; Bader, D.; Gusset, F.; Benheddi, M.; Williamson, C.; Kosinski, M.; Petrik, M.; and Grosch, G. 2022. Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research*, 23(124): 1–6.

Laliberté-Riverin, S.; Bellemare, J.; Sirois, F.; and Brochu, M. 2020. Determination of hydrogen embrittlement stress intensity threshold by fractography.

Li, Z.; Kovachki, N. B.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A. M.; and Anandkumar, A. 2020.

Fourier Neural Operator for Parametric Partial Differential Equations. *CoRR*, abs/2010.08895.

Lim, B.; Loeff, N.; Arik, S.; and Pfister, T. 2021. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.

Lu, L.; Jin, P.; and Karniadakis, G. E. 2019. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *CoRR*, abs/1910.03193.

Melaina, M. W.; Antonia, O.; and Penev, M. 2013. Blending Hydrogen into Natural Gas Pipeline Networks. A Review of Key Issues.

Menter, F. R. 1993. ZONAL TWO EQUATION k-w TURBULENCE MODELS FOR AERODYNAMIC FLOWS.

Mohitpour, M.; Golshan, H.; and Murray, A. 2003. *Pipeline Design and Construction*. Fairfield, CA: American Society of Mechanical Engineers.

Sanchez-Gonzalez, A.; Godwin, J.; Pfaff, T.; Ying, R.; Leskovec, J.; and Battaglia, P. W. 2020. Learning to Simulate Complex Physics with Graph Networks. *CoRR*, abs/2002.09405.

Schlichting, H.; and Gersten, K. 2016. *Boundary-layer theory*. Berlin, Germany: Springer, 9 edition.

Thankachan, T.; Prakash, K. S.; David Pleass, C.; Rammasamy, D.; Prabakaran, B.; and Jothi, S. 2017. Artificial neural network to predict the degraded mechanical properties of metallic materials due to the presence of hydrogen. *International Journal of Hydrogen Energy*, 42(47): 28612–28621.

Wang, R.; Walters, R.; and Yu, R. 2021. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. In *International Conference on Learning Representations*.