PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES Paris

Data and Physics-Based Approaches for Stick-Slip Estimation

Approches Basées sur les Données et la Physique pour l'Estimation du Stick-Slip

Soutenue par

Hana YAHIA

Le 09 décembre 2025

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Energétique - ISMME

Spécialité

Mathématique et Automatique

Composition du jury :

Claire VALENTIN Professeur des Universités Université Claude Bernard Lyon 1	*Présidente du jury*
Roman SHOR Associate Professor Texas A&M University	*Rapporteur*
Rémi FLAMARY Professeur École Polytechnique	*Rapporteur*
Rémi BARDENET Directeur de recherche CNRS	*Examinateur*
Mohamed MAHJOUB Ingénieur de recherche Helmerich & Payne	*Examinateur*
Bruno FIGLIUZZI Maître-Assistant Mines Paris, Université PSL	*Examinateur*
Laurent GERBAUD Ingénieur de recherche Mines Paris, Université PSL	*Examinateur*
Florent DI MEGLIO Professeur Mines Paris, Université PSL	*Directeur de thèse*

MINES PARIS | PSL

Mines Paris

DOCTORAL THESIS

Data and Physics-Based Approaches for Stick-Slip Estimation

Author: Hana YAHIA

Supervisors: Florent DI MEGLIO
Bruno FIGLIUZZI
Laurent GERBAUD
Stephane MENAND
Mohamed MAHJOUB

Centre Automatique et Systèmes – Mines Paris – PSL
Centre de Morphologie Mathématique – Mines Paris – PSL
Géologie de l'Ingénieur et Géomécanique – Mines Paris – PSL
Helmerich & Payne





Je ne suis pas folle, je suis juste moi . . .

Résumé

Cette thèse étudie des méthodes permettant d'estimer la sévérité des vibrations de type "stick-slip" au fond du puits lors du forage, en utilisant comme entrée des mesures de surface échantillonnées à basse fréquence. Deux approches sont proposées : la première vise à estimer le "Stick-Slip Index (SSI)", un indicateur couramment utilisé dans l'industrie pour évaluer la sévérité des vibrations torsionnelles, directement à partir des mesures de surface. La seconde consiste à reconstruire la variation de la vitesse de rotation du trépan à partir de ces mêmes mesures. Une fois cette variation obtenue, elle permet d'inférer la sévérité des vibrations torsionnelles au fond du puits. En conséquence, la thèse est structurée en deux grandes parties, chacune étant consacrée à l'une de ces approches.

Dans la première partie, un modèle d'apprentissage automatique est utilisé pour estimer le SSI à partir de séquences temporelles de mesures de surface. L'objectif est d'entraîner un modèle capable de bien généraliser sur différents puits présentant des conditions géologiques variées. À cette fin, plusieurs stratégies sont explorées, notamment la normalisation des données, l'intégration de caractéristiques dérivées de lois physiques, l'application de l'apprentissage par transfert ou "transfer learning (TL)", ainsi que l'augmentation de données afin d'obtenir un jeu de données plus équilibré. L'influence de l'architecture du modèle est étudiée en comparant les réseaux récurrents à mémoire court et long terme ou "Long Short Term Memory (LSTM)" et les modèles de type "Transformer". En outre, des stratégies avancées de généralisation, telles que la généralisation de domaine adversariale ou "Adversarial Domain Generalization (ADG)" et la minimisation du risque invariant ou "Invariant Risk Minimization (IRM)", sont employées afin d'améliorer la robustesse face aux changements de domaine et à la variabilité opérationnelle.

Dans la seconde partie, deux méthodes hybrides sont étudiées pour prédire la vitesse de rotation du trépan et les coefficients de frottement entre la tige de forage et le puits, à partir de mesures de surface échantillonnées à 1 Hz et d'un modèle dynamique torsionnel non linéaire simplifié. La première méthode repose sur un réseau de neurones informé par la physique ou "Physics-Informed Neural Network (PINN)", une approche hors ligne utilisée pour résoudre le problème d'identification du système, le modèle physique non linéaire étant intégré dans la fonction de perte. La seconde méthode est une approche en ligne, issue de la théorie du contrôle, basée sur une version améliorée du filtre de "Salted Kalman Filter (SKF)" adaptée aux systèmes hybrides non linéaires. Dans les deux cas, les méthodes sont évaluées sur des données simulées et réelles afin d'en évaluer l'applicabilité pratique et la robustesse.

Abstract

This thesis investigates methods for estimating the severity of downhole stick-slip vibrations during drilling, using surface measurements with a low-frequency sample rate as input. Two approaches are proposed: The first focuses on estimating the Stick-Slip Index (SSI), an industry-standard indicator for assessing the severity of torsional vibrations, directly from surface measurements. The second seeks to reconstruct the variation of the bit rotation speed from surface measurements. Once this variation is obtained, it provides information about the severity of downhole torsional vibrations. Accordingly, the thesis is structured into two main parts, each dedicated to one of these approaches.

In the first part, a data-based machine learning model is used to estimate the SSI from time-series sequences of surface measurements. The objective is to train a model that generalizes well across different wells with varying geological conditions. To this end, several strategies are explored, including data normalization, the incorporation of engineered features derived from physical laws, the application of transfer learning, and the augmentation of data to achieve a more balanced dataset. The influence of model architecture is examined through comparisons between Long Short-Term Memory (LSTM) networks and Transformer-based models. Furthermore, advanced generalization strategies, such as Adversarial Domain Generalization (ADG) and Invariant Risk Minimization (IRM), are employed to enhance robustness to domain shifts and operational variability.

In the second part, two hybrid-based methods are investigated to predict the bit rotation speed and the friction coefficients between the drill string and the borehole using 1-Hz surface measurements and a simplified nonlinear torsional dynamic model. The first method relies on a Physics-Informed Neural Network (PINN), an offline approach used to solve the system identification problem with the nonlinear physical model incorporated into the loss function. The second method is an online, control-theoretic approach based on an enhanced version of the Salted Kalman Filter (SKF) adapted to hybrid nonlinear systems. In both cases, the methods are evaluated on simulated and field data to assess their practical applicability and robustness.

Remerciements

I would first like to thank the members of the jury who kindly agreed to evaluate this work, for their presence, the time devoted to reading my thesis, as well as for the comments they will share with me during my defense. I am also deeply grateful for their efforts in traveling to Paris to attend my defense and for sharing their expertise and knowledge with me.

Cette thèse est ma première expérience de travail avec des gens pendant une aussi longue période, c'est pour cela que je n'oublierai aucun de ceux avec qui j'ai commencé ma carrière. À la tête de cette liste se trouve Florent, mon directeur de thèse et mon ami, c'est l'encadrant qui m'a le plus soutenu, qui me connaît le mieux, qui a supporté ma tête dure, qui me critique le plus, qui a supporté mon excitation incontrôlable et mes réactions parfois un peu folles lorsque nous obtenions de bons résultats, ainsi que mon impatience à explorer d'autres pistes. Je garde en mémoire nos discussions au labo avec nos deux voix hautes, où les gens pensaient que nous étions en train de nous disputer, alors qu'en réalité c'était tout simplement parce que nous étions passionnés par ce que nous faisions. Il a toujours été là aux moments de succès et surtout d'échec, il a consacré de son temps pour me calmer quand il le fallait, pour m'encourager, pour me soutenir, pour m'écouter, tout simplement pour que je ne sois pas seul. Merci de m'avoir appris tant de choses, merci pour ta confiance qui m'a permis de tester plein d'idées et d'apprendre énormément. C'est la raison pour laquelle je me demande ce que je fais dans la vie: je sens que je peux tout faire, tant que l'on aime ce que l'on fait. Merci de m'avoir permis d'adorer le métier de chercheur et de découvrir le plaisir et mon amour derrière ce métier. Merci pour ton attention aux petits détails, merci de m'avoir soutenu, merci d'avoir partagé avec moi cette expérience magnifique de ma vie, merci d'être un ami, merci pour tout.

Ensuite, Bruno, l'encadrant le plus gentil du monde (et celui qui a sauvé ma thèse ;-)), je suis tellement heureuse d'avoir eu la chance de travailler avec une personne aussi professionnelle que toi. Merci de m'avoir tant appris, de m'avoir fait confiance, encouragée et soutenue chaque fois que j'en avais besoin, même sans que j'aie à le demander. Merci d'être tout le temps là, même si tu n'es pas disponible physiquement tu es toujours là pour me soutenir par mail. Nos points ensemble vont beaucoup me manquer. Avec Florent, travailler avec vous est un véritable plaisir, que j'espère pouvoir poursuivre encore beaucoup beaucoup plus longtemps.

Laurent, je n'ai pas eu la chance de te voir autant que Florent et Bruno, mais chacune de nos rencontres m'a fait adorer discuter avec toi. Ton sourire constant dégage une énergie positive incroyable, tu es vraiment exceptionnel. Merci d'avoir participé à ma thèse, pour ton aide, ton écoute, ta bienveillance et ton sourire si communicatif. J'espère aussi avoir l'occasion de continuer à travailler avec toi à l'avenir.

Ensuite, Stéphane et Mohamed, merci pour votre bienveillance, pour m'avoir si bien accueillie au sein de DrillScan, merci pour le temps que vous avez consacré à ma thèse, et pour tous les efforts que vous avez fournis en sa faveur. Merci à toute l'équipe de DrillScan pour m'avoir si adorablement accueillie, merci à l'équipe science qui m'ont consacré de leurs temps pour m'aider et m'expliquer ce que je ne comprenais pas, merci à tous les développeurs qui m'ont aidée à lancer mes calculs, et particulièrement Jérémy, qui est juste incroyable. Il a supporté mes demandes de secours à la dernière minute de travail et, malgré tout, il a toujours été bienveillant, et n'a jamais hésité à tout mettre en œuvre pour résoudre mes problèmes.

Merci à toute l'équipe du CAS, avec qui je passais le plus de temps. Vous étiez comme ma famille. Merci à tous les permanents pour leur bienveillance, leur gentillesse et le partage de leurs connaissances. Merci à mes collègues doctorants que j'ai pu rencontrer dès le début de ma

thèse, et à mes collègues actuelles avec qui je partage de super moments pour discuter, rigoler, apprendre et s'amuser. Un remerciement spécial à François PACAUD, qui m'a aidée pour tous les problèmes de serveur GPU, mémoire, versions... Merci beaucoup pour ta patience face à mes dépassements d'utilisation du serveur et pour mes notebooks qui occupaient la majorité de sa mémoire. Désolée pour toutes les fois où je t'ai empêché de lancer tes calculs parce que j'avais utilisé le serveur en totalité, et pour le nombre de fois où j'ai "craqué" le serveur. Merci beaucoup de m'avoir supportée, et félicitations : je te libère enfin le serveur !

Je souhaite bien sûr aussi remercier ma famille, principalement mes parents, mes amours Bechir et Aicha. Merci beaucoup pour votre encouragement et soutien. Merci de m'avoir encouragée à faire ce que je veux et à laisser libre cours à ma folie comme je le souhaite. Merci à ma mère qui m'appelle tous les jours juste pour savoir si j'ai bien avancé dans ma thèse, sans toujours comprendre ce que je fais, merci de m'avoir supportée jusqu'à maintenant, de me préparer à manger, et d'être toujours à mes côtés moralement. Merci à mon père qui ne veut pas que j'arrête mes études, même avec un doctorat ! Mais désolée papa, il faut s'arrêter à un certain moment. Tout cela, c'est grâce à vous et c'est pour vous ! Je vous adore.

Merci à mes sœurs Fatma et Rim et mon frère Mohamed, et bien sûr mes neveux, mon beau-gosse Islem et mon petit amour Adem, à qui je dédie cette thèse, qu'ils seront obligés de lire quand ils seront grands, parce que c'est la thèse de leur magnifique tante.

Ma famille, vous êtes chanceux de m'avoir dans votre vie ! Je vous apporte de l'ambiance, je vous rends fiers, et je vous remplis de joie chaque jour, je ne sais pas vous faites quoi sans moi. Merci pour votre amour, votre soutien et pour tout ce que vous faites pour moi.

Et je finis par la personne la plus importante : moi, Hana, ou mon pseudonyme préféré, Hanou. Merci à moi-même pour tout l'effort que j'ai fourni, pour l'amour que je porte à mon métier et à ma thèse, et pour tout le temps que je me suis consacré. Merci à ma folie, mon enthousiasme et mon excitation. Merci à mes neurones qui supportent que je les fasse fonctionner 24/24 sans pause. Merci aux idées qui surgissent juste avant le sommeil pour résoudre un bug ou un problème, et qui m'empêchent de dormir tellement je suis excitée de les tester le matin et de les partager avec mes encadrants. Merci à mon corps et à ma santé qui continuent de me soutenir, même si je ne prends pas toujours soin d'eux. Merci à mes boutons qui ne manquent jamais une occasion de m'accompagner sur le visage et de montrer mon stress. Merci à mes discussions quotidiennes avec moi-même, qui m'aident à supporter la solitude. Merci à ma maturité (pas sûre :-)), à ma force et surtout à ma faiblesse, un équilibre nécessaire pour garder les pieds sur terre. Merci à tous les échecs que j'ai rencontrés, et aux reviewers qui ont refusé tous mes articles. Ils me font encore un peu mal, mais ils m'ont construite. Merci à mes rêves, plus je les laisse s'envoler, plus ils montent haut. Merci à mon nounourse, qui m'accompagne et partage ma vie avec moi. Merci Hanou, pour tout. Je sais que ce n'était pas facile et que c'était fatigant, mais tu l'as fait. Je suis tellement fière de toi. C'est vrai que tu m'as épuisée avec ta tête dure, mais je suis contente pour toi. J'espère toujours te voir heureuse, j'espère que tout ira mieux, inchallah. Continue à nourrir ta folie, ne la lâche pas, et fais toujours ce qui te fait plaisir.

Contents

Résumé i

Abstract iii

Remerciements vi

List of figures xii

List of tables xxi

Acronyms xxiv

1 Introduction - version française 1
- 1.1 Contexte général . . . 1
- 1.2 Contexte du forage . . . 4
- 1.3 Plan de la thèse . . . 5
- 1.4 Publications . . . 6

2 Introduction 9
- 2.1 General context . . . 9
- 2.2 Drilling context . . . 11
- 2.3 Outline of the thesis . . . 13
- 2.4 Publications . . . 14

I Generalizable time series ML models for Stick-Slip Index prediction 15

3 Quantifying stick-slip oscillations 17
- 3.1 Stick-slip oscillations . . . 17
- 3.2 Stick-slip severity . . . 19
  - 3.2.1 Stick-Slip Index . . . 19
  - 3.2.2 Torque Variation Index . . . 20
  - 3.2.3 Power spectrum of the surface torque . . . 21
  - 3.2.4 Comparison of proposed stick-slip severity quantification methods . . . 21
- 3.3 Review of machine learning studies on stick-slip detection . . . 26
- 3.4 Presentation of the dataset . . . 27
  - 3.4.1 Data cleaning . . . 27
  - 3.4.2 Data analysis and visualization . . . 29

3.4.3 Well-to-well statistical comparison . . . 30
3.4.4 Class imbalance . . . 33
3.5 Contribution of Part I . . . 34

4 Stick-Slip Index regression and generalizability 37
4.1 Baseline model for SSI prediction . . . 38
4.1.1 Long Short-Term Memory . . . 38
4.1.2 Baseline model architecture . . . 38
4.2 Data normalization . . . 39
4.2.1 Feature-by-feature normalization . . . 39
4.2.2 Sequence normalization . . . 39
4.2.3 Deep Adaptive Input Normalization . . . 39
4.2.4 Layer Normalization . . . 40
4.3 Transfer learning . . . 40
4.3.1 Definition . . . 40
4.3.2 Implementation . . . 41
4.4 Data augmentation . . . 41
4.5 Physics integration . . . 42
4.6 Results and discussions . . . 43
4.6.1 Comparison of data normalization methods . . . 43
4.6.2 Transfer learning contribution . . . 44
4.6.3 Impact of data augmentation . . . 48
4.6.4 Addition of physical features . . . 49
4.7 Comparison of predicted SSI with TVI . . . 51
4.8 LSTM vs Transformer . . . 52
4.8.1 Transformer . . . 53
4.8.2 Results and discussions . . . 55
4.9 Permutation Feature Importance . . . 57
4.10 Real-time application . . . 59
4.11 Conclusion . . . 60

5 Domain Generalization 63
5.1 Review . . . 64
5.2 Methodology . . . 64
5.2.1 Adversarial Domain Generalization . . . 65
5.2.2 Invariant Risk Minimization . . . 67
5.3 Models characteristics . . . 68
5.3.1 Baseline model . . . 68
5.3.2 Adversarial Domain Generalization model . . . 68
5.3.3 Invariant Risk Minimization model . . . 70
5.4 Experiments . . . 71
5.4.1 Data processing . . . 71
5.4.2 Grid search for hyperparameters tuning . . . 71
5.4.3 Regularization coefficient . . . 72
5.4.4 Number of generator hidden layers and weighting coefficient . . . 74
5.5 Results and discussions . . . 74
5.5.1 Comparison of ADG, IRM, and baseline models . . . 74

5.5.2 Transfer learning application . . . 76
5.6 Causes of misprediction of SSI . . . 83
5.6.1 Synchronization problem . . . 83
5.6.2 Undetected stick-slip . . . 83
5.6.3 Mislabeled data . . . 84
5.6.4 Domain mismatch . . . 85
5.7 Conclusion . . . 85

II Downhole velocity and friction coefficients prediction from low-frequency surface data 87

6 Bit rotation speed and friction coefficients estimation - Introduction 89
6.1 Problem statement . . . 90
6.2 Review . . . 90
6.3 Drill string torsional dynamical model and friction parameters . . . 92
6.3.1 Lumped torsional drilling model . . . 92
6.3.2 Friction model . . . 93
6.3.2.1 Smooth friction model . . . 94
6.3.2.2 Stiction friction model . . . 95
6.3.3 Torque-On-Bit model . . . 96
6.4 Simplified torsional dynamical model . . . 97
6.5 The objective . . . 99

7 Physics-Informed Neural Network 101
7.1 Review . . . 102
7.2 Objective . . . 103
7.3 Physics-Informed Neural Network . . . 104
7.4 PINN for estimation in drilling . . . 105
7.4.1 PINN's physical model . . . 105
7.4.2 PINN's loss function . . . 106
7.4.3 Choice of loss weighting coefficients . . . 108
7.4.4 PINN's architecture . . . 109
7.5 Transfer learning for PINN . . . 109
7.6 Numerical results . . . 110
7.6.1 Application on simulated data . . . 111
7.6.2 Application of transfer learning on simulated data . . . 113
7.7 Application on field data . . . 115
7.8 Conclusion . . . 118

8 Projected Salted Kalman Filter 119
8.1 Hybrid dynamical systems feature . . . 120
8.2 Review . . . 121
8.3 Objective . . . 122
8.4 Salted Kalman Filter . . . 122
8.4.1 Definition . . . 122
8.4.2 Challenges associated with stiction . . . 124

8.5 The Projected Salted Kalman Filter . . . 128
8.5.1 Linear constraints . . . 129
8.5.1.1 Linear equality constraints . . . 129
8.5.1.2 Linear inequality constraints . . . 129
8.5.2 Nonlinear constraints . . . 130
8.5.2.1 Nonlinear equality constraints . . . 130
8.5.2.2 Nonlinear inequality constraints . . . 130
8.5.3 PSKF Algorithm . . . 130
8.6 Adaptive estimation of noise covariance matrices . . . 131
8.7 Comparison of SKF and PSKF . . . 133
8.7.1 Case 1: One-mass lumped torsional dynamic model . . . 133
8.7.2 Case 2: Two-mass lumped torsional dynamic model . . . 133
8.7.3 Results and discussions . . . 135
8.8 PSKF validation on complex torsional physical models: Multi-mass and PDE Cases . . . 138
8.8.1 Case 1: PSKF with identical multi-mass model for generation and estimation . . . 139
8.8.1.1 Multi-mass lumped torsional models with hybrid stick-slip dynamics . . . 139
8.8.1.2 Ablation studies . . . 140
8.8.2 Case 2: Performance of PSKF with lumped models on PDE-generated sequences . . . 149
8.8.2.1 Evaluation of PSKF with one-mass model on PDE-based simulations . . . 151
8.8.2.2 Evaluation of PSKF with two-mass model on PDE-based simulations . . . 154
8.9 PSKF application on field data . . . 154
8.10 Conclusion . . . 155

9 PINN and PSKF comparison **157**
9.1 Objective . . . 157
9.2 Comparison on simulated data . . . 158
9.2.1 Parameters identification with `nlgreyest` . . . 159
9.2.2 Comparison results . . . 159
9.3 Comparison on field data . . . 162
9.3.1 Field data description . . . 162
9.3.2 Results and discussions . . . 163
9.3.2.1 Application on a selection of short sequences of field data . . . 163
9.3.2.2 Application on long sequences of field data . . . 168
9.3.2.3 Prediction of severe downhole stick-slip for sequences with extremely low TVI . . . 174
9.3.2.4 Conclusion . . . 175
9.4 Conclusion . . . 177

10 Conclusion - version française **179**

11 Summary and perspectives **183**

Bibliography **187**

SI metric conversion factors **209**

List of figures

1.1 Représentation schématique simplifiée du système de forage : BHA désigne la partie inférieure de la tige de forage; ω_t, ω_{Motor} et ω_b représentent respectivement les vitesses de rotation en surface, du moteur de fond et de l'outil; couple à l'outil ou "Torque-On-Bit (TOB) et WOB correspondent respectivement au couple et au poids sur l'outil. 5

2.1 Simplified schematic representation of the drilling system: Bottom-Hole-Assembly (BHA) denotes the lower part of the drill string; ω_t, ω_{Motor}, and ω_b represent the rotational speeds at the surface, the downhole motor, and the bit, respectively; Torque-On-Bit (TOB) and Weight-On-Bit (WOB) correspond to the torque and the weight-on-bit, respectively. 12

3.1 Example of a severe stick-slip sequence triggered by an increase in WOB from 40 klbf to 50 klbf. From top to bottom, the plots show the variations of: surface torque, downhole WOB, ROP, downhole motor rotation speed, total rotation speed (surface motor + downhole motor), and the maximum, minimum, and mean bit rotation speed. The stick-slip cycle has a period of approximately 9 seconds. 19

3.2 Examples of 60-second bit rotation speed sequences sampled at 1 Hz: (a) Sequence 1: Severe stick-slip with high SSI; (b) Sequence 2: Smooth drilling with low SSI. 20

3.3 Surface torque variation sequences corresponding to the 60-second bit rotation speed intervals shown in Fig. 3.2, sampled at 1 Hz: (a) Sequence 1: Moderate stick-slip; (b) Sequence 2: Smooth drilling with low TVI. The maximum, minimum, and mean values of surface torque, required for calculating the TVI, are determined over a 60-second interval. 21

3.4 Power spectrum of the surface torque time variations of the two sequences presented in Fig. 3.3. 22

3.5 Example sequence showing 1-Hz field surface data: Surface torque, ROP, and surface rotation speed, as well as 1-Hz downhole data: WOB, motor and bit rotation speeds. The figure also displays the stick-slip indicators: SSI, TVI, and the surface torque power spectrum sum. 24

3.6 Comparison of the three proposed indicators for stick-slip detection. A total of 250 000 sequences, each lasting 60 seconds, from a field well are used to calculate the SSI, TVI, and the surface torque spectrum energy. The categories for SSI are defined as follows: Low (SSI $\leq$ 0.3), Moderate (0.3 < SSI $\leq$ 0.5), Severe (0.5 < SSI $\leq$ 0.8), and Extreme (SSI > 0.8). 25

3.7 Example trajectories of the four wells 1, 5, 6, and 7 described in Table 3.1. . . . 29

3.8 Heatmaps illustrating the relationships between the input features across the entire dataset from all wells: (a) Pearson correlation, (b) Spearman correlation. 31
3.9 Kolmogorov–Smirnov heatmaps of: (a) Surface torque; (b) Surface WOB; (c) ROP; (d) Flow rate; (e) Total rotation speed; and (f) SSI for all wells described in Table 3.1. 32
3.10 Histograms of: (a) Surface torque; (b) Surface WOB; (c) ROP; (d) Flow rate; (e) Total rotation speed; and (f) SSI for wells 5, 7, and 9, described in Table 3.1. 33
3.11 Heatmap of Pearson correlation coefficients between each input feature and the SSI across some test wells. 34
3.12 Distribution of SSI values across all well data described in Table 3.1. 34

4.1 Architecture of a LSTM Unit (Source). 38
4.2 Example of 60-second generated sequence with a SSI = 1.05: (a) Bit and surface rotation speed; (b) Surface torque. The sequence is generated at depth = 2000 ft, using the lateral trajectory of well 4 (see Fig. 3.7). 42
4.3 Baseline model results tested on four wells with different types of normalization: F-by-F Norm (Feature-by-feature normalization), Sequence Norm (Sequence Normalization), DAIN, and Layer Norm (LN), shown for two cases: (a) without TL and (b) with TL application. Wells A, B, C, and D correspond to wells 4, 6, 7, and 9, respectively, described in Table 3.1. 44
4.4 Comparison of the predicted SSI over time for well D, withou TL application, applying different normalization methods:(a) F-by-F Norm (Feature-by-feature normalization); (b) Sequence normalization; (c) DAIN; and (d) LN. 45
4.5 Comparison of the predicted SSI over time for well B using TL, applying different normalization methods:(a) F-by-F Norm (Feature-by-feature normalization); (b) Sequence normalization; (c) DAIN; and (d) LN. 46
4.6 Comparison LSTM model predictions on part of well D: (a) Without TL; (b) With TL. 47
4.7 Comparison of the LSTM model predictions on well A: (a) Without TL; (b) With TL. 48
4.8 MAE evolution over time for the baseline model with LN as the normalization method, with and without TL for Well A. The TL is applied only once, using the first 10 % of well A's drilling data. For the remaining 90 % of the data, the SSI is predicted for each 10 % or 20 % segment, and the mean of MAE of each segment is calculated. 48
4.9 Distribution of true ssi values, calculated using downhole data, within the first 10 % of available drilling data from Wells A and D, used for transfer learning. . 49
4.10 Comparison of TL performance on well A after data augmentation: (a) With the addition of synthetic data; (b) With the addition of sequences from other wells (including a zoomed-in view of a short sequence). 50
4.11 MAE evolution over time for baseline model applied to Well A: with TL: with TL using only the first 10 % of Well A's drilling data; Gen Seq: the first case with additional synthetic sequences; and Other Wells Seq: the first scenario with additional sequences from other wells. The SSI is predicted for each 10% or 20% segment, and the mean MAE of each segment is reported. 51

4.12 MAE values for the hybrid model (with physical features) and the data-based model (without physical features) across the test wells. 52
4.13 Predicted SSI using baseline model over time of sequence presented in Fig. 3.5. 53
4.14 Comparison of the true SSI, predicted SSI using baseline model and TVI over test wells. The categories for SSI are defined as follows: Low (SSI $\leq$ 0.3), Moderate (0.3 < SSI $\leq$ 0.5), Severe (0.5 < SSI $\leq$ 0.8), and Extreme (SSI > 0.8). 54
4.15 Architecture of the Transformer [1]. 55
4.16 True and predicted SSI values over time, for both LSTM baseline model and Transformer-based model of a sequence from the first test well (well 7 in Table. 3.1). 57
4.17 PFI of input surface measurements used by the baseline model to predict the SSI. "Baseline" refers to the baseline model's performance on the test data without any feature shuffling. 58
4.18 Depth variation of total rotation speed in a test well. 59
4.19 PFI of input surface measurements and physical features discussed in Section 4.5, used by the hybrid model to predict the SSI. "Baseline-Hybrid" refers to the model's performance on the test data without any feature shuffling. 60
4.20 Example of real-time application of the proposed model on a test well (Well 8 in Table 3.1). From left to right: bit and hole depth, surface torque, true SSI calculated using Eq. (3.1) from 1-Hz measured downhole data, and predicted SSI obtained with the data-based baseline model. 60

5.1 Architecture of our LSTM baseline model. 69
5.2 Architecture of ADG model. The generator is composed of a sequence of LSTM and LN layers; the SSI predictor and domain classifier are fully connected neural networks; a GRL is used to reverse the direction of the gradients during backpropagation. 70
5.3 Average MSE of SSI validation data predictions for the three tested cases (Table. 5.1) with varying regularization and weighting coefficient over three different intializations: The ADG model is trained for each combination of hyperparameters using three distinct initializations. The validation SSI error is calculated for each one, and the average error is then computed across the three initializations. 73
5.4 Average MSE and normalized DTW of SSI validation data predictions for the three tested cases with varying regularization and weighting coefficient in the ADG model. 74
5.5 Average MSE of SSI validation data predictions for the three tested cases with varying generator hidden layer number and weighting coefficient over three different intializations: The ADG model is trained for each combination of hyperparameters using three distinct initializations. The validation SSI error is calculated for each one, and the average error is then computed across the three initializations. 75
5.6 Average of the MSE and normalized DTW of SSI validation data predictions for the three tested cases with varying generator hidden layer number and weighting coefficient for the ADG model. 76

5.7 Confusion matrices illustrating the performance evaluation of the baseline, IRM and ADG models over the three test wells. The true and predicted SSI values are classified into four categories: sequences with SSI $\in [0, 0.3[$ are assigned to the first class; SSI $\in [0.3, 0.5[$ are classified into the second class; if the SSI $\in [0.5, 0.7[$ belong to the third class; and sequences with SSI ≥ 0.7 are assigned to the fourth class. 77

5.8 True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. In the two sequences, all models accurately predict the SSI. 78

5.9 True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. In the two sequences, all models fail to capture the variations in SSI. 79

5.10 True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. For the three test sequences, the ADG and IRM provide more accurate SSI predictions than the baseline. The last two sequences the ADG model predicts the SSI with slightly greater accuracy than the IRM model. 80

5.11 True and predicted SSI values over time, for both ADG and baseline models applied on sequences from the three test wells before and after the application of TL. 82

5.12 Variation in surface torque (surface measurement) and bit rotation speed (downhole measurements) for a test sequence. This sequence exhibits a synchronization issue between surface and downhole measurements. The predicted SSI for the 60-second red and blue sequences are high (0.77 and 0.84 respectively) due to torque variations. However, when using the mismatched sequence, the true SSI values for the red and blue sequences are 0.1 and 0.92 respectively. Consequently, the model mispredicts the SSI of the red sequence due to the synchronization problem. 84

5.13 Variation in surface torque (measured at the surface) and bit rotation speed (measured downhole) during a test sequence. Stick-slip behavior occurs at the bit, but no surface torque variation is observed (due to friction). The predicted SSI using the trained ADG model for the 60-second red sequence is 0.14 and the true SSI is 0.87. 85

5.14 Variation in surface torque (measured at the surface) and bit rotation speed (measured downhole) during a test sequence. In this example, a peak in downhole bit rotation speed within the 60-second red segment results in a high true SSI (0.72), even though no stick-slip occurs. This makes the sequence mislabeled. 86

6.1 Drill string mass-spring system, where ω_t and τ_t denote the surface rotational speed and surface torque, respectively. Each mass behaves as a torsional pendulum characterized by a angular displacement ϕ_i, linear spring with torsional stiffness K_i, damping coefficient C_i, and rotational inertia I_i, computed using Eq. (6.1). 93

6.2 Friction models [2, 3]. 95

6.3 The two simplified lumped-mass models considered in this work. 97

7.1 Schematic view of PINN [4]: t is the input time; $\hat{\omega}_{\theta,p}$ and $\hat{\Delta}_{\theta,p}$ are the predicted states by the PINN, p are the predicted friction coefficients, defined as additional trainable variables. 110

7.2 True and predicted values of the friction coefficients $(\hat{\mu}_s, \hat{\mu}_d)$ obtained by the PINN applied to 30-second sequences of PDE-based simulated data with a stiction friction model, for two cases: (a) Case 1: PINN using a one-mass nonlinear lumped torsional dynamic model in the loss function; (b) Case 2: PINN using a two-mass nonlinear lumped torsional dynamic model in the loss function. . . . 112

7.3 (a) Low-frequency (1 Hz) and high-frequency (100 Hz) downhole bit rotation speed generated by the PDE-based model with static friction coefficient $\mu_s = 0.15$ and dynamic friction coefficient $\mu_d = 0.12$; (b) Comparison of the 1-Hz true bit rotation speed and the predicted bit rotation speed by the one-mass PINN, with predicted friction coefficients $\hat{\mu}_s = 0.152$ and $\hat{\mu}_d = 0.114$. 113

7.4 Frequency spectra of (a) the true and predicted bit rotation speeds shown in Fig. 7.3 at 1 Hz, and (b) the true bit rotation speed at 100 Hz, plotted with a logarithmic x-axis, highlighting the PINN's ability to capture key dynamic features. 113

7.5 Surface torque variations of the original sequence used to train the source model, and the new sequence used for transfer learning application. 114

7.6 Bit rotation speed predicted by the PINN after TL application. 114

7.7 True and predicted friction coefficients (μ_s, μ_d) of the PDE-generated test sequences using the one-mass PINN with transfer learning. 115

7.8 Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained by the PINN, using real surface torque and surface rotational speed as inputs for test well 1. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; Surface RPM - surface rotational speed; DH RPM range - maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM - mean of measured downhole bit rotation speed; DH motor RPM - measured downhole motor rotation speed; Predicted DH RPM - predicted downhole bit rotation speed by the PINN. 116

7.9 Variations of downhole bit rotation speed measurements and predicted $\hat{\omega}_b$ values by the PINN for tets well 1 over six drilling days. The moving minimum, maximum, and average of both the measured and predicted bit rotation speeds, along with the moving average of the downhole motor rotation speed, are calculated over a 120-second window. Definitions of the terms shown: DH RPM range - maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM - mean of measured downhole bit rotation speed; DH motor RPM - measured downhole motor rotation speed; Predicted DH max, mean, and min RPM - moving maximum, mean, and minimum of the predicted downhole bit rotation speed by PINN within the 120-second window. 117

8.1 Simplified schematic of a two-mass rotational system, with inertia I_1 ans I_2, angular velocities ω_1 and ω_2, external torque τ, connected by a spring with stiffness K. 125

8.2 Limitation of the SKF: when the predicted state $\hat{x}_{k|k-1}$ lies in the stick phase ($q = 0$), i.e., $\hat{x}_{k|k-1} \in D_0$, the correction term may push the updated state outside D_0 into D_1 or D_{-1} without crossing the corresponding guard conditions $g_{(0,1)}$ or $g_{(0,-1)}$. This occurs through a violation of the equality constraint defining D_0 ($\omega_2 = 0$). 127
8.3 Variability domain of ω_b in the stick phase: (a) Without extension of f_0: $\omega_2 = 0$ until the external force $K\Delta$ surpasses $\mu_s F^N$; (b) With extension of f_0: ω_2 can evolve over all of $\mathbb{R}$ due to the correction term, while still in the 'stick' mode, until $K\Delta > \mu_s F^N$. 128
8.4 Estimated and true static friction coefficient using SKF and PSKF in Case 1,with the one-mass lumped torsional dynamic model. 136
8.5 Estimated and true dynamic friction coefficient using SKF and PSKF in Case 1, with the one-mass lumped torsional dynamic model. 136
8.6 Estimated and true static friction coefficient using SKF and PSKF in Case 2, with the two-mass lumped torsional dynamic model. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences. 136
8.7 Estimated and true dynamic friction coefficient using SKF and PSKF in Case 2, with the two-mass lumped torsional dynamic model. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences. 137
8.8 Comparison of estimated and true bit rotation speed using SKF and PSKF for a test sequence in both cases: (a) Case 1: using the one-mass lumped torsional dynamic model; (b) Case 2: using the two-mass lumped torsional dynamic model. 137
8.9 Comparison of estimated and true friction coefficients using SKF and PSKF for a test sequence in Case 1. 138
8.10 Comparison of estimated and true friction coefficients using SKF and PSKF for a test sequence in Case 2. 138
8.11 Comparison of estimated and true friction coefficients using PSKF with the three-mass lumped torsional dynamic model at two update frequencies: (a) 100 Hz and (b) 1000 Hz. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences. 142
8.12 Comparison of estimated and true friction coefficients using PSKF with the four-mass lumped torsional dynamic model at two update frequencies: (a) 100 Hz and (b) 1000 Hz. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences. 143
8.13 Comparison of estimated and true bit angular velocity using PSKF with a three-mass model and 100 Hz update frequency for a test sequence. 143
8.14 Comparison of estimated and true friction coefficients over time using PSKF with three-mass lumped model at 100-Hz update frequency for a test sequence. 144
8.15 Comparison of estimated and true bit angular velocity using PSKF with a four-mass model and a 1000-Hz update frequency. Two sets of initial friction coefficient values are tested for the same sequence: Sequence 1: $\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$, and sequence 2 : $\mu_s^{Init} = 0.3$, $\mu_d^{Init} = 0.15$. 145
8.16 Comparison of estimated and true friction coefficients using PSKF with a four-mass lumped torsional dynamic model and 1000-Hz update frequency for a test sequence. The two subfigures show results with different initializations. 146

8.17 The three regions used for generating random initializations of the static friction coefficient values: Region I (Far): $\mu_s^{Init} \in [0, 0.5\,\mu_{s_i}] \cup [1.5\,\mu_{s_i}, 1]$; Region II (Middle): $\mu_s^{Init} \in [0.5\,\mu_{s_i}, 0.7\,\mu_{s_i}] \cup [1.3\,\mu_{s_i}, 1.5\,\mu_{s_i}]$; Region III (Close): $\mu_s^{Init} \in [0.7\,\mu_{s_i}, 0.9\,\mu_{s_i}] \cup [1.1\,\mu_{s_i}, 1.3\,\mu_{s_i}]$. μ_s^{Init} is the initial value. The red cross in the figure represents the true static friction coefficient. 146

8.17 Effect of physical model complexity on the basin of attraction of the PSKF for estimating the static friction coefficient μ_s. The experiments consider three test sequences with true values of $\mu_s = 0.15$, $\mu_s = 0.3$, and $\mu_s = 0.4$, respectively. Each circle represents a distinct initialization run, while the red cross indicates the true value. The background shading denotes the initialization regions: blue for values close to the true μ_s, green for moderate deviations, and red for distant initializations. The estimation error is defined as Error $= |\mu_s - \hat{\mu}_s|$. 149

8.18 MAE of the predicted static (μ_s) and dynamic (μ_d) friction coefficients using the PSKF applied to three test sequences with ground truth values: $\{\mu_{s_1} = 0.15,\ \mu_{d_1} = 0.075\}$, $\{\mu_{s_2} = 0.3,\ \mu_{d_2} = 0.165\}$, and $\{\mu_{s_3} = 0.4,\ \mu_{d_3} = 0.24\}$. Each sequence is processed using four physical models: one-mass, two-mass, three-mass, and four-mass systems. For each sequence, 100 runs are conducted with randomly selected initializations drawn from three predefined regions: Far, middle, and close. The MAE values represent the average error over the 100 runs for each region. 150

8.19 Trajectory of the test well used for generating sequences in the PDE-based model simulations. 151

8.20 Comparison of estimated and true friction coefficients using PSKF with a one-mass lumped torsional dynamic embedded model and 1-Hz update frequency. The simulated data are generated using a PDE-based model. Results are shown for two different initialization strategies: fixed values vs. scaled true values. . . 153

8.21 Example of one-mass PSKF performance on a test sequence generated using a PDE-based model at 1-Hz update frequency: (a) shows the estimated and true bit rotation speed over time; and (b) presents the estimated and true static and dynamic friction coefficients. 153

8.22 Frequency spectra of (a) the true and predicted bit rotation speeds shown in Fig. 8.21 at 1 Hz, and (b) the true bit rotation speed at 100 Hz, plotted with a logarithmic x-axis. 154

8.23 Comparison between actual bit rotation speed measurements and PSKF predictions using 1-Hz surface torque and surface rotational speed as inputs, for a test sequence from Well 7 (see Table 3.1). 'DH TOB range' refers to the max–min range of downhole measured TOB, and 'DH mean TOB' to its mean value. Similarly, 'DH RPM range' and 'DH mean RPM' represent the range and mean of the measured downhole bit rotation speed. 'DH motor RPM' corresponds to the downhole measured motor rotational speed. 156

9.1 Predicted static and dynamic friction coefficients for the test PDE-generated sequences, obtained using the three methods: PINN, PSKF, and *nlgreyest*. . . . 160

9.2 MAE and MAPE of the three methods, PINN, PSKF, and *nlgreyest*, for the predicted static (μ_s) and dynamic (μ_d) friction coefficients. 161

9.3 Inclination of test wells 1, 5, 6, and 7 with depth. Red dots indicate the available data for each well, which are used to test both methods, PINN and PSKF. . . . 164

9.4 Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with the PINN, using real surface torque and surface rotational speed as inputs for test well 1. The measured bit rotation speed is recorded by downhole sensors and received after drilling. Definitions of the terms shown: Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by PINN. 165

9.5 Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with the PINN, using real surface torque and surface rotational speed as inputs for test well 7. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by the PINN. 166

9.6 Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with PSKF, using real surface torque and surface rotational speed as inputs for: (a) test well 1 and (b) test well 7. Definitions of the terms shown: Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by PINN. 167

9.7 Variations of downhole bit rotation speed measurements and predicted values $\hat{\omega}_b$ obtained with (a) PINN and (b) PSKF, using the real surface torque and surface rotational speed as inputs for test wells 1,7,6,5, respectively. The moving minimum, maximum, and average of both the measured and predicted bit rotation speeds, along with the moving average of the downhole motor rotation speed, are calculated over a 120-second window. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH max, mean, and min RPM – moving maximum, mean, and minimum of the predicted downhole bit rotation speed by PINN within the 120-second window. 172

9.8 Prediction results of bit rotation speed and friction coefficients for a sequence generated using the two-mass lumped torsional dynamic model (Eq. (6.8)), with the PSKF embedded with the same physical model. The surface rotation speed is constant to 60 rpm. Shown are: (a) Surface torque; and (b) true and predicted bit rotation speed. 176

9.9 True and predicted friction coefficient variations over time at 1 Hz, corresponding to the sequence shown in Fig. 9.8. 177

List of tables

3.1 Summary of well characteristics, where vertical wells are drilled straight down from the surface into the target formation or reservoir, and lateral wells are drilled vertically to a certain depth, then deviates horizontally within the target. 28

3.2 Summary statistics of input and target characteristics. The total number of data points is 1 633 596, where each data point corresponds to one timestamped measurement across the variables. 28

4.1 MAE values of SSI prediction using the baseline model with LN as the normalization method, with and without transfer learning. Wells A, B, C, and D correspond to wells 4, 6, 7, and 9, respectively, described in Table 3.1. 47

4.2 Normalized DTW calculated between the true and predicted SSI for the trained LSTM baseline and Transformer-based models on the three test wells. Test wells 1, 2, and 3 correspond, respectively, to wells 7, 8, and 9 in Table 3.1. . . . 57

5.1 Validation data selection for the three tested cases. 72

5.2 Normalized DTW of the trained ADG, IRM and baseline models on the three test wells: Each model is trained with five different initializations, and the error is reported as the average across these five runs. 81

5.3 Comparison of model performance, for both ADG and baseline models, before and after applying TL, using normalized DTW as the evaluation metric across the three test wells: Each source model undergoes partial retraining with the first 10 % of labeled surface drilling data from each test well. 81

7.1 MAE and MAPE of the friction coefficients predicted by the PINN across all test sequences of the two cases: Case 1: PINN using a one-mass nonlinear lumped torsional dynamic model in the loss function; Case 2: PINN using a two-mass nonlinear lumped torsional dynamic model in the loss function. . . . 112

7.2 MAE and MAPE of the predicted friction coefficients using the one-mass PINN across all test sequences, with and without transfer learning. 115

7.3 MAE and Pearson correlation coefficient (r) between the moving maximum, mean, and minimum of true and predicted bit rotation speeds over the long six-day sequences presented in Fig. 7.9 for test well 1. 117

8.1 Parameter values of the one-mass lumped torsional model used in Case 1 Eq. (6.7). 133

8.2 Parameter values of the two-mass lumped torsional model used in Case 2 Eq. (6.8). 133

8.3 MAE and MAPE of the friction coefficients estimated using SKF and PSKF across all test sequences of the two cases: Case 1: using the one-mass lumped torsional dynamic model; Case 2: using the two-mass lumped torsional dynamic model. . . . 135
8.4 Parameter values of the three-mass lumped torsional model Eq. (8.38) with $N = 3$. 141
8.5 Parameter values of the four-mass lumped torsional model Eq. (8.38) with $N = 4$. 141
8.6 PSKF performance on static friction coefficient estimation for different physical models and update frequencies, For the one-mass and two-mass lumped models, only a 1-Hz update frequency is tested, as it is sufficient to accurately predict the friction coefficients using the PSKF observer. . . . 142
8.7 PSKF performance on dynamic friction coefficient estimation for different physical models and update frequencies. . . . 142
8.8 Physical parameters of the one-mass and two-mass lumped torsional dynamic models used for the test well. For the one-mass model, parameters represent the entire drill string. In the two-mass model, the drill string is divided into vertical (I_1, K_1, F_1^N, L_1) and lateral (I_2, K_2, F_2^N, L_2) sections. . . . 152
8.9 Comparison of one-mass PSKF performance for friction coefficient estimation at 1-Hz update frequency on simulated data generated from a PDE-based model. Results are presented for two initialization strategies: fixed initial values ($\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$), and scaled initial values ($\mu_s^{Init} = 0.5\mu_s$, $\mu_d^{Init} = 0.5\mu_d$), where μ_s and μ_d are the true values. . . . 152

9.1 Benchmark of PINN, PSKF, and nlgreyest approaches based on accuracy, computational efficiency, and model sensitivity. . . . 162
9.2 MAE, normalized DTW, and Pearson correlation coefficient (r) between the mean, maximum, and minimum field downhole measured rotation speeds and the predicted bit rotation speed by both methods, PINN and PSKF, for the short test sequence of well 1. Improvement is calculated using Eq. (9.1). . . . 167
9.3 MAE, normalized DTW, and Pearson correlation coefficient (r) between the mean, maximum, and minimum field downhole measured rotation speeds and the predicted bit rotation speed by both methods, PINN and PSKF, for the short test sequence of well 7. . . . 168
9.4 MAE, Pearson correlation coefficient (r), and normalized DTW between the mean, maximum, and minimum of the field downhole measured rotation speeds and those of the predicted bit rotation speeds obtained with both methods, PINN and PSKF, for: (a) test well 1, (b) test well 7, (c) test well 6, and (d) test well 5. The improvement row is computed as relative error reduction with respect to PSKF using Eq. (9.1). . . . 173
9.5 Parameter values of the two-mass lumped torsional model (Eq. (6.8)) used to generate the test sequence shown in Fig. 9.8. . . . 175

Acronyms

ADA Adversarial Domain Adaptation

ADG Adversarial Domain Generalization

BHA Bottom-Hole Assembly

D-WIS Drilling and Wells Interoperability Standard Industry

DA Domain Adaptation

DAIN Deep Adaptive Input Normalization

DG Domain Generalization

DSE Dynamic State Estimation

DTW Dynamic Time Warping

ECDF Empirical Cumulative Distribution Functions

EKF Extended Kalman Filter

ERM Empirical Risk Minimization

FCNN Fully Connected Neural Network

FDM Finite Difference Method

FEA Finite Element Analysis

GP Gaussian Process

GRL Gradient Reversal Layer

IRM Invariant Risk Minimization

KS Kolmogorov-Smirnov

LN Layer Normalization

LSTM Long Short Term Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

ML Machine Learning

MSE Mean Squared Error

MWD Measurements While Drilling

NER Named Entity Recognition

NLP Natural Language Processing

ODE Ordinary Differential Equation

PCA Principal Component Analysis

PDE Partial Differential Equation

PFI Permutation Feature Importance

PIML Physics-Informed Machine Learning

PINN Physics-Informed Neural Network

PSKF Projected Salted Kalman Filter

RNN Recurrent Neural Network

ROP Rate Of Penetration

SKF Salted Kalman Filter

SSI Stick-Slip Index

SVM Support Vector Machine

TL Transfer Learning

TOB Torque-On-Bit

TVI Torque Variation Index

WITSML Wellsite Information Transfer Standard Markup Language

WOB Weight-On-Bit

Chapter 1

Introduction - version française

1.1 Contexte général

Les systèmes d'ingénierie modernes, allant de l'aérospatiale et de l'énergie à la fabrication et aux technologies autonomes, dépendent de plus en plus d'algorithmes avancés pour la surveillance, le contrôle et l'optimisation en temps réel. Un défi central pour ces algorithmes est l'estimation des états internes du système ou des variables qui ne sont pas directement mesurables. L'inférence précise et efficace de ces grandeurs non mesurées à partir des données disponibles des capteurs est cruciale pour garantir une prise de décision sûre et intelligente.

Pour y répondre, un large éventail de techniques d'estimation a été développé, dont beaucoup s'appuient sur les *données temporelles*, c'est-à-dire des séquences d'observations numériques collectées au fil du temps à partir de variables mesurables du système. Contrairement à des échantillons aléatoires, les données temporelles sont intrinsèquement ordonnées, et cette structure temporelle est essentielle pour capturer la dynamique des systèmes en évolution [5, 6]. En conséquence, la modélisation des séries temporelles est devenue un outil fondamental dans divers secteurs industriels et domaines de recherche.

La disponibilité croissante des données issues des capteurs et l'amélioration des capacités de calcul ont accéléré l'adoption des techniques d'apprentissage automatique ou "Machine Learning (ML)" pour prédire les variables non mesurées du système. En particulier, des architectures spécialisées de réseaux de neurones ont émergé pour apprendre à partir de séquences temporelles. Parmi elles, les réseaux de neurones récurrents (RNNs) se distinguent par leur capacité à modéliser les dépendances séquentielles grâce à des boucles de rétroaction au sein de l'architecture du réseau [7, 8]. Cependant, les RNNs standard sont limitées par le problème de gradients qui disparaissent ou explosent lors de l'entraînement par rétropropagation à travers le temps [9], ce qui entrave leur capacité à conserver l'information à long terme et conduit à un apprentissage instable ou inefficace.

Pour surmonter ce problème, les réseaux récurrents à mémoire court et long terme ou "Long Short Term Memory (LSTM)" ont été introduits par Hochreiter and Schmidhuber [10]. Les LSTMs améliorent les RNNs en incorporant des cellules de mémoire et des mécanismes de

portes qui régulent le flux d'information, permettant ainsi au réseau d'apprendre efficacement les dépendances à long terme. Par conséquent, les LSTM sont devenus un outil standard pour la prévision de séries temporelles et la modélisation de séquences dans de nombreux domaines d'application.

Une autre architecture appliquée avec succès aux tâches de séries temporelles est le réseau neuronal convolutif o "Convolutional Neural Network (CNN)" [11]. Initialement développés pour la reconnaissance d'images, les CNNs ont montré de solides performances dans la classification et la prévision de séries temporelles multivariées [12]. Les Auto-encodeurs ont également contribué de manière significative à la modélisation des séries temporelles, notamment dans des contextes d'apprentissage non supervisé tels que la détection d'anomalies et la réduction de dimensionnalité [13]. Plus récemment, l'architecture Transformer [1] a suscité un intérêt croissant pour ses performances supérieures en traitement automatique des langues ou "Natural Language Processing (NLP)" et en vision par ordinateur. Son succès a également éveillé l'intérêt pour la modélisation des séries temporelles. Contrairement aux RNNs et LSTMs, les Transformers ne reposent pas sur la récurrence ; ils utilisent des mécanismes d'auto-attention et un encodage positionnel pour modéliser les relations séquentielles. Cette architecture permet un entraînement hautement parallélisé et s'avère efficace pour capturer les dépendances à long terme dans les données temporelles.

Cependant, un défi commun aux modèles ML présentés est leur *généralisabilité*, c'est-à-dire la capacité d'un modèle entraîné à bien performer sur des données de test non vues, c'est à dire dont la distribution diffère de celle des données d'entraînement. En pratique, l'hypothèse courante selon laquelle les données d'entraînement et de test partagent une distribution similaire est souvent violée. Améliorer la généralisabilité repose généralement sur l'augmentation de la quantité et de la diversité des données d'entraînement. Toutefois, lorsque seules des données limitées sont disponibles, cette approche n'est pas envisageable. Dans ce cas, des stratégies alternatives, incluant des techniques d'analyse des données, la conception de l'architecture du modèle et les procédures d'entraînement, peuvent être appliquées pour renforcer la généralisabilité.

Alors que certaines méthodes ont été testées sur des ensembles d'images, cette dissertation se concentre sur les séries temporelles. Ainsi, l'objectif de **la première partie** est de répondre à la question suivante :

> *Quelles stratégies permettent d'améliorer la performance des modèles ML sur des séries temporelles non vues ?*

Nous abordons cette question dans le contexte du forage, où l'objectif est d'entraîner un modèle ML "data-driven" généralisable pour estimer la gravité des vibrations torsionnelles dans le puits à partir de séquences de mesures en surface. Les modèles sont entraînés sur des données de terrain limitées collectées sur certains puits et testés sur des données d'autres puits situés dans différentes régions géographiques, afin d'évaluer leur généralisabilité.

Dans **la deuxième partie** de cette dissertation, nous changeons de focus. Alors que les approches évoquées ci-dessus relèvent des modèles "data-driven", dont le but est d'estimer des variables non mesurées uniquement à partir des données observées, de nombreux scénarios réels nécessitent de dépasser l'estimation et d'entrer dans le domaine de *l'identification de systèmes*. L'objectif ici est de retrouver des paramètres inconnus, des états cachés, voire les équations régissant un système. Cela s'obtient en combinant les données avec une connaissance a priori de la structure physique du système.

À cette fin, des modèles hybrides ML ont été développés. Parmi eux, les Réseaux de neurones informés par la physique ou "Physics-Informed Neural Network" (PINNs) [14] constituent une approche puissante qui intègre directement les lois physiques, typiquement exprimées sous forme d'équations aux dérivées partielles (EDP), dans le processus d'entraînement. Les PINNs réalisent cela en incorporant les résidus des EDPs, la perte supervisée sur les variables cibles, ainsi que les conditions initiales et aux limites, dans la fonction de perte. Plutôt que d'apprendre uniquement à partir des données, les PINNs sont guidées par les équations régissant le système, ce qui leur permet de résoudre à la fois des problèmes directs et inverses.

Dans la littérature classique sur la théorie du contrôle, la tâche d'identification de systèmes est parfois formulée comme un problème de *conception d'observateur*. L'objectif est de reconstruire l'état interne d'un système dynamique à partir des seules mesures disponibles. Les paramètres inconnus du modèle peuvent être incorporés dans le vecteur d'état. Alors que la conception d'observateurs linéaires est bien établie [15], la conception d'observateurs pour des systèmes non linéaires reste un domaine de recherche complexe et actif. En effet, un cadre universel pour la conception d'observateurs pour systèmes non linéaires n'existe pas [16]. Néanmoins, plusieurs méthodes efficaces ont été proposées, notamment les observateurs à gain élevé [17] et le filtre de Kalman étendu ou "Extended Kalman Filter (EKF)" [18], qui offrent des solutions approximatives pour l'estimation d'état non linéaire sous certaines hypothèses.

Ces approches classiques d'observateurs sont principalement adaptées aux systèmes à dynamique lisse. Cependant, de nombreux systèmes présentent un comportement non lisse, combinant évolution continue et événements discrets tels que commutations, transitions logiques ou sauts. L'estimation d'état pour ces systèmes [19] introduit une complexité supplémentaire, car les observateurs doivent tenir compte des sauts d'état brusques, des changements de mode ou des transitions hybrides, en plus de suivre la dynamique continue. Les techniques d'estimation traditionnelles sont souvent insuffisantes pour gérer ces discontinuités et comportements mixtes.

Pour relever ces défis, des cadres d'observateurs spécialisés ont été développés spécifiquement pour les systèmes dynamiques hybrides. Par exemple, Ackerson and Fu [20] figurent parmi les premiers à formuler le problème d'estimation d'état pour les systèmes à commutation. Plus tard, Liu [21] introduisent des observateurs à gain commuté conçus pour les systèmes non linéaires avec comportements à commutation. Plus récemment, Tran and Bernard [22] proposent un observateur hybride de type Kalman adapté aux systèmes hybrides généraux avec dynamique linéaire, en supposant que les instants de saut des solutions sont connus exactement.

Dans cette dissertation, nous concentrons sur le "Salted Kalman Filter (SKF)" [23], un observateur conçu pour les modèles hybrides avec instants de saut inconnus. Les deux approches, modèles hybrides ou "machine learning (ML)" et théorie des observateurs, reposent sur un modèle physique et des données d'entrée mesurées. Des difficultés apparaissent lorsque le modèle physique incorporé est non linéaire et que les données d'entrée mesurées ne sont disponibles qu'à faible fréquence d'échantillonnage, impliquant un contenu informationnel limité.

Ainsi, dans la deuxième partie de cette thèse, nous abordons un problème d'identification de système impliquant un modèle de friction non linéaire et des mesures de données rares. Nous adaptons et appliquons des méthodes issues à la fois de l'apprentissage automatique, via l'utilisation des PINNs, et de la théorie des observateurs, à travers le Salted Kalman Filter, à ce problème et en évaluons les mérites.

Nous traitons ce problème dans le contexte du forage, où l'objectif est de prédire la vitesse de rotation dans le puits et les coefficients de friction entre la train de tiges de forage et le trou de forage, à partir de mesures en surface à basse fréquence. Le modèle physique utilisé dans

les deux méthodes inclut une non-linéarité représentant le frottement de Coulomb. Les deux approches sont comparées et testées sur des données de terrain provenant de différents puits.

1.2 Contexte du forage

Le forage rotatif est la méthode la plus couramment utilisée dans les applications pétrolières, gazières et géothermiques. Comme illustré à la Fig. 1.1, il consiste à utiliser un outil de forage appelé trépan, qui détruit généralement la roche selon l'une des deux méthodes suivantes : soit par cisaillement (trépan à traînée), soit par indentation (trépan à rouleaux coniques). Le trépan est relié à une installation de surface par une tige de forage constituée d'une série de tubages et d'une partie inférieure appelée "Bottom-Hole-Assembly (BHA)". Le trépan doit être mis en rotation et poussé contre la formation rocheuse. La force à l'outil, ou "Weight-On-Bit (WOB)", nécessaire pour forer est obtenue grâce au poids de la tige de forage.

La rotation est appliquée soit à l'aide d'une table rotative, soit d'un "top drive" en surface, parfois assistée par un moteur situé dans la BHA selon certaines applications. Les moteurs au fond de trou sont particulièrement utilisés pour les puits directionnels (latéraux) et lorsque la technologie du trépan nécessite des vitesses de rotation plus élevées, impossibles à atteindre uniquement avec la rotation en surface. Ils permettent d'éviter la torsion excessive de la tige en concentrant la majeure partie du couple rotatif près de l'outil, plutôt que de le perdre à travers le couple et la traînée sur la tige [24].

De plus, des fluides de forage et un système de circulation sont utilisés pour suspendre et transporter les déblais, ainsi que pour refroidir l'outil et les équipements au fond de trou. Le fluide de forage, également appelé boue de forage, est injecté vers le bas à l'aide de pompes à haute pression et débit, et retourne en surface en emportant les déblais à travers l'espace annulaire entre la tige et le puits.

D'autres équipements peuvent être utilisés pour améliorer l'efficacité du forage, tels que les "shock subs" pour amortir les vibrations lors du forage de formations dures, ou les "drilling jars", utilisés pour libérer le tubage en appliquant un choc lorsqu'il est bloqué [25].

Lors du forage, différents types de vibrations, torsionnelles, latérales et axiales, peuvent apparaître de façons indépendantes ou combinées. Détecter et contrôler leur intensité est essentiel pour assurer l'efficacité du forage et éviter les dommages aux équipements. À cette fin, des capteurs avancés installés au fond de trou, proche de l'outil, sont utilisés pour mesurer des paramètres critiques tels que le couple et la vitesse de rotation. Ces mesures peuvent être stockées en mémoire interne et récupérées quand la BHA est sortie du trou[1], ou transmises en temps réel à la surface via la télémetrie par impulsions dans la boue. Cependant, ces capteurs sont coûteux, et la transmission des données en temps réel est limitée par la faible bande passante de la télémetrie. Par conséquent, seules certaines données à basse fréquence peuvent être transmises en temps réel, tandis que les données à haute fréquence sont généralement utilisées pour l'analyse post-forage.

Étant donné le coût élevé et la disponibilité limitée en temps réel des mesures au fond de trou, l'utilisation des mesures en surface pour l'estimation et la prédiction des paramètres du puits, tels que les vibrations des équipents de forage, est devenue un domaine de recherche

[1] En terminologie de forage, un "run" correspond à la période d'opération pendant laquelle une BHA est déployée dans le puits et reste au fond de trou jusqu'à son retrait pour maintenance, remplacement ou à la fin d'une phase de forage.

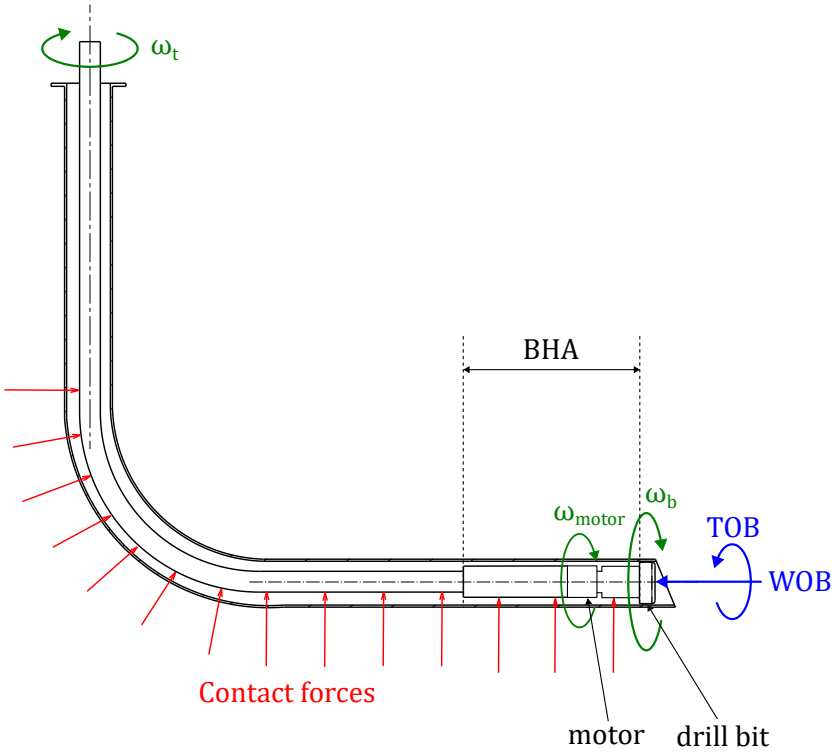


Figure 1.1: Représentation schématique simplifiée du système de forage : BHA désigne la partie inférieure de la tige de forage; ω_t, ω_{Motor} et ω_b représentent respectivement les vitesses de rotation en surface, du moteur de fond et de l'outil; couple à l'outil ou "Torque-On-Bit (TOB) et WOB correspondent respectivement au couple et au poids sur l'outil.

important. De nombreuses études ont proposé des approches "data-driven" pour relever ce défi, notamment en exploitant les données de forage collectées par les capteurs de la plateforme de forage, telles que le couple, la vitesse de rotation, le poids au crochet, la position du bloc et le débit de la boue. Cela a conduit à une tendance croissante à tirer parti des données de surface pour optimiser la performance du forage et réduire les problèmes opérationnels.

1.3 Plan de la thèse

Cette thèse est divisée en deux grandes parties. La première partie se concentre sur l'amélioration de la généralisabilité des modèles ML pour prédire la gravité des vibrations torsionnelles au fond du puits à partir de séquences de mesures de surface à 1 Hz. Le défi central est de s'assurer que le modèle entraîné performe bien sur des données de forage non vues provenant de puits différents de ceux utilisés pour l'entraînement. Pour y répondre, plusieurs stratégies sont explorées :

- Le Chapitre 3 introduit les vibrations torsionnelles au fond de puits, les méthodes pour les détecter à l'aide de mesures de surface et de fond de trou, ainsi que les défis liés à la construction de modèles ML généralisables à des puits non vus. Il décrit également les jeux de données de forage utilisés dans cette thèse.

- Le Chapitre 4 présente un modèle de référence ("baseline") servant de point de comparaison. Quatre stratégies pour améliorer la généralisabilité sont ensuite évaluées : (i) la normalisation des données, appliquée pendant ou avant l'entraînement afin de réduire les écarts de distribution entre les données d'entraînement et de test ; (ii) le "transfer learning (TL)", où une petite quantité de données du puits cible est utilisée pour affiner un modèle préalablement entraîné ; (iii) l'augmentation de données, à partir de données simulées ou de terrain, pour traiter le déséquilibre des classes de sévérité des vibrations torsionnelles ; et (iv) l'intégration de caractéristiques physiques issues de modèles physiques qui encodent des informations sur la géométrie du puits, la composition de la tige et la trajectoire. De plus, le modèle de référence est comparé à une architecture basée sur Transformer.

- Le Chapitre 5 explore des stratégies plus avancées pour améliorer la généralisabilité des modèles. Deux approches principales sont étudiées : la généralisation de domaine adversariale ou "Adversarial Domain Generalization (ADG)", qui cherche à apprendre des représentations invariantes au domaine, et la minimisation du risque invariant ou "Invariant Risk Minimization (IRM)", visant à extraire des caractéristiques prédictives stables à travers différents environnements de données.

La deuxième partie de la thèse se concentre sur la prédiction de la vitesse de rotation de l'outil et des coefficients de frottement à partir de mesures de surface à 1 Hz, assistée par un modèle physique simplifié. Cette partie comprend les contributions suivantes :

- Le Chapitre 6 introduit le problème, passe en revue les modèles existants et leurs limites, et présente notre modèle physique simplifié ainsi que les hypothèses formulées. Il décrit également nos deux solutions proposées.

- Le Chapitre 7 détaille la première solution, un réseau de neurones physique-informé (PINN). Dans cette méthode, le modèle physique simplifié est intégré directement dans la fonction de perte du réseau. Le PINN est testé sur des données simulées et de terrain, et constitue une solution hors ligne.

- Le Chapitre 8 présente la seconde solution, le "Projected Salted Kalman Filter (PSKF)", une extension du "Salted Kalman Filter (SKF)". Comme le PINN, il est validé sur des données simulées et de terrain. Contrairement au PINN, le PSKF fonctionne en temps réel.

- Le Chapitre 9 compare les deux approches, PINN et PSKF, sur les jeux de données simulées et de terrain. Pour les simulations, une troisième méthode à l'état de l'art est incluse pour référence. Pour la validation sur le terrain, des données provenant de quatre puits différents sont utilisées.

1.4 Publications

Certaines des résultats présentés dans cette thèse ont été publiés. Ils sont listés ci-dessous en fonction des chapitres correspondants :

- Chapitre 4:
  [26] Hana Yahia, Thomas Romary, Laurent Gerbaud, Bruno Figluizzi, Florent Di Meglio, Stephane Menand, and Mohamed Mahjoub. Combining machine-learning and physics-based models to mitigate stick-slip in real-time. In *SPE/IADC Drilling Conference and Exhibition*. SPE, 2024

  [27] Hana Yahia, Thomas Romary, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Real-time stick-slip mitigation using combined machine learning and physics based techniques. In *International Petroleum Technology Conference*. IPTC, 2024

- Chapitre 5:
  [28] Hana YAHIA, Bruno Figliuzzi, Florent Di Meglio, Stephane Menand, Mohamed MAHJOUB, et al. Domain generalization for time series: Enhancing drilling regression models for stick-slip index prediction. *Transactions on Machine Learning Research*

- Chapitre 7:
  [29] Hana Yahia, Florent DI Meglio, Bruno Figliuzzi, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Physics-informed neural network to estimate downhole torsional vibrations and friction coefficients. In *Abu Dhabi International Petroleum Exhibition and Conference*. SPE, 2025

Chapter 2

Introduction

2.1 General context

Modern engineering systems, ranging from aerospace and energy to manufacturing and autonomous technologies, are increasingly dependent on advanced algorithms for real-time monitoring, control, and optimization. A central challenge faced by these algorithms is the estimation of internal system states or variables that are not directly measurable. Accurate and efficient inference of these unmeasured quantities from available sensor data is critical to ensure safe and intelligent decision-making. To address this, a wide range of estimation techniques are developed, many of which rely on *time series data*, i.e., sequences of numerical observations collected over time from measurable system variables. Unlike random samples, time series data is inherently ordered, this temporal structure is essential for capturing the dynamics of evolving systems [5, 6]. As a result, time series modeling has become a foundational tool across various sectors of industry and research.

The growing availability of sensor data and increased computational capabilities have accelerated the adoption of *data-driven* machine learning (ML) techniques for predicting unmeasured system variables. In particular, specialized neural network architectures have emerged to learn from temporal sequences. Among these, Recurrent Neural Networks (RNNs) stand out for their ability to model sequential dependencies using feedback loops within the network architecture [7, 8]. However, standard Recurrent Neural Networks (RNNs) are limited by the vanishing or exploding gradient problem during training via backpropagation through time [9], which obstructs their ability to retain long-term information and leads to unstable or ineffective learning. To overcome this issue, the Long Short Term Memory (LSTM) networks are introduced by Hochreiter and Schmidhuber [10] as an enhancement of RNN. They introduce memory cells and gating mechanisms to regulate the flow of information, thereby enabling the network to effectively capture long-term dependencies. Consequently, LSTMs have become a standard tool in time series forecasting and sequence modeling across multiple application domains. Another architecture successfully applied to time series tasks is the Convolutional Neural Network (CNN) [11]. Originally developed for image recognition, CNNs have demonstrated

strong performance in multivariate time series classification and forecasting [12]. Autoencoders have also contributed significantly to time series modeling, particularly in unsupervised learning contexts such as anomaly detection and dimensionality reduction [13]. More recently, the Transformer architecture [1] has gained widespread attention for its superior performance in Natural Language Processing (NLP) and computer vision. Its success has sparked interest in time series modeling as well. Unlike RNNs and LSTMs, Transformers do not rely on recurrence, instead, they use self-attention mechanisms and positional encoding to model sequential relationships. This architecture allows for highly parallelized training and has proven effective at capturing long-range dependencies in temporal data.

A common challenge for the presented ML models is their *generalizability*, that is, the ability of a trained model to perform well on unseen test data whose distribution differs from that of the training data. In practice, the common assumption that training and test data share a similar distribution is often violated. Typically, improving generalizability relies on increasing the quantity and diversity of the training data. However, when only limited training data is available, this approach is not feasible. In such cases, alternative strategies, including data analysis techniques, model architecture design, and training procedures, can be applied to enhance generalizability. While some methods have been tested on image datasets, in this dissertation we focus on time series data. Therefore, the objective of **the first part** is to address the following question:

> *What strategies can improve the performance of ML models on unseen time series data?*

We address this question in the context of drilling, where the objective is to train generalizable data-driven ML model to estimate the severity of torsional downhole vibrations from sequences of surface measurements. The models are trained on limited field data collected from certain wells and subsequently tested on data from other wells located in different geographical regions, in order to evaluate their generalizability.

In **the second part** of this dissertation, we shift our focus. While the approaches discussed above fall under the category of data-driven models, where the goal is to estimate unmeasured variables using only observed data, many real-world scenarios require going beyond estimation and entering the domain of *system identification*. The aim here is to recover unknown parameters, hidden states, or even the governing equations of a system. This is achieved by combining data with prior knowledge of the system's physical structure.

To this end, *hybrid ML models* have been developed. Among them, Physics-Informed Neural Network (PINN) [14] is a powerful approach that incorporates physical laws, typically expressed as Partial Differential Equations (PDEs), directly into the training process. PINNs achieve this by embedding the PDE residuals, the supervised loss on target variables, as well as initial and boundary conditions, into the loss function. Rather than learning solely from data, PINNs are guided by the governing equations of the system, enabling them to solve both forward and inverse problems.

In the classical literature on control theory, the task of system identification is sometimes formulated as an *observer design* problem. The goal is to reconstruct the internal states of a dynamical system from available measurements and its physical model. Unknown model parameters can be incorporated into the state vector. While linear observer design is well established [15], designing observers for nonlinear systems remains a challenging and active research area. In fact, a universal observer design framework for nonlinear systems does not exist [16].

Nonetheless, several effective methods have been proposed, including high-gain observers [17] and the Extended Kalman Filter (EKF) [18], both of which offer approximate solutions for nonlinear state estimation under certain assumptions. These classical observer approaches are primarily tailored to systems with smooth dynamics. However, many systems exhibit non-smooth behavior, that is, they combine continuous-time evolution with discrete events such as switching, logic-based transitions, or jumps. State estimation for such systems [19] introduces additional complexity, as observers must account for abrupt state jumps, mode switches, or hybrid transitions, in addition to tracking continuous dynamics. Traditional estimation techniques often fall short in handling these discontinuities and mixed behaviors. To address these challenges, specialized observer frameworks have been developed specifically for hybrid dynamical systems. For instance, Ackerson and Fu [20] are among the first to formulate the state estimation problem for switching systems. Later, Liu [21] introduce gain-switching observers designed for nonlinear systems with switching behaviors. More recently, Tran and Bernard [22] propose a hybrid Kalman-like observer tailored for general hybrid systems with linear time-varying dynamics and output maps, assuming that the jump times of the solutions are known exactly. In this dissertation, we focus on the Salted Kalman Filter (SKF) [23], an observer designed for hybrid models with unknown jump times.

Both approaches, hybrid ML models and observer theory, rely on a physical model and measured input data. Difficulties arise when the embedded physical model is non-smooth and the measured input data are only available at a low sampling frequency, which implies limited information content.

For that, in the second part of this thesis, we tackle a system identification problem involving a non-smooth friction model and sparse measurement data. We adapt and apply methods from both machine learning, through the use of PINNs, and observer theory, via the Salted Kalman Filter (SKF), to this problem and evaluate their performance.

We address this problem in the context of drilling, where the objective is to predict the downhole rotation speed and the friction coefficients between the drill string and the borehole, using low-frequency surface measurements. The physical model employed in both methods includes a nonlinearity representing Coulomb friction. Both approaches are compared and tested on field data from different drilling wells.

2.2 Drilling context

Rotary drilling is the most common method of drilling in oil & gas and geothermal applications. As depicted on Fig. 2.1, it consists in using a drilling device called bit, which usually destroys the rock in one of the two following ways: either by a shearing action (drag bits) or by indentation (roller-cones bits). The bit is linked to a rig located at the surface by a drill string comprising a series of pipes and a lower part called Bottom-Hole-Assembly (BHA). The drill bit has to be rotated and pushed against the rock formation. The force, or Weight-On-Bit (WOB), required for the bit cutters to engage the formation is obtained from the weight of the drill string. The rotation is applied either by the use of a rotary table or a top drive at the surface, and with assistance of a downhole motor in some applications. Downhole motors are particularly used for directional (lateral) wells and when the bit technology requires higher rotary speeds that are not technically possible with only rotation from the surface. They help avoiding drill string twist offs by allowing most of the rotational torque to be concentrated near the bit instead of lost through torque and drag on the drill string [24]. In addition, drilling fluids

and a circulation system are used to suspend and carry the cuttings and to cool down the drill bit and downhole equipment. The drilling fluid, also called mud, is delivered to the drill bit using pumps at high pressure and flow rate, and returns to the surface carrying the cuttings through the annulus between the drill string and wellbore. Other equipment can be used to improve the drilling efficiency such as shock subs to damp the vibrations caused while drilling hard formation, or drilling jars that are used to free the drill string by delivering an impact load when a stuck pipe is faced [25].

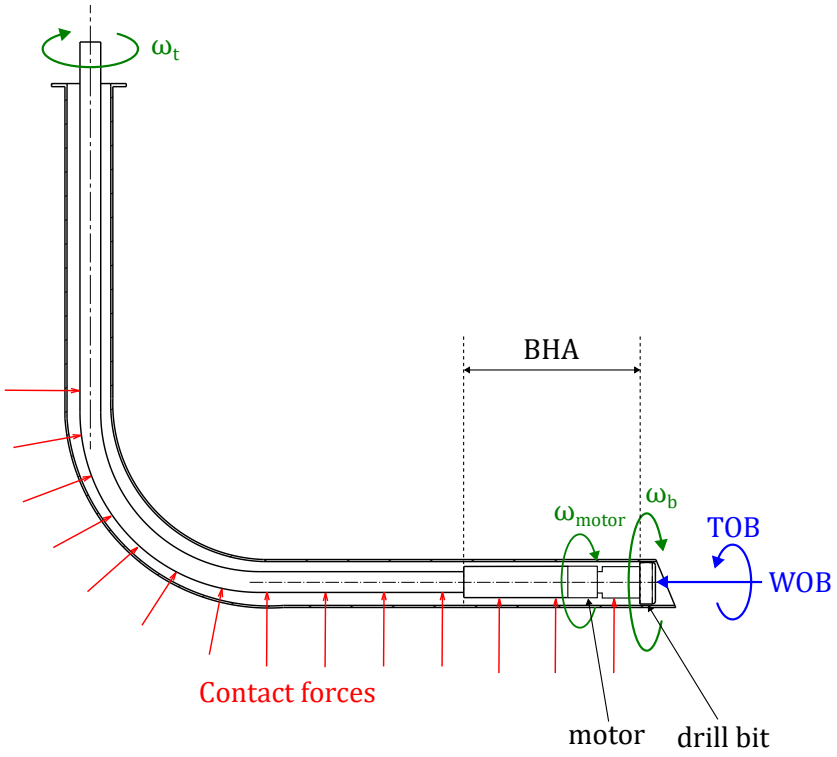


Figure 2.1: Simplified schematic representation of the drilling system: Bottom-Hole-Assembly (BHA) denotes the lower part of the drill string; ω_t, ω_{Motor}, and ω_b represent the rotational speeds at the surface, the downhole motor, and the bit, respectively; Torque-On-Bit (TOB) and Weight-On-Bit (WOB) correspond to the torque and the weight-on-bit, respectively.

During drilling, different types of vibrations, torsional, lateral, and axial, may occur independently or in combination. Detecting and controlling their severity is essential to ensure drilling efficiency and avoid equipment damage. To this end, advanced downhole sensors installed at the bit are used to measure critical parameters such as torque and rotational speed. These measurements can either be stored in onboard memory, retrieved at the end of the run[1], or transmitted to the surface in real-time via mud pulse telemetry. However, these sensors are expensive, and real-time data transmission is limited by the low bandwidth of mud telemetry. Consequently, only low-frequency data can be transmitted to the surface in real-time, whereas high-frequency data is typically used for post-drilling analysis.

[1] In drilling terminology, a run refers to the operation period during which a Bottom-Hole-Assembly (BHA) is deployed into the wellbore and remains downhole until it is pulled out for maintenance, replacement, or at the end of a drilling phase.

Given the high cost of downhole sensors and the limited real-time availability of downhole measurements, estimating and predicting downhole parameters, such as the severity of vibrations, using surface measurements has become an important area of research. Numerous studies have proposed data-driven approaches to address this challenge, particularly by leveraging surface drilling data collected from rig sensors, including torque, rotational speed, hook load, block position, and flowrate. This has led to a growing trend toward extracting value from surface data to optimize drilling performance and mitigate operational problems.

2.3 Outline of the thesis

This thesis is divided into two main parts. The first part focuses on improving the generalizability of ML model for predicting the severity of torsional downhole vibrations using sequences of 1-Hz surface measurements. The central challenge is to ensure that the trained model performs well on unseen drilling data from wells that differ from those used during training. To address this, we explore several strategies:

- Chapter 3 introduces torsional downhole vibrations, methods for detecting them using surface and downhole measurements, and the challenges of building ML models that generalize to unseen wells. It also describes the drilling datasets used in this thesis.
- Chapter 4 presents a baseline ML model, which serves as a reference point. We then evaluate four strategies to enhance generalizability: (i) data normalization, applied during or prior to training to reduce train–test distribution shifts; (ii) transfer learning (TL), where a small amount of target well data is used to fine-tune a trained model; (iii) data augmentation, using simulated or field data to address class imbalance in torsional vibration severity; and (iv) integration of physics-based features derived from physical models that encode information about well geometry, drill string composition, and trajectory. In addition, the baseline model is compared with a Transformer-based architecture.
- Chapter 5 explores more advanced strategies to improve model generalizability. Two main approaches are investigated: Adversarial Domain Generalization (ADG), which seeks to learn domain-invariant representations, and Invariant Risk Minimization (IRM), which aims to extract features that remain predictive across different data environments.

The second part of the thesis focuses on predicting downhole bit rotation speed and friction coefficients from 1-Hz surface measurements, assisted by a simplified physical model. This part includes the following contributions:

- Chapter 6 introduces the problem, reviews existing models and their limitations, and presents our simplified physical model together with the assumptions made. It also outlines our two proposed solutions.
- Chapter 7 details the first solution, a Physics-Informed Neural Network (PINN). In this method, the simplified physical model is embedded directly into the loss function of a neural network. The PINN is tested on both simulated and field data, and represents an offline solution.

- Chapter 8 presents the second solution, the Projected Salted Kalman Filter (PSKF), an extension of the Salted Kalman Filter (SKF). Like the PINN, it is validated on both simulated and field data. Unlike the PINN, the PSKF operates in an online fashion.
- Chapter 9 compares the two approaches, PINN and PSKF, across both simulated and field datasets. For simulations, a third state-of-the-art method is included for benchmarking. For field validation, data from four different wells are used.

2.4 Publications

Some of the results of this thesis have been published. They are listed below according to the corresponding chapters:

- Chapter 4:
  [26] Hana Yahia, Thomas Romary, Laurent Gerbaud, Bruno Figluizzi, Florent Di Meglio, Stephane Menand, and Mohamed Mahjoub. Combining machine-learning and physics-based models to mitigate stick-slip in real-time. In *SPE/IADC Drilling Conference and Exhibition*. SPE, 2024

  [27] Hana Yahia, Thomas Romary, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Real-time stick-slip mitigation using combined machine learning and physics based techniques. In *International Petroleum Technology Conference*. IPTC, 2024
- Chapter 5:
  [28] Hana YAHIA, Bruno Figliuzzi, Florent Di Meglio, Stephane Menand, Mohamed MAHJOUB, et al. Domain generalization for time series: Enhancing drilling regression models for stick-slip index prediction. *Transactions on Machine Learning Research*
- Chapter 7:
  [29] Hana Yahia, Florent DI Meglio, Bruno Figliuzzi, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Physics-informed neural network to estimate downhole torsional vibrations and friction coefficients. In *Abu Dhabi International Petroleum Exhibition and Conference*. SPE, 2025

Part I
Generalizable time series ML models for Stick-Slip Index prediction

In the first part of this thesis, we focus on applying machine learning to the drilling field, specifically to estimate the severity of downhole torsional vibrations using time series surface measurements. We rely on an indicator known as the Stick-Slip Index (SSI). Quantifying downhole vibrations from surface measurements has the potential to increase equipment durability, improve drilling efficiency and therefore decrease overall drilling costs.

However, in the drilling field, each well has its own specific characteristics,: trajectory, geological formation, hole diameter, tools used, etc. Therefore, training a machine learning model for SSI prediction using data from a limited set of wells with particular characteristics does not guarantee that the model will perform reliably on other wells with different conditions. Ensuring the reliability and generalizability of SSI prediction models across different well conditions remains a major challenge, as models trained on specific conditions often struggle to generalize to new, unseen wells. This difficulty arises from the complex and nonlinear nature of the stick-slip phenomenon, which depends on many interacting variables. Moreover, drilling data is often noisy, low-frequency, and unbalanced, making it hard for models to learn robust and generalizable patterns.

In this first part of the thesis, we explore and evaluate advanced strategies aimed at improving the generalization of machine learning models for SSI prediction using time series of surface measurements as input, across different wells. Chapter 3 introduces the stick-slip phenomenon, outlines the challenges in developing machine learning models for SSI prediction, and describes the drilling data collected and used in this study. In Chapter 4, a baseline model is presented, and several techniques are investigated to enhance its generalizability. These include various data normalization methods, Transfer Learning (TL), data augmentation, and the integration of physics-based features. Chapter 5 then explores more advanced strategies that aim to capture domain-invariant representations or learn generalizable features through Adversarial Domain Generalization (ADG) and Invariant Risk Minimization (IRM), respectively.

Chapter 3

Quantifying stick-slip oscillations

Résumé du chapitre

Ce chapitre commence par une introduction au phénomène de stick-slip, suivie d'une revue des approches d'apprentissage automatique existantes développées pour détecter les vibrations au fond du puits. Il présente également diverses méthodes utilisées pour évaluer la sévérité des vibrations torsionnelles et les compare. Enfin, le chapitre inclut une section d'exploration des données, présentant le jeu de données utilisé dans les deux chapitres suivants.

Chapter Summary

This chapter begins with an introduction to the stick-slip phenomenon, followed by a review of existing ML approaches developed for detecting downhole vibrations. It also presents various methods used to assess the severity of torsional vibrations and compares them. Finally, the chapter includes a data exploration section that covers the dataset used in the following two chapters.

3.1 Stick-slip oscillations

Stick-slip is a low-frequency torsional oscillatory phenomenon encountered during rotary drilling and characterized by an irregular downhole rotation speed. During stick-slip, the bit velocity (ω_b) fluctuates between a minimum value, which can be zero in the stick phase, and a maximum value that can reach several times the surface rotation speed (ω_t) in the slip phase [30]. This behavior is considered as one of the most damaging forms of vibration affecting the drill string. Sticking arises when the rotational energy transmitted through the drill string is insufficient to overcome the frictional torque along the string and the Torque-On-Bit (TOB), causing the bit to stop rotating for a duration typically ranging from 1 to 5 seconds. Meanwhile, the

top drive continues rotating at a constant speed, causing torsional energy to build up. Once this energy exceeds the resisting torque, it is released suddenly, leading to a spike in bit speed (slip phase) [31]. A key factor contributing to stick-slip oscillations is the velocity-dependent nature of friction, commonly described by the *Stribeck effect* [32]. According to this effect, friction is highest at low relative velocities and decreases rapidly as the relative sliding speed increases. In drilling, this means that when the bit momentarily sticks, torque builds up until it overcomes static friction, triggering a sudden slip as friction drops according to the Stribeck curve.

Another two possible causes of stick-slip are the regenerative effect[1], where the vibration history imprinted in the bottom-hole pattern amplifies the self-excited oscillations [33, 34], and the second reason the axial vibrations. Several studies have highlighted the coupling between axial and torsional modes [35, 34, 33], in particular, axial vibrations are often regarded as a precursor to torsional vibrations, as they can initiate or amplify stick-slip dynamics [34].

Stick-slip vibrations can occur both off-bottom, when the bit is not in contact with the rock, and on-bottom, due to either the interaction between the bit and the rock or the frictional contact between the drill string and the borehole [36]. These vibrations are typically observed under conditions of high Weight-On-Bit (WOB) and low angular velocity, as increasing the WOB or decreasing the rotational speed can trigger dynamic instabilities that lead to stick-slip. Among the contributing factors, lithology is recognized as one of the primary risk factors for stick-slip [37, 38]. Its influence can be further amplified by the aggressiveness and wear of the bit, both of which play a significant role in increasing the likelihood of stick-slip [36, 39]. These vibrations result in increased bit wear, reduced Rate Of Penetration (ROP), and premature fatigue failure of the drill string due to repeated torque loading and unloading cycles [30].

An example of severe stick-slip behavior is shown in Fig. 3.1, triggered by an increase in the WOB from 40 klbf to 50 klbf, and which leads to a decrease in ROP from 75 ft/h to 60 ft/h. The figure displays 1-Hz surface measurements of torque and rotation speed, alongside 1-Hz downhole measurements (obtained after the well is drilled), including WOB, ROP, and downhole rotation speed. The downhole bit rotation speed measurements are represented by three curves: maximum, minimum, and mean values of high-frequency downhole data over a 2.56-second interval (note: the 80-Hz high frequency data itself is not accessible). As WOB increases, severe stick-slip oscillations with a period of approximately 9 seconds emerge, causing the mean bit rotation speed to fluctuate between a minimum of 105 rpm (the downhole motor speed), and a maximum of 240 rpm. The surface rotation speed remains constant at 45 rpm as imposed by a controller. However, the surface torque fluctuates in response to the variations in bit rotation speed.

Notably, the minimum bit rotation speed registers zero values, regardless of the downhole motor's rotation speed. This indicates that the high-frequency bit rotation speed sensor detects instances where the bit velocity drops to zero even though the downhole motor continues rotating. This phenomenon, known as bit-sticking, can lead to either motor back-drive events, which occurs when the motor is powerful enough to drive the drill string backward, or motor microstalls, which are high frequency events compared to motor stall, and are not captured by surface data sampled at less than 10 Hz [40].

[1] The regenerative effect originates from the cutting action of the drill bit. Each blade of the vibrating bit leaves an undulating rock surface which, when cut by the subsequent blade, further excites the bit vibrations [33].



Figure 3.1: Example of a severe stick-slip sequence triggered by an increase in WOB from 40 klbf to 50 klbf. From top to bottom, the plots show the variations of: surface torque, downhole WOB, ROP, downhole motor rotation speed, total rotation speed (surface motor + downhole motor), and the maximum, minimum, and mean bit rotation speed. The stick-slip cycle has a period of approximately 9 seconds.

3.2 Stick-slip severity

To effectively diagnose and monitor stick-slip vibrations, it is essential to quantify their severity. The industry standard for this purpose is the Stick-Slip Index (SSI), which is based on downhole measurements of the bit's rotational speed, as described below. In addition, we propose two 'naive' indices derived solely from surface measurements and assess their relevance using field data.

3.2.1 Stick-Slip Index

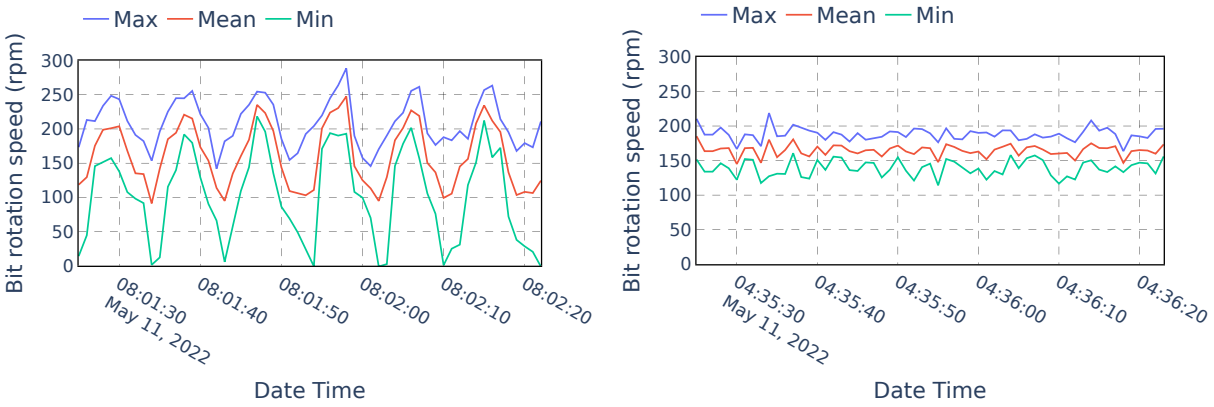
To quantify the severity of stick-slip, the industry standard metric, referred to us Stick-Slip Index (SSI), or sometimes Stick-Slip Severity Index, is calculated. The fluctuations in bit rotation speed (ω_b) over a specified time period, typically 30 seconds, are normalized as follows:

$$SSI = \frac{\omega_{b,\max} - \omega_{b,\min}}{2\ \omega_{b,avg}} \tag{3.1}$$

where $\omega_{b,\max}$, $\omega_{b,\min}$, and $\omega_{b,avg}$ are, respectively, the maximum, minimum, and mean values of ω_b computed over a given time period. As a result, a SSI value $\geq$ 0.8 indicates an extreme

stick-slip condition, characterized by bit rotation speed oscillating between zero and at least twice the surface rotation speed. Values of SSI between 0.5 and 0.8 correspond to severe torsional vibrations, while values between 0 and 0.5 indicate either weak torsional oscillations or relatively smooth drilling conditions [41]. Fig. 3.2 illustrates two field examples of bit rotation speed over 60-seconds intervals sampled at 1 Hz. The first sequence, characterized by a high SSI (0.89), indicates extreme stick-slip vibrations. In contrast, the second sequence reflects smooth drilling conditions, with a low SSI value of 0.31.

Computing the SSI requires a bottomhole velocity measurement. In the absence of a sensor, some companies rely on Torque Variation Index (TVI).



(a) Sequence 1: SSI = 0.89

(b) Sequence 2: SSI = 0.31

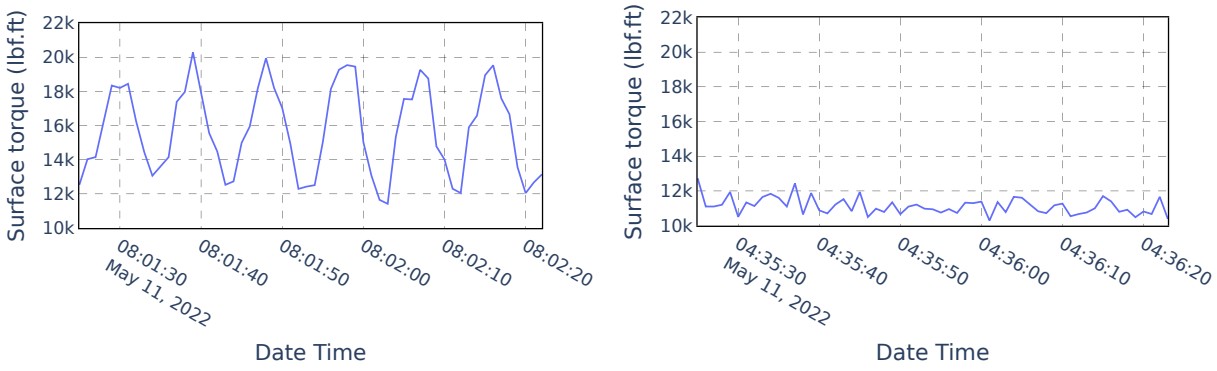
Figure 3.2: Examples of 60-second bit rotation speed sequences sampled at 1 Hz: (a) Sequence 1: Severe stick-slip with high SSI; (b) Sequence 2: Smooth drilling with low SSI.

3.2.2 Torque Variation Index

To analyze stick-slip severity using surface measurements only, a common method is to calculate the surface Torque (τ_t) Variation Index (TVI) over a specified time period, which measures the range of τ_t oscillations relative to its mean, similarly to the SSI:

$$TVI = \frac{\tau_{t,\max} - \tau_{t,\min}}{2\ \tau_{t,\text{avg}}} \tag{3.2}$$

where $\tau_{t,\max}$, $\tau_{t,\min}$, and $\tau_{t,\text{avg}}$ are, respectively, the maximum, minimum, and mean values of τ_t computed over a given time period. Fig. 3.3 presents the surface torque variation sequences, sampled at 1 Hz, associated with the 60-second bit rotation speed intervals shown in Fig 3.2. Stick-slip is generally considered to occur when the relative variations in surface torque exceed 30 % [38]. Accordingly, a TVI value below 0.1 indicates low amplitude torque fluctuations, values between 0.1 and 0.3 suggest moderate variation, and a TVI above 0.3 typically reflects high torque variability associated with severe stick-slip vibrations. It is important to note that the surface indicators may significantly underestimate the true severity of downhole vibrations, and fail to capture the full extent of the dynamic interactions occurring at the bit, more details are discussed in Section 3.2.4.



(a) Sequence 1: TVI = 0.28

(b) Sequence 2: TVI = 0.1

Figure 3.3: Surface torque variation sequences corresponding to the 60-second bit rotation speed intervals shown in Fig. 3.2, sampled at 1 Hz: (a) Sequence 1: Moderate stick-slip; (b) Sequence 2: Smooth drilling with low TVI. The maximum, minimum, and mean values of surface torque, required for calculating the TVI, are determined over a 60-second interval.

3.2.3 Power spectrum of the surface torque

Another approach to quantify the severity of stick-slip using surface measurements is to compute the power spectrum of surface torque variations [42]. For instance, Fig. 3.4 shows the normalized power spectrum with respect to total signal power, for the two sequences presented in Fig. 3.3. During severe stick-slip, the surface torque signal exhibits a dominant frequency component. This frequency is typically close to the estimated first natural torsional frequency of the drill string assembly [36, 43, 25]. By comparing the two normalized spectra, the sequence exhibiting severe stick-slip can be clearly distinguished from the one corresponding to smooth drilling conditions. To further support this distinction, the normalized power spectrum is summed over successive 0.1-Hz intervals. This binned integration allows the identification of dominant frequency ranges contributing to the overall torque fluctuations. The cumulative power in the i-th frequency bin is defined as:

$$P_i = \sum_{f \in [f_i, f_i + \Delta f)} P_{\text{norm}}(f), \quad \text{with } \Delta f = 0.1\,\text{Hz} \tag{3.3}$$

where P_{norm} is the normalized power spectrum, f_i is the starting frequency of bin i, and Δf is the bin width. The severity of stick-slip in each sequence is assessed by the maximum cumulative power across all frequency bins, $\max_i P_i$. In the first sequence, which exhibits pronounced stick-slip, the peak energy of 0.45 is concentrated in the frequency range 0.05–0.15 Hz. In contrast, the second sequence shows no stick-slip, with a lower maximum energy of 0.13 occurring between 0.35 and 0.45 Hz. However, we show, in the next section, that this indicator does not enable an accurate classification of stick-slip severity.

3.2.4 Comparison of proposed stick-slip severity quantification methods

To compare the proposed methods, Fig. 3.5 presents an example of a 1-Hz field data sequence. It includes surface measurements (surface torque, ROP and surface rotation speed) along with downhole data (WOB, motor and bit rotation speeds) also sampled at 1 Hz. For each 60-second window of downhole bit rotation speed, the SSI is computed using Eq. (3.1), and assigned to the

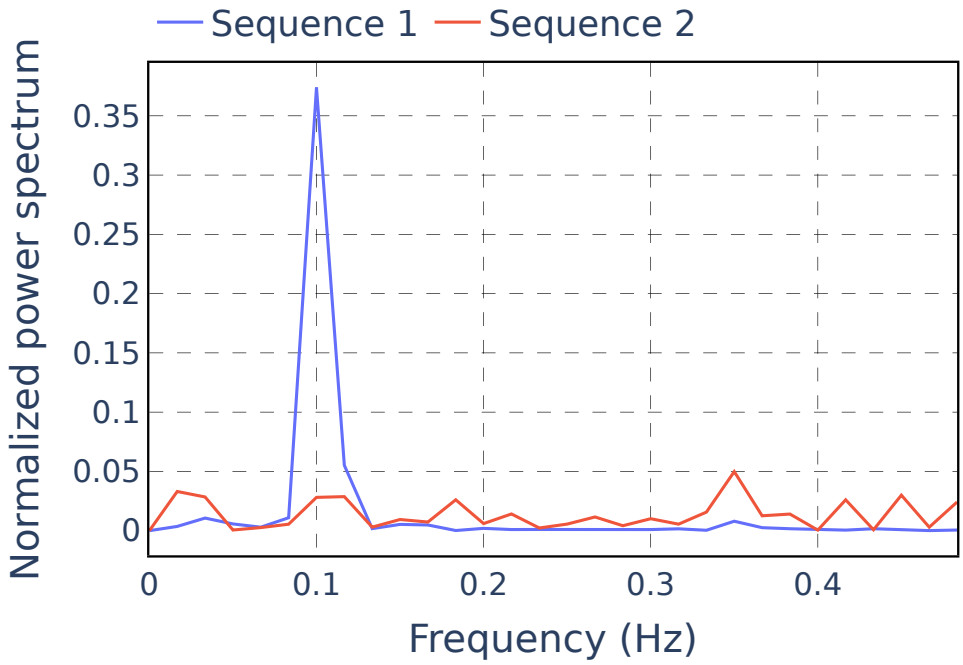


Figure 3.4: Power spectrum of the surface torque time variations of the two sequences presented in Fig. 3.3.

midpoint of the interval. Similarly, the TVI (Eq. (3.2)) is calculated for each 60-second segment of surface torque data. In the case of the power spectrum based method, the power spectrum of surface torque is computed over each 60-second window, and the maximum energy within a 0.1-Hz frequency band is extracted as the quantification metric. The computed SSI, TVI, and the power spectrum sum values are plotted in Fig. 3.5. As shown in Fig. 3.5, the three indicators: SSI, TVI, and the power spectrum sum are initially low, indicating the absence of stick-slip vibrations. As the sequence progresses, their values rise and fall in response to the severity of torsional vibrations. In general, the three indicators are correlated, as downhole vibrations tend to be transmitted to the surface. However, between 05 : 55 and 06 : 05, a discrepancy is observed: the SSI reaches a very high value (approximately 1), while both the TVI and the power spectrum sum decrease. This divergence can be explained by several factors:

- Downhole vibrations attenuation during propagation to surface due to viscous damping and friction between the drill string and the borehole.
- High-frequency oscillations: The SSI is derived from the maximum, minimum, and mean values of high frequency downhole rotation speed (the high-frequency measurements vary across wells, in our study, they are 80 Hz for most wells, with the maximum, minimum, and mean calculated over a 2.56-second interval). As a result, it captures high-frequency oscillatory behavior that is not detectable with 1-Hz surface measurements. Therefore, SSI can provide more information compared to the TVI or the sum of the power spectrum of surface torque [40].
- Surface torque aliasing due to under-sampling of high-frequency surface data. As noted earlier, no details are provided regarding the preprocessing of the high-frequency data.

To conduct a thorough analysis comparing TVI and the power spectrum of surface torque against the reference SSI, 250 000 sequences (each lasting 60 seconds) from a field well are used to calculate the SSI, TVI and the energy of surface torque, using 1-Hz downhole and surface data as explained previously. As SSI represents the truth value, the sequences are categorized into four classes based on SSI values: Low (SSI $\leq$ 0.3, with 117 030 sequences), Moderate (0.3 $<$ SSI $\leq$ 0.5, with 75 280 sequences), Severe (0.5 $<$ SSI $\leq$ 0.8, with 21 607 sequences), and Extreme (SSI $>$ 0.8, with 36 083 sequences). The results are presented in Fig. 3.6. As shown, the mean values of both TVI and surface torque energy slightly increase with the severity of downhole torsional vibrations (SSI), unlike SSI itself, where the mean of each class exhibits a clear jump from one class to the next. However, for example, a sequence with a surface torque energy of 0.2 can be classified into any of the four severity classes: Low, Moderate, Severe, or Extreme. Consequently, the severity of downhole vibrations cannot always be reliably inferred from only surface torque [25]. This limitation arises primarily from signal attenuation or the presence of high-frequency oscillations, which may not be captured at the surface using only the surface torque, as discussed earlier. Another drawback of the power spectrum method based on surface torque is the difficulty in defining a clear threshold to determine whether the observed variations are significant, unlike the SSI or TVI indicators, which have more interpretable ranges.

In conclusion, characterizing the severity of downhole torsional vibration using surface measurements remains a challenging task. In the next two chapters, generalizable data-driven machine learning models are proposed to predict the downhole SSI using 1-Hz surface measurements.

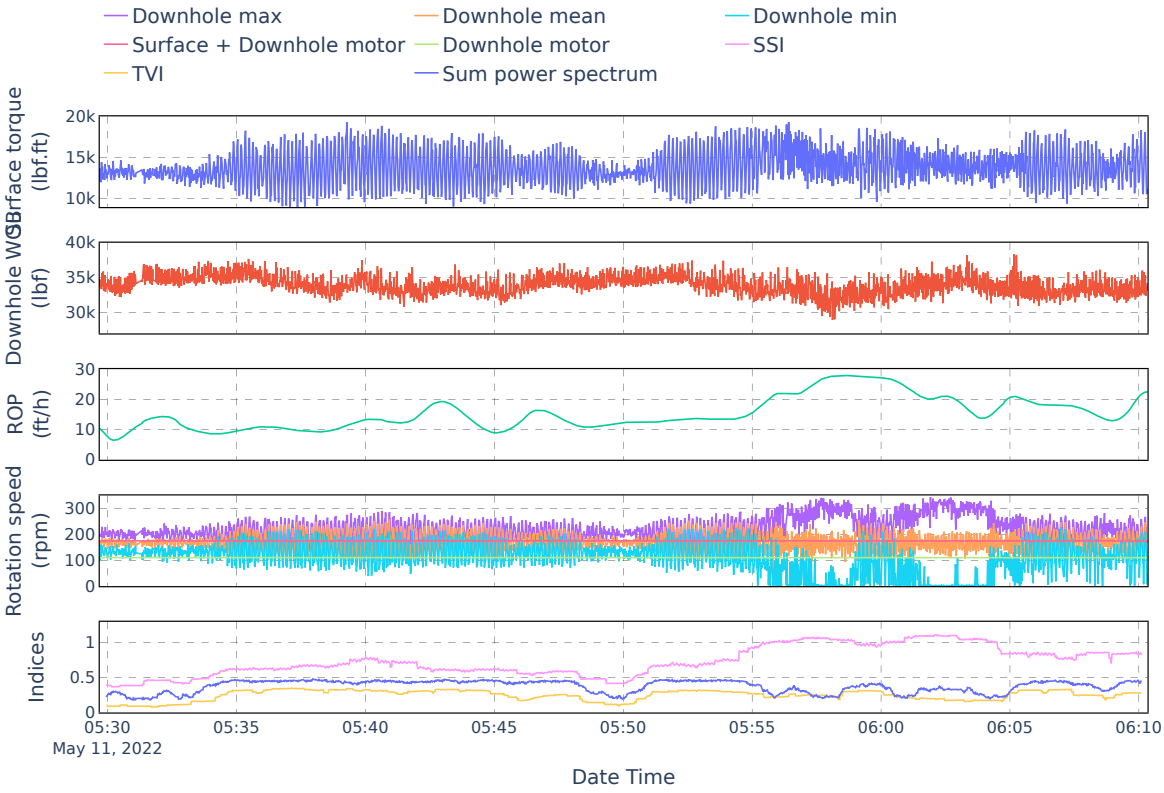
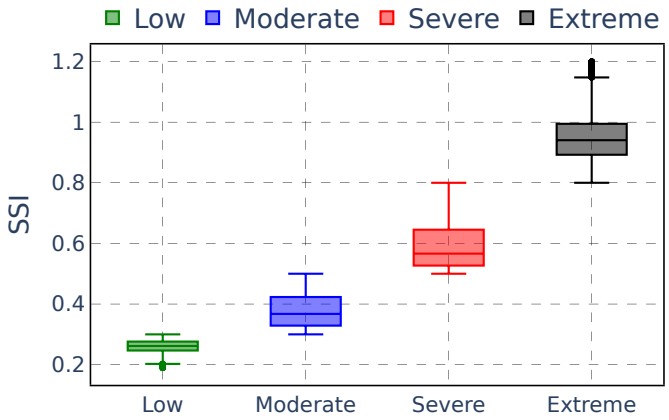
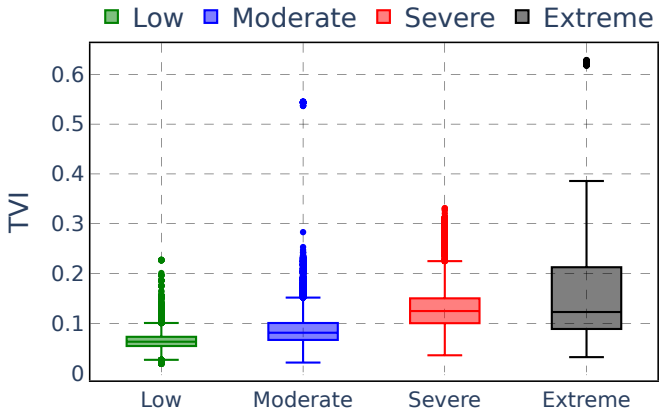


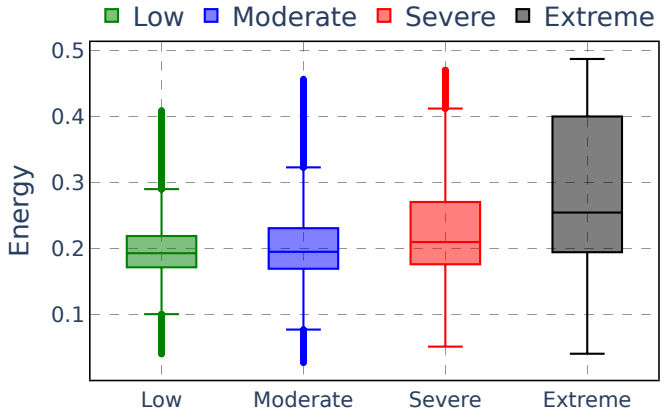
Figure 3.5: Example sequence showing 1-Hz field surface data: Surface torque, ROP, and surface rotation speed, as well as 1-Hz downhole data: WOB, motor and bit rotation speeds. The figure also displays the stick-slip indicators: SSI, TVI, and the surface torque power spectrum sum.



(a) SSI



(b) TVI



(c) Surface torque spectrum energy

Figure 3.6: Comparison of the three proposed indicators for stick-slip detection. A total of 250 000 sequences, each lasting 60 seconds, from a field well are used to calculate the SSI, TVI, and the surface torque spectrum energy. The categories for SSI are defined as follows: Low (SSI $\leq$ 0.3), Moderate (0.3 < SSI $\leq$ 0.5), Severe (0.5 < SSI $\leq$ 0.8), and Extreme (SSI > 0.8).

3.3 Review of machine learning studies on stick-slip detection

In recent years, several solutions for detection and mitigation of stick-slip have been proposed. These solutions can be classified into three categories: drill string and BHA design optimization [44, 45], bit design optimization [46], and drilling parameters optimization [47]. Analyzing drilling parameters and dynamics can contribute to reduce the occurrence and severity of stick-slip [48]. Physical models can be employed to study and characterize this phenomenon usually under numerous simplifying assumptions. This is typically still computationally expensive and requires a detailed drill string configuration, due to the complexity of capturing the interactions of the drill string with both the borehole and the drilling fluid. With the development of machine learning models and data analysis techniques, many authors use data-driven models to detect downhole events automatically and reduce the need for frequent inspections by the crews. Zha and Pham [49] develop a binary classification model to detect stick-slip using surface data measured at 100 Hz. Okoli et al. [50] propose five ML models for classifying the downhole axial and lateral vibrations using surface drilling data. Ignova et al. [51] develop a classification model using K-Means cluster analysis algorithm and Principal Component Analysis (PCA) pattern recognition technique. The model is trained by 2048 measurements data set for the classification of good or damaged drilling using high-frequency shock data for the model training process. Wiktorski et al. [52] utilize Support Vector Machine (SVM) for developing two models: one for detecting stick–slip and the other for lateral vibrations by training the models using surface measurements as input. Elahifar and Hosseini [53] introduce a machine learning approach combining model agnostic regression techniques with Bayesian Optimized Extra Trees (BO_ET) to predict stick-slip events in real-time using surface and downhole data as inputs. Baumgartner and van Oort [54] classify downhole high-frequency acceleration data with sampling rates of 400 Hz as stick-slip/no stick-slip or whirl/no whirl. Hegde et al. [55] compare a list of machine learning classification algorithms such as logistic regression, SVM, and random forests to classify stick-slip severity of surface measurements. They use data from vertical wells with 12 different geological formations. A separate model is built for each formation because the main attributes such as lithology, BHA, drill bit, and drilling fluid are assumed to remain the same during drilling or changed minimally. If these attributes were changed, then a new model would have to be retrained on data collected while drilling with the changed parameters.

As an alternative, some researchers attempt to incorporate physics-based principles into their model during the training process. Generating simulated data is one way to combine data with physics. Millan et al. [56] train two classification models using historical offset well data and simulated well data to detect and characterize downhole lateral shocks and stick-slip vibration mode from real high-frequency surface measurements. The frequency spectrum of the data is calculated and given as input to the models. The surface torque signal is in particular recorded at 10 Hz and 25 Hz to detect stick-slip and at 50 Hz for whirling. Adding physical features is another method to use physics, such as calculating critical rotation speeds and combining it with surface data to optimize ROP, as exposed by Behounek et al. [57]. An application of Physics-Informed Machine Learning (PIML) is proposed in [48] to predict the stick-slip class for the next stand, given historical well logs in the same geographical region and previous stands in the well. In their work, a physics based model is used to calculate the SSI, which is added as another feature in the data-based model along with the surface measurements to predict the final stick-slip class.

The effort to detect downhole vibrations using ML models is however facing four key challenges that are not addressed by the literature:

(i) Generalization of the ML models: The developed ML models are used only for wells in the same specific drilling field as the training data. When getting out of the distribution of the training data, there is an elevated risk of observing inconsistencies in the predictions [48].

(ii) Prediction of the SSI: The developed ML models are binary classification models, only predicting if the event exists or not, but lack information about the severity.

(iii) Use of downhole data as input features: Advanced downhole sensors can detect and analyze stick-slip vibrations in detail. However, they are expensive, and real-time data transmission is limited by mud telemetry bandwidth. Only low-frequency data can be sent to the surface in real-time, while high-frequency data is only used for post-drilling analysis.

(iv) Sampling frequency of surface data: High-frequency surface data is rarely available. In most industrial setups, the sampling rate is around 1 Hz. Although higher-frequency measurements are possible, data storage and transmission still typically occur at about 1 Hz.

In the next two chapters, we propose to address these issues by designing a machine learning model to estimate the SSI from 1-Hz surface measurements. To this end, we leverage the data available at DrillScan, which we now describe.

3.4 Presentation of the dataset

The data has been collected on a number of distinct wells, from different well sites in different geographical locations, as outlined in Table 3.1. Fig. 3.7 presents example trajectories of the wells used in this study. Some of these wells are from the same field and share similar characteristics. To assess the generalizability of the trained model, it is ensured that wells from the same field are consistently used either as test data or training data, but never both.

3.4.1 Data cleaning

For each well described in Table 3.1, 1-Hz time series of surface and downhole industrial drilling data are collected by the operator, contractor, or drilling service companies. However, no details are provided regarding the preprocessing steps applied to the original high frequency signals such as outlier removal, smoothing, or noise filtering that may have been used to produce the 1-Hz data. The surface data, which serves as input to the regression model includes surface torque, surface estimated WOB, ROP, flow rate [2], and total rotation speed variations, which is the sum of the surface rotation speed and downhole motor rotation speed estimated from surface measurements[3]. The surface estimated WOB is derived from the difference between off-bottom and on-bottom hookload measurements, whereas the ROP is inferred from

[2]Flow rate in drilling operations typically refers to the volume of drilling fluid (mud) pumped through the drill string per unit of time

[3]To approximate the downhole motor velocity, a Gaussian Process (GP) is used to learn the nonlinearity between the differential pressure ΔP, flow rate, and downhole motor rotation speed [58]

Well	Field number	Type of wellbore trajectory	Well length (ft)	Bit diameter (inch)	Drill pipe diameter (inch)	BHA length (ft)	Available drilling data duration (day)
1	1	Lateral	18 120	8 3/4	5	167.9	1.53
2	1	Lateral	18 204	8 1/2	5	166.9	1.06
3	1	Lateral	17 921	8 3/4	5	136.94	0.85
4	2	Lateral	27 723	8 3/4	5 1/2	101.03	1.30
5	3	Lateral	19 355	6 3/4	5 1/2	128.99	1.88
6	4	Lateral	29 077	8 3/4	5	130.52	3.30
7	5	Lateral	28 645	8 3/4	5 1/2	97.36	6.16
8	6	Vertical	20 619	8 3/4	5	110.31	1.08
9	6	Vertical	20 679	8 3/4	5	111.23	1.70

Table 3.1: Summary of well characteristics, where vertical wells are drilled straight down from the surface into the target formation or reservoir, and lateral wells are drilled vertically to a certain depth, then deviates horizontally within the target.

block position measurements, that track how much pipe has been run into the well. These features are chosen based on their relevance to the stick-slip problem and on their availability on the majority of wells. Downhole data includes only bit rotation speed, used to label surface data. A total of 5 819 166 one-second data points are collected across all wells.

To synchronize downhole data with surface data, a clock shift is determined for each well and applied to the downhole data, relying on cross-correlation. Following this alignment, an operation detection process is applied to extract only rotary drilling data[4]. This is done using process knowledge-based criteria such as surface rotation speed exceeding 10 rpm and bit depth matching the hole depth. After cleaning, 1 633 596 data points are retained. Surface data is then labeled: the SSI is computed using Eq. (3.1) and assigned to the midpoint of the interval. Table. 3.2 presents the descriptive statistics for the input features and the target SSI variable.

	Min	Mean	Max	std	25%	50%	75%
Surface torque (klbf.ft)	0	13.73	29.97	4.87	10.57	12.81	16.87
Surface WOB (klbf)	0	26.62	90.10	17.49	11.60	25.72	39.90
ROP (ft/h)	0.59	58.15	200.73	50.27	16.06	38.53	98.97
Flow rate (gpm)	214.00	487.46	724.83	94.26	497.00	526.02	550.34
Total rotation speed (rpm)	75.66	191.13	320.68	34.00	170.96	183.75	211.13
SSI	0	0.35	1.9	0.21	0.21	0.3	0.45

Table 3.2: Summary statistics of input and target characteristics. The total number of data points is 1 633 596, where each data point corresponds to one timestamped measurement across the variables.

[4]Refers to data from rotary drilling operations, the conventional drilling method in which the entire drill string is rotated from the surface using equipment such as a top drive or rotary table. The alternative is sliding drilling, typically used during directional drilling, where the drill string remains stationary while a downhole motor rotates the bit.

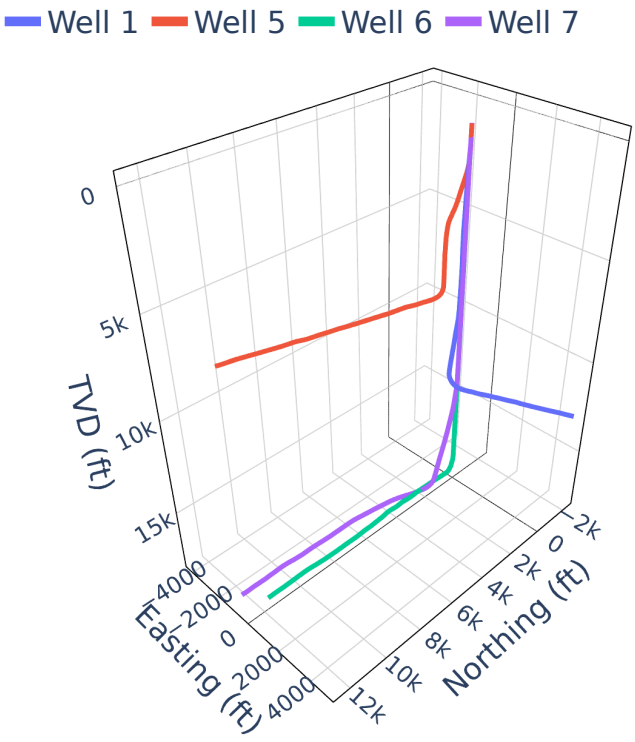


Figure 3.7: Example trajectories of the four wells 1, 5, 6, and 7 described in Table 3.1.

3.4.2 Data analysis and visualization

To help understand the relationship between input features and the output, we propose to rely on simple data analysis and visualization techniques. The strength of the association between input and output variables is quantified using Pearson's correlation coefficients and Spearman's rank correlation coefficients, calculated according to Eq.(3.4) and Eq.(3.5), respectively.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.4}$$

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{3.5}$$

In Eq.(3.4), n denotes the sample size, corresponding in our case to the total number of data points across all wells. The variables x and y represent the sample values, while $\bar{x}$ and $\bar{y}$ denote their respective means. In Eq.(3.5), d_i corresponds to the difference between the ranks[5] of the i-th observation of two groups x and y. Both Pearson and Spearman coefficients indicate stronger

[5]Rank indicates the position of an observation within its respective sorted list.

correlation when their values are close to 1 or -1, whereas values near zero suggest independence between the input parameters. The heatmaps in Fig. 3.8 illustrate the correlations among the input features across the entire dataset from all wells. Some correlations are observed between input features, such as negative correlation between ROP and surface WOB, and positive correlation between ROP and total rotation speed, highlighting the interdependence of drilling parameters.

3.4.3 Well-to-well statistical comparison

In this section, a closer look is taken at the data of the wells listed in Table 3.1. Since our goal is to create a generalizable SSI prediction model that works well across different wells, it is important to confirm that the data from each well are sufficiently different and do not come from the same distribution. Although Table 3.1 already points out some differences like bit diameter and well trajectory, this analysis focuses on comparing the actual feature distributions across wells.

To compare data distributions across the different wells, the Kolmogorov-Smirnov (KS) test is used. It is a non-parametric statistical test that evaluates whether two independent samples are drawn from the same distribution, without making any assumptions about the shape of that distribution. The test calculates the Empirical Cumulative Distribution Functionss (ECDFs) of the two samples and quantifies the maximum difference between them. An ECDF shows, for any given value, the proportion of data points that are less than or equal to that value. It is a simple way to visualize and compare data distributions. For a sample of n data points $x_1, x_2, \ldots, x_n$, the ECDF at a value x is defined as:

$$\text{ECDF}(x) = \frac{\text{Number of data points } \leq x}{n} \tag{3.6}$$

The KS statistic is given by the largest vertical distance between the two ECDFs:

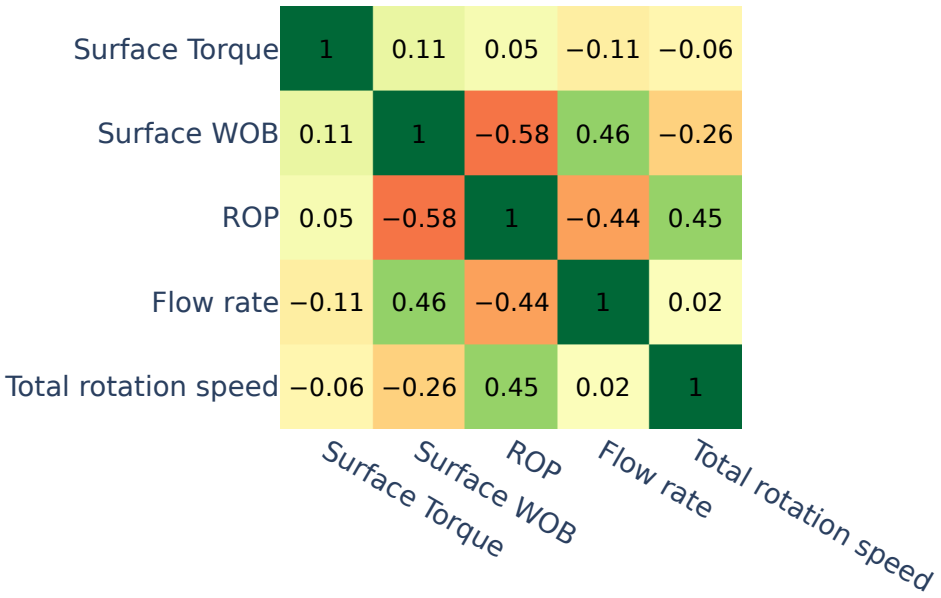
$$\text{KS} = \sup_x |\text{ECDF}_1(x) - \text{ECDF}_2(x)| \tag{3.7}$$

where $\text{ECDF}_1(x)$ and $\text{ECDF}_2(x)$ denote the ECDFs of the two samples. A higher KS value indicates a greater difference between the distributions. This test is especially useful for checking whether the data from different wells are distinct enough to justify training generalizable models.

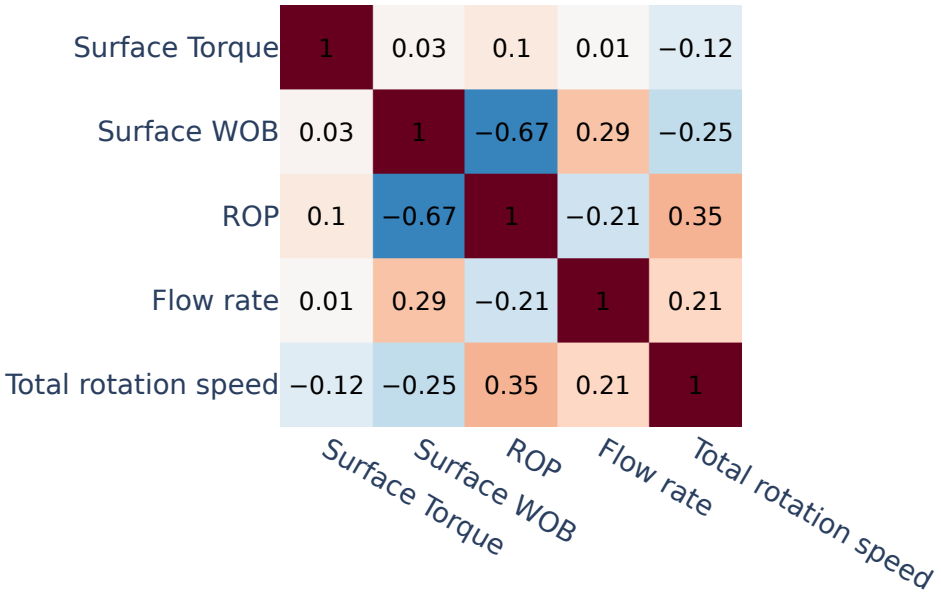
Fig. 3.9 presents KS-based heatmaps for surface torque, surface estimated WOB, ROP, total rotation speed, and SSI, comparing all wells listed in Table 3.1. To illustrate the distribution differences more concretely, Fig. 3.10 displays histograms of the same variables for three example wells.

Overall, most wells show noticeable differences, as reflected by high KS values. For example, the surface torque distributions between wells 5 and 8, and the surface estimated WOB distributions between wells 2 and 6, differ significantly. Wells from the same field are not systematically similar according to this test. For example, Wells 8 and 9 appear similar, whereas Wells 1 and 3 show distinct distribution patterns.

To deepen the analysis, since the main objective of the first part of this dissertation is to train a generalizable SSI prediction model using surface measurements, Fig. 3.11 presents the correlation calculated using Pearson coefficient, between each of the input features and the SSI across some test wells. As can be seen, the correlations differ from one well to another. For



(a) Pearson correlation

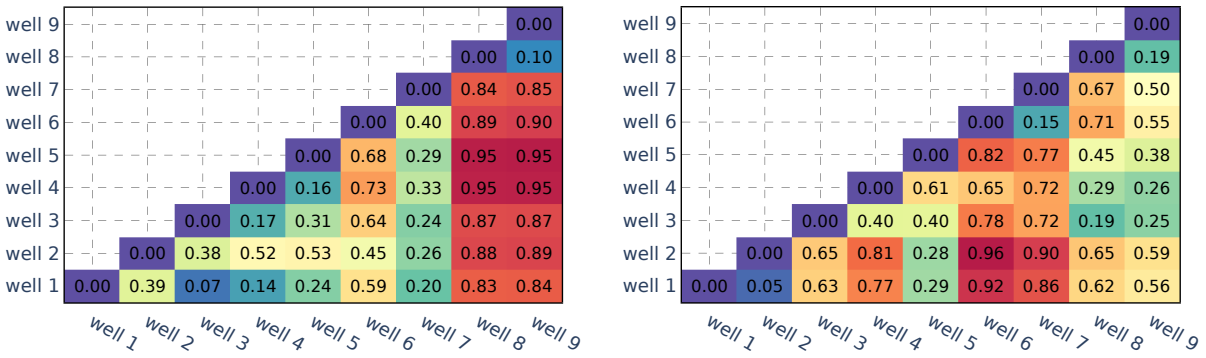


(b) Spearman correlation

Figure 3.8: Heatmaps illustrating the relationships between the input features across the entire dataset from all wells: (a) Pearson correlation, (b) Spearman correlation.

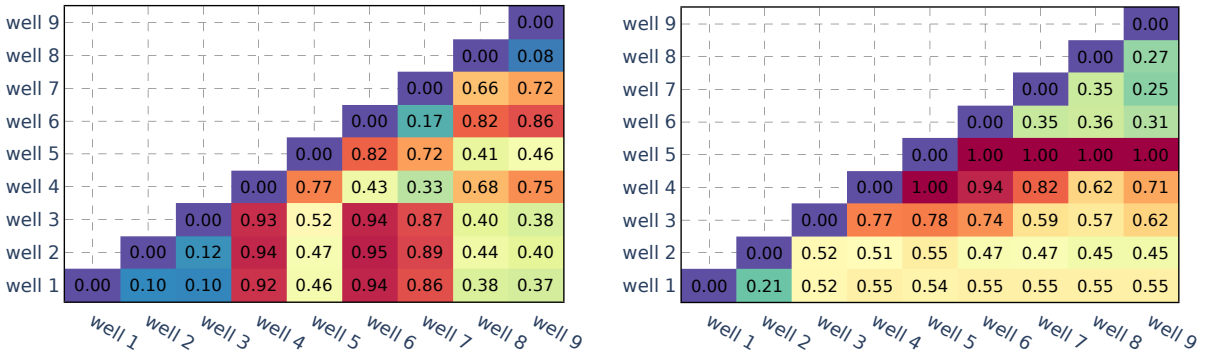
example, in Well 4, surface torque is identified as the feature most negatively correlated with SSI, whereas in Well 6 it is identified as the least correlated.

In conclusion, although the number of wells is relatively small, the statistical and visual



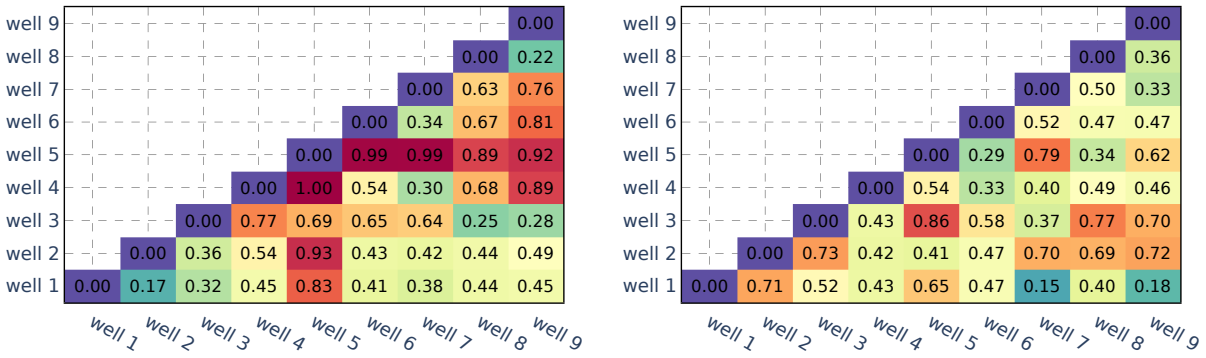
(a) Surface torque

(b) Surface WOB



(c) ROP

(d) Flow rate

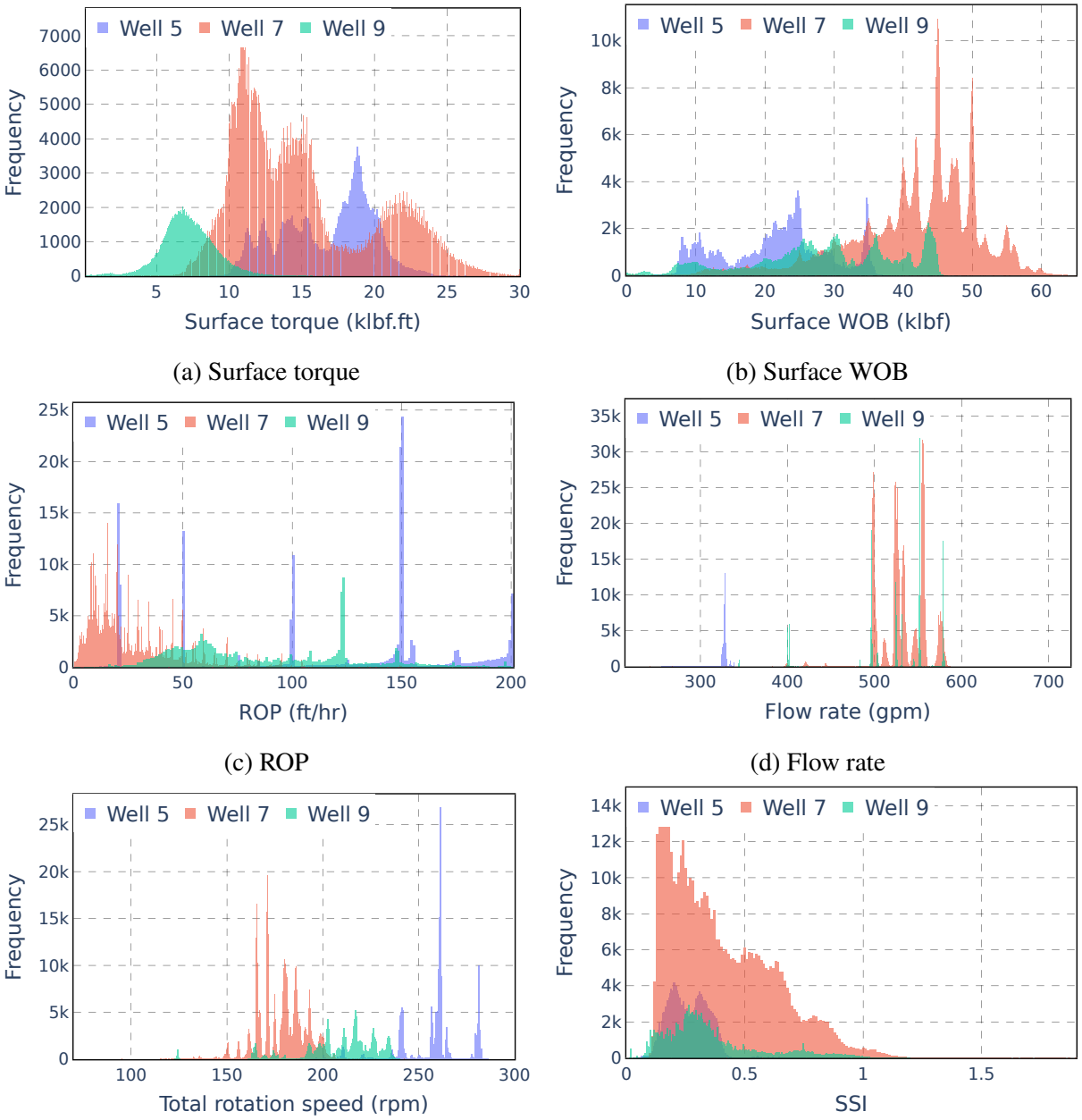


(e) Total rotation speed

(f) SSI

Figure 3.9: Kolmogorov–Smirnov heatmaps of: (a) Surface torque; (b) Surface WOB; (c) ROP; (d) Flow rate; (e) Total rotation speed; and (f) SSI for all wells described in Table 3.1.

examinations including feature distributions, Kolmogorov–Smirnov test, and correlation analyses, demonstrate substantial variability across wells in terms of surface torque, surface estimated WOB, ROP, flow rate, and total rotation speed. This variability presents both a challenge and an opportunity: while it increases the complexity of model training, it also provides a valuable testing ground for evaluating the generalizability of machine learning models.



(a) Surface torque

(b) Surface WOB

(c) ROP

(d) Flow rate

(e) Total rotation speed

(f) SSI

Figure 3.10: Histograms of: (a) Surface torque; (b) Surface WOB; (c) ROP; (d) Flow rate; (e) Total rotation speed; and (f) SSI for wells 5, 7, and 9, described in Table 3.1.

3.4.4 Class imbalance

Training machine learning models on imbalanced datasets remains a challenging task. In our case, low torsional vibrations occurrences outweighs severe and extreme sticking events (see Fig. 3.12). This imbalance can negatively impact the model's ability to accurately estimate the SSI of high torsional vibration sequences. To address this, one approach is to increase the number of sequences representing severe stick-slip, or alternatively, to reduce the number of sequences representing the more prevalent low SSI sequences. In our case, this issue arises in Chapter 4, during the application of transfer learning (TL), where data augmentation is em-

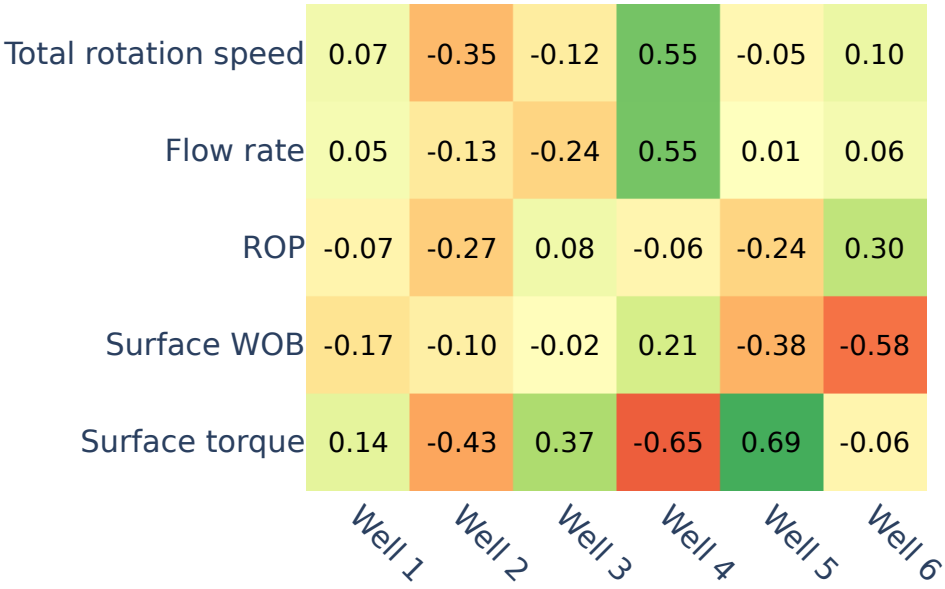


Figure 3.11: Heatmap of Pearson correlation coefficients between each input feature and the SSI across some test wells.

ployed by adding either synthetic or field derived sequences to enhance model performance. Further details are provided in Section 4.4.

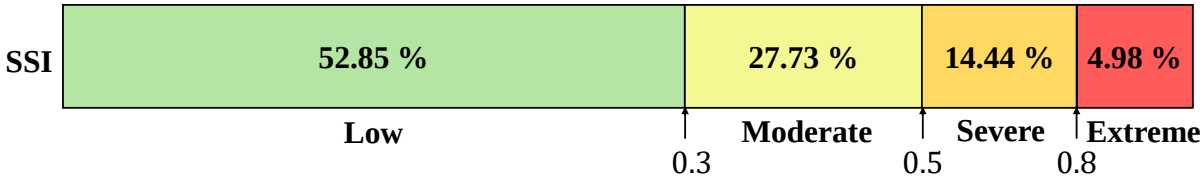


Figure 3.12: Distribution of SSI values across all well data described in Table 3.1.

3.5 Contribution of Part I

In the first part of this manuscript, the main contribution is the introduction of a generalizable ML model to predict a continuous stick-slip severity index using surface measurements. The model is trained on 1-Hz field data, described in Section 3.4, and is designed with a focus on generalization across varying drilling conditions. To achieve this, several strategies are evaluated:

- Testing various data normalization methods (Chapter 4): Different scaling and standardization methods are assessed to improve model generalazability.

- Incorporating physically meaningful features alongside surface measurements (Chapter 4): Physically meaningful features derived from domain knowledge, are integrated with raw surface measurements to enhance model interpretability and predictive power.
- Applying a transfer learning (TL) technique (Chapters 4 and 5): The trained model is allowed to leverage knowledge from previously seen conditions and adapt more effectively to new and unseen domains.
- Augmenting the data with additional sequences (Chapter 4): Addition of synthetic or field derived sequences, to increase data diversity and improve generalization.
- Exploring domain generalization methods in ML (Chapter 5): Advanced ML methods are employed to improve the model's ability to generalize across different operational contexts and field scenarios.

Chapter 4

Stick-Slip Index regression and generalizability

Résumé du chapitre

Ce chapitre aborde les limites de la généralisabilité des modèles d'apprentissage automatique dans le cadre de la prédiction de l'SSI. Il présente un modèle de régression de référence et explore plusieurs stratégies visant à améliorer sa fiabilité et sa généralisabilité à l'aide de données de forage temporelles. Ces stratégies incluent différentes méthodes de normalisation des données, l'apprentissage par transfert, l'augmentation de données et l'intégration de caractéristiques basées sur la physique. De plus, le chapitre examine une approche basée sur les Transformers, proposée comme une méthode efficace pour capturer les dépendances sur de longues séquences, et compare ses performances à celles du modèle de référence. Certaines parties de ce travail ont été publiées dans les conférences *SPE/IADC Drilling Conference and Exhibition* [26] et *International Petroleum Technology Conference (IPTC)* [27].

Chapter Summary

This chapter addresses the limitations of machine learning model generalizability in the context of SSI prediction. It introduces a baseline regression model, and explores several strategies to improve its reliability and generalizability using time-series drilling data. These strategies include different data normalization methods, transfer learning, data augmentation, and the integration of physics-based features. Furthermore, the chapter examines a Transformer-based approach, which has been proposed as an effective method for capturing long-sequence dependencies, and compares its performance against the baseline. Parts of this work are published in the conferences *SPE/IADC Drilling Conference and Exhibition* [26] and *International Petroleum Technology Conference (IPTC)* [27].

4.1 Baseline model for SSI prediction

To determine the baseline model, the existing literature on predicting downhole vibrations remains relatively limited, with no clear consensus on the optimal architecture for addressing this problem. Recent studies employ relatively straightforward machine learning techniques. For instance, Saadeldin et al. [59] utilize methods such as SVMs, radial basis functions, and functional networks to predict downhole drill string vibrations (lateral, axial, and torsional vibrations), while Elahifar and Hosseini [53] relies on decision trees for stick-slip prediction (see Section 3.3). To the best of our knowledge, our approach, first published in [26], is among the first to leverage deep learning techniques for SSI prediction using time series as input.

4.1.1 Long Short-Term Memory

Long Short-Term Memory (LSTM) [10] is a type of deep neural network designed to handle time series and sequential data. It is particularly efficient at capturing long-term dependencies across time steps, as discussed in [60]. Functioning similarly to a Recurrent Neural Network (RNN), an LSTM feeds the output of one time step into the next. What sets LSTM apart is its internal memory state, which allows it to store and retrieve information over extended sequences, hence mitigating the vanishing gradient problem commonly encountered in traditional RNN. As shown in Fig. 4.1, each prediction depends on the current input, previous output, and internal state.

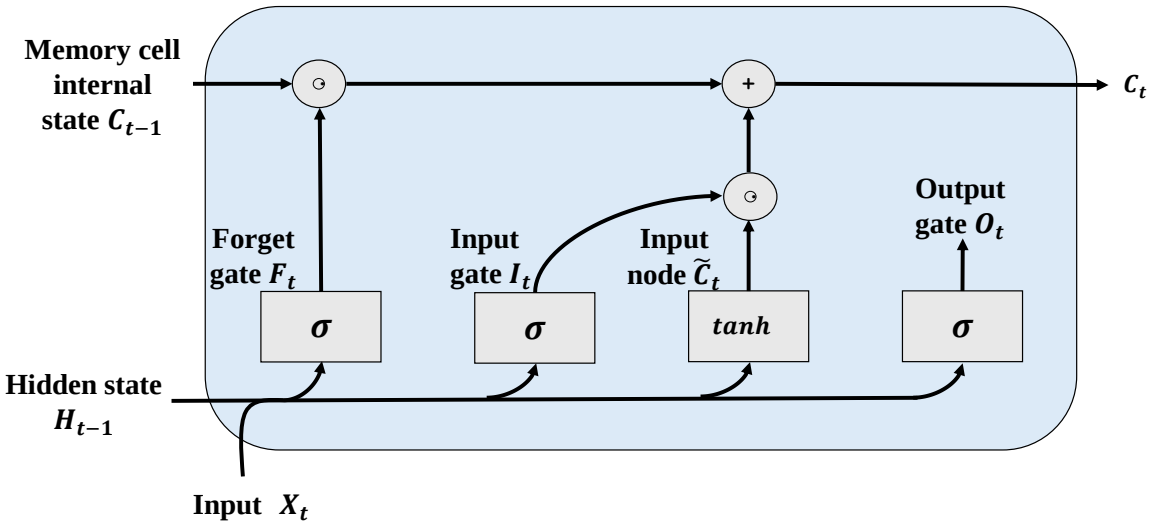


Figure 4.1: Architecture of a LSTM Unit (Source).

4.1.2 Baseline model architecture

The baseline model processes 60-second sequences of 1-Hz surface data as input (surface torque, surface estimated WOB, ROP, flow rate, and total rotation speed), and predicts one

SSI for each sequence. Therefore, the input X_t belongs to $\mathbb{R}^{60\times5}$. Since the input is time series data, the baseline architecture must be capable of modeling temporal dependencies. The LSTM method, described above, is selected for this purpose. The model architecture consists of five LSTM layers, each with 64 neurons, followed by a dense output layer with a single neuron to predict the SSI. Regularization (bias, kernel, and recurrent) is applied uniformly across all layers with a coefficient of 10^{-4}.

In the next sections, we test additional strategies for boosting the generalization performance of the baseline model by testing four different techniques: data normalization, transfer learning, data augmentation, and physics integration.

4.2 Data normalization

Data normalization is an important pre-processing step used whenever the features of the dataset have different ranges [61]. It helps the model to give equal contribution to each feature and to maintain the overall data distribution. The main objective of the normalization is to ease the generalization of the model from one field to a completely different one. In this section, four different types of normalization are tested: feature-by-feature normalization, sequence normalization, Deep Adaptive Input Normalization (DAIN), and Layer Normalization (LN) [26].

4.2.1 Feature-by-feature normalization

Feature-by-feature normalization involves normalizing each feature independently. For each feature, the mean μ and the standard deviation σ are calculated using only data from the training set. These values are then used to normalize both the testing and training data with the standardization scaling method, which consists in centering each variable at zero and standardizing the variance at one as follows:

$$X' = \frac{X-\mu}{\sigma} \tag{4.1}$$

where X and X' represent, respectively, a feature before and after the normalization.

4.2.2 Sequence normalization

Sequence normalization consists of the same steps as feature-by-feature normalization, but the scaling is applied separately to each 60-second sequence, rather than across the entire training data. For each sequence of length L, with $L = 60$ in our case, the mean $\mu_{[0,L]}$ and standard deviation $\sigma_{[0,L]}$ are calculated for each feature, and used to normalize the corresponding sequence:

$$X' = \frac{X-\mu_{[0,L]}}{\sigma_{[0,L]}}. \tag{4.2}$$

4.2.3 Deep Adaptive Input Normalization

DAIN [62] is a deep learning layer that learns a normalization scheme according to data distribution. This method consists of the following steps: First, for each sequence of length L, the mean μ^i and standard deviation σ^i for each feature i are computed:

$$\mu^i = \frac{1}{L}\sum_{j=1}^{L} x_j^i \tag{4.3}$$

$$\sigma^i = \sqrt{\frac{1}{L}\sum_{j=1}^{L}(x_j^i - \mu^i)^2} \tag{4.4}$$

Then, a shifting operator α^i is defined using a linear transformation of μ^i such that:

$$\alpha^i = W_a\, \mu^i \tag{4.5}$$

where $W_a \in \mathbb{R}^{d\times d}$ is the weight matrix of the first neural layer, with d representing the number of input features of the model, which is responsible for shifting the measurements across each dimension. Similarly, an operator β^i is defined as a linear transformation of σ^i such that:

$$\beta^i = W_b\, \sigma^i \tag{4.6}$$

where $W_b \in \mathbb{R}^{d\times d}$ is the weight matrix of the scaling layer. W_a and W_b are trained in the same time with the regression model. Then, each sequence is normalized using the new values of mean and standard deviation as follows:

$$X' = \frac{X - \alpha}{\beta}. \tag{4.7}$$

Finally, the new sequence is passed through a learnable gating mechanism that determines how much of each feature should be retained. To incorporate DAIN for normalization in the baseline model (see Section 4.1.2), we simply add a DAIN layer before the LSTM layers of the baseline architecture.

4.2.4 Layer Normalization

LN consists in estimating the normalization statistics from the summed inputs to the neurons within a hidden layer [63]. Each row of the sequence is centered and scaled by subtracting its mean and dividing by its standard deviation. Multiply the normalized sequence Y by a learnable parameter Γ and then add a bias Δ, which are also learned using the training data (Eq. (4.8)).

$$Y' = \Gamma\, Y + \Delta \tag{4.8}$$

To integrate LN into the baseline model architecture, each LSTM layer is followed by an LN layer. A comparison of the performance of these normalization methods is provided in Section 4.6.1.

4.3 Transfer learning

4.3.1 Definition

TL [64] focuses on transferring the knowledge from one domain (called source domain) to another (target domain) to improve the learning performance or minimize the number of labeled

examples required in the target domain. As explained in [65, 66], it works by reusing learned representations and features from a pretrained model and adapting them to a new task. This enables the model to work effectively even with limited data from the test domain. The re-training can be performed on all or parts of the source model. Full model fine-tuning involves re-learning the weights of all the layers; and partial re-training is applied only to certain layers, while the others are kept the same. This process is particularly useful with limited target domain data, as it reduces overfitting while still benefiting from the learned representations.

4.3.2 Implementation

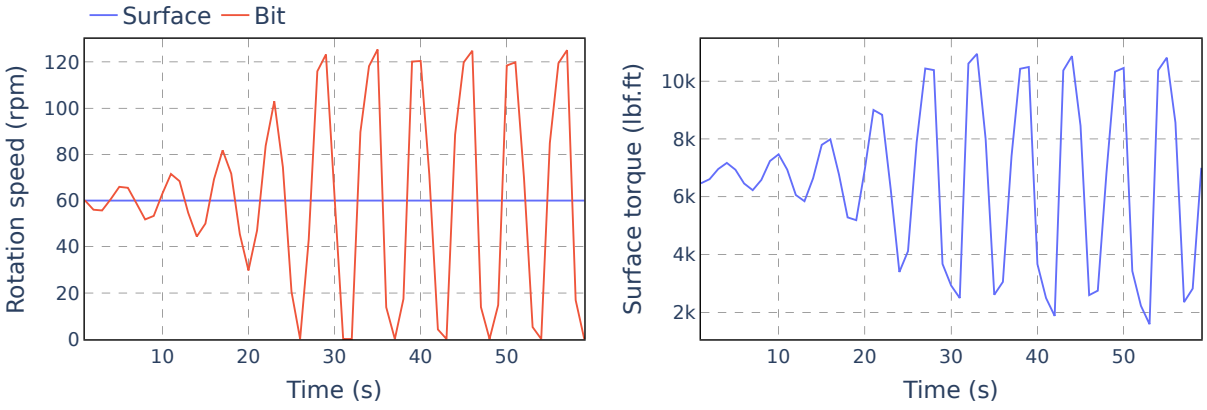
We assume in this section that a first SSI prediction model has already been trained. To improve its performance, TL can be applied using a small portion of the drilling data of the test well. Practically, this involves collecting downhole data from the new well before using the model for SSI prediction. Downhole data can typically be collected during tripping-out operations, which refers to the process of pulling the drill string out of the wellbore. This operation is often necessary for changing the drill bit, performing maintenance, or collecting measurements using memory-based tools or other downhole sensors used to assess wellbore conditions and formation properties.

The TL strategy proposed here consists in retraining only the first layer of the original LSTM model, i.e., only the parameters of the first layer are optimized to match the data from the new well. This yields a lighter computational burden while still providing a significant performance increase. However, it is possible that the data used for retraining contains no stick-slip occurrences. This is a critical point, as applying transfer learning with an unbalanced dataset will result in a model with low precision [49]. To mitigate this issue, a data augmentation process is applied by adding sequences with high SSI to the drilling data of the subject well [67]. This process is described in the next section.

4.4 Data augmentation

Data augmentation has been shown to improve the robustness of machine learning models [68]. In our case, two data augmentation methods to add sequences with high SSI are tested: incorporating synthetic sequences generated using the stick-slip physics-based model proposed by Dao et al. [69], or adding sequences from wells located in other fields. The physical model generates sequences with high SSI values based on parameters such as bit size, hole diameter, drill string configuration, trajectory, and drilling settings of the target well. An example of a generated sequence with SSI = 1.05 is illustrated in Fig. 4.2. The sequence is generated using the lateral trajectory of well 4 (see Fig. 3.7), at a depth of 2000 ft, with static and dynamic friction coefficients equal to 0.4 and 0.3 respectively, and a constant surface rotation speed at 60 rpm.

The second method consists of randomly selecting sequences with high SSI from wells previously used to train the source model. These sequences are then combined with data from the new well, creating a diverse dataset with a wide range of SSI values, which can be used for applying TL.



(a) (b)

Figure 4.2: Example of 60-second generated sequence with a SSI = 1.05: (a) Bit and surface rotation speed; (b) Surface torque. The sequence is generated at depth = 2000 ft, using the lateral trajectory of well 4 (see Fig. 3.7).

4.5 Physics integration

Machine learning models work by capturing the trends and patterns in the data. The sole dependence of these models on the availability of labeled data makes them agnostic to the scientific principles of the processes, with no guarantee to produce scientifically consistent predictions [48]. Combining the scientific knowledge of physics-based models with machine learning models might be a potential way to improve the generalization and the scientific consistency of the model's prediction.

One way to integrate physics for SSI prediction is to add physical features, calculated using physical models, as inputs alongside the surface measurements. For each well, trajectory, BHA, and set of drilling parameters, the following physical features are calculated:

- Total side force [70] refers to the sum of all side forces along the drill string, each side force being the component of a contact force between the drill string and the wellbore that acts perpendicular to the wellbore wall.

- Equivalent rigidity, which represents the drill string's overall resistance to torsional deformation.

$$K_{\text{eq}} = \frac{1}{\sum_{i=1}^{N} \frac{L_i}{G J_i}}$$

  where L_i and J_i are respectively the length and inertia of the element i, and G is its shear modulus [71].

- Bit aggressiveness defined as

$$Agg_{\text{Bottom}} = \frac{TOB}{WOB}$$

  where TOB and WOB denote the surface-estimated torque-on-bit and weight-on-bit, respectively.

4.6 Results and discussions

To demonstrate the application of the proposed baseline model on real data, two scenarios are tested. The first one is to train the model using drilling data of multiple wells, then test it directly, with no retraining, on a new well. The second scenario evaluates the effectiveness of transfer learning. Instead of using the pre-trained model directly on the subject well, the model is fine tuned incrementally using newly received surface and downhole drilling data from the well. For testing, the first 10 % of the total available drilling data of the new well is used for TL. The model is tested on multiple wells which are not part of the training data set. For model evaluation, Mean Absolute Error (MAE) is calculated between the predicted and true SSI (calculated using downhole bit rotation speed) as follows:

$$MAE = \text{mean}(|SSI_{\text{True}} - SSI_{\text{Predicted}}|) \tag{4.9}$$

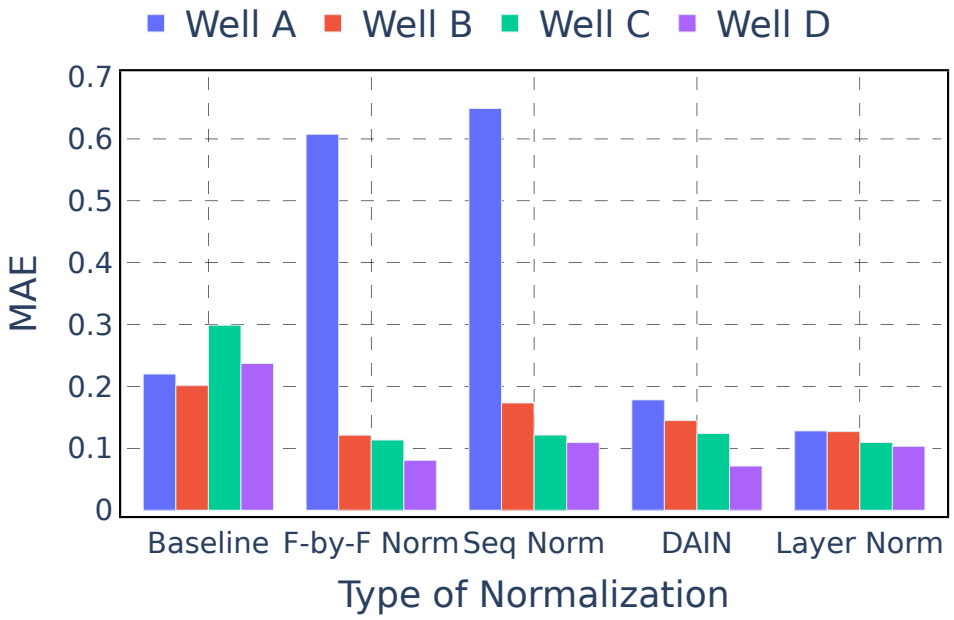
In this section, the analysis of the proposed techniques used to improve model generalizability is based only on data from wells 4, 6, 7, and 9 (see Table 3.1). These wells were chosen as the analysis was carried out before the full dataset became available. Additional information about the well data is provided in Section 3.4. For clarity, we refer to well 4 as A, well 6 as B, well 7 as C, and well 9 as D.

4.6.1 Comparison of data normalization methods

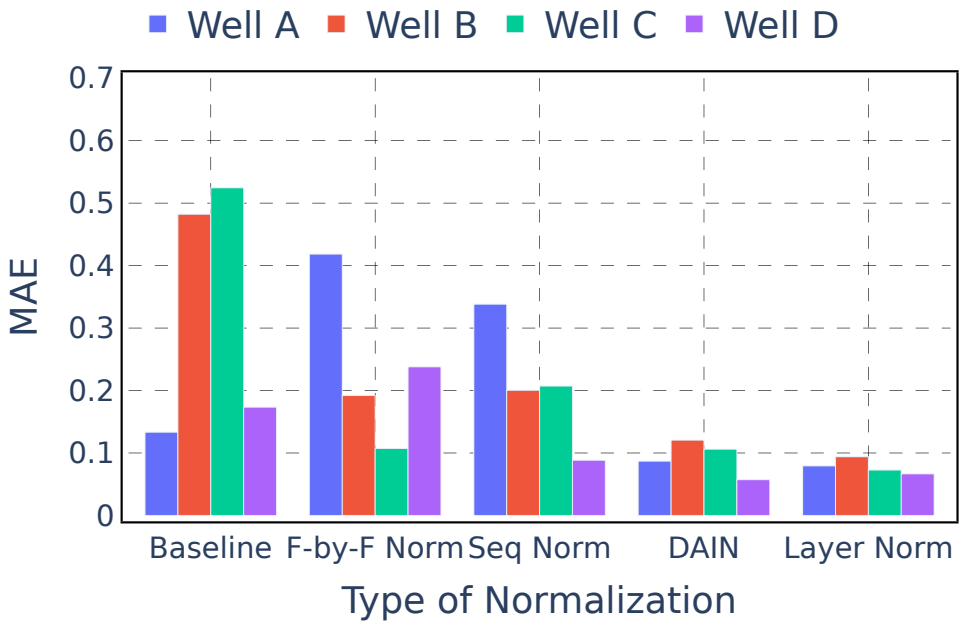
The impact of the different normalization strategies on the baseline model's generalization performance are evaluated under two scenarios: with and without TL. The performance of the proposed baseline model and the normalization methods are shown in Fig. 4.3. Due to the limited dataset of only four wells (A, B, C, and D), the evaluation consists of four cases. In each case, the model is trained on three wells and tested on the remaining one. For example, the model is trained on wells B, C, and D and tested on well A in the first case, while in the second case, it is trained on wells A, C, and D and tested on well B. This approach ensures that each well is used once as a test set, providing a comprehensive assessment of the model's performance across all wells. The results show that LN and DAIN methods yield the lowest error values. Even when the model is partially retrained by transfer learning (Fig. 4.3b and Fig. 4.5), they contribute always the most into the generalization of the model compared to feature-by-feature normalization and sequence normalization.

Examples of the model SSI predictions for two test wells, with and without the application of TL, are presented in Fig. 4.4 and Fig. 4.5 respectively. Each point of the plots represents a 60-second long sequence. The results highlight the effectiveness of DAIN and LN in improving the generalization of the LSTM baseline model on wells from different drilling fields than those used for training. However, for some sequences, the model may over- or underestimate the predicted SSI, particularly for those with high true SSI values (SSI > 0.8). This is likely due to the rarity of such sequences in the training data, as discussed in Section 3.4.4.

Given the performance of the LN method, where the MAE is slightly lower than that of DAIN (Fig. 4.3), it will be used to normalize the data in all subsequent tests.



(a) Without TL

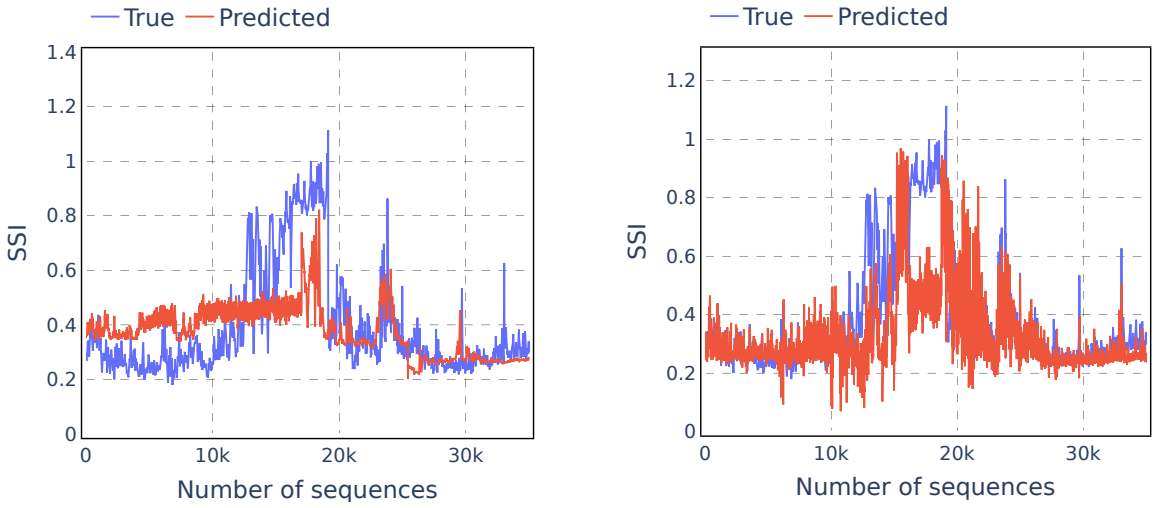


(b) With TL

Figure 4.3: Baseline model results tested on four wells with different types of normalization: F-by-F Norm (Feature-by-feature normalization), Sequence Norm (Sequence Normalization), DAIN, and Layer Norm (LN), shown for two cases: (a) without TL and (b) with TL application. Wells A, B, C, and D correspond to wells 4, 6, 7, and 9, respectively, described in Table 3.1.

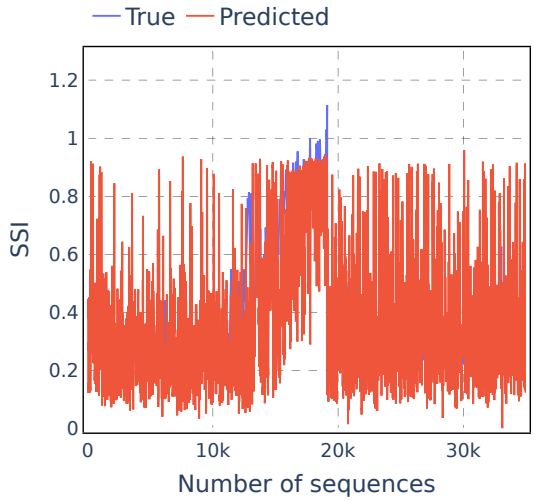
4.6.2 Transfer learning contribution

Fig. 4.6 illustrates the actual and predicted values of SSI over time for well D with and without the application of transfer learning. After applying TL using the first 10 % of the available drilling data from well D, which contains sequences with different levels of SSI (Fig. 4.9a), the

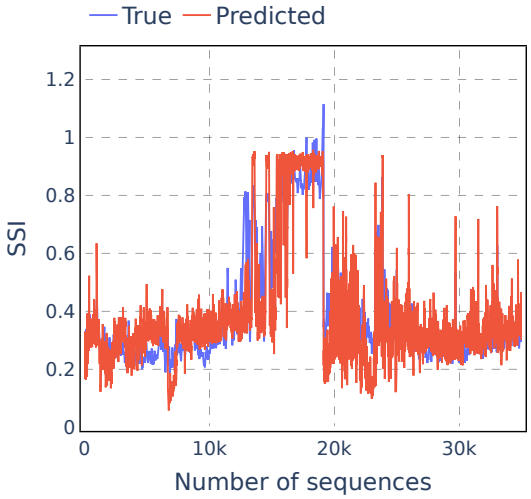


(a) Baseline

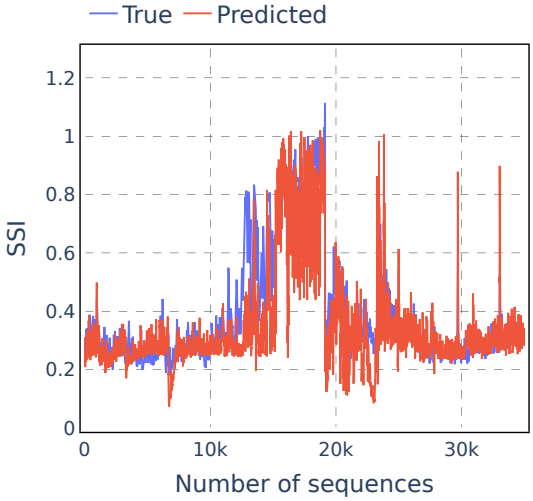
(b) F-by-F normalization



(c) Sequence normalization

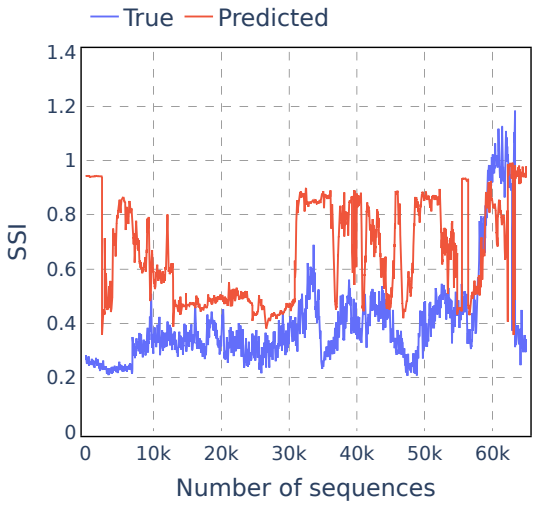


(d) DAIN

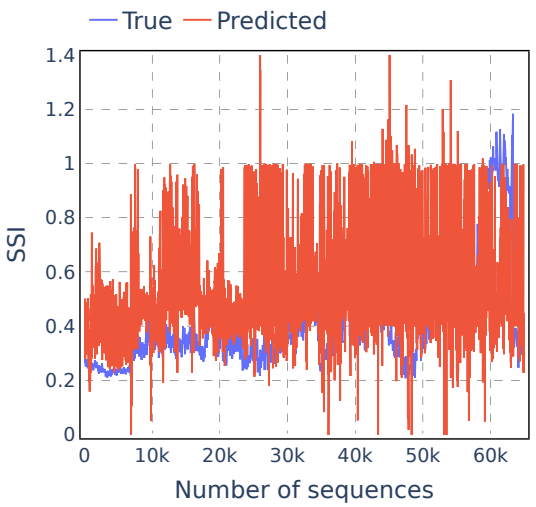


(e) LN

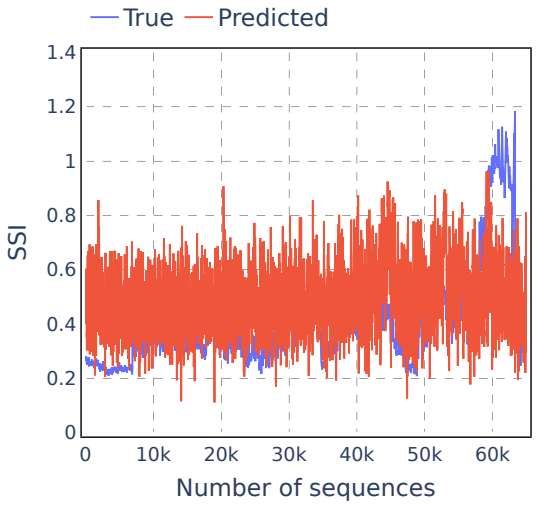
Figure 4.4: Comparison of the predicted SSI over time for well D, withou TL application, applying different normalization methods:(a) F-by-F Norm (Feature-by-feature normalization); (b) Sequence normalization; (c) DAIN; and (d) LN.



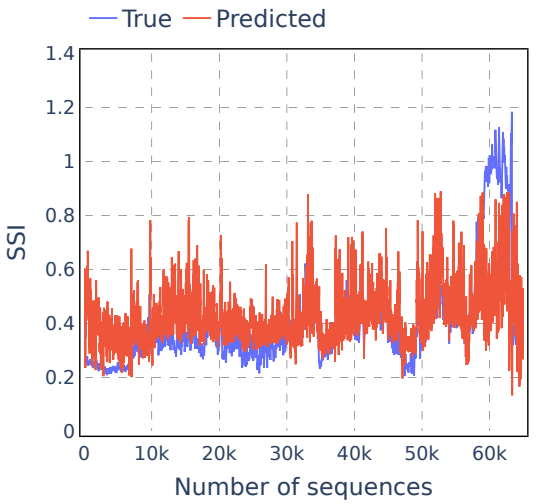
(a) Baseline



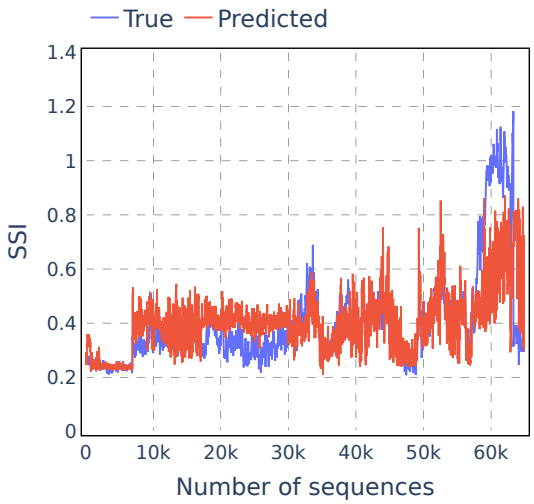
(b) F-by-F normalization



(c) Sequence normalization



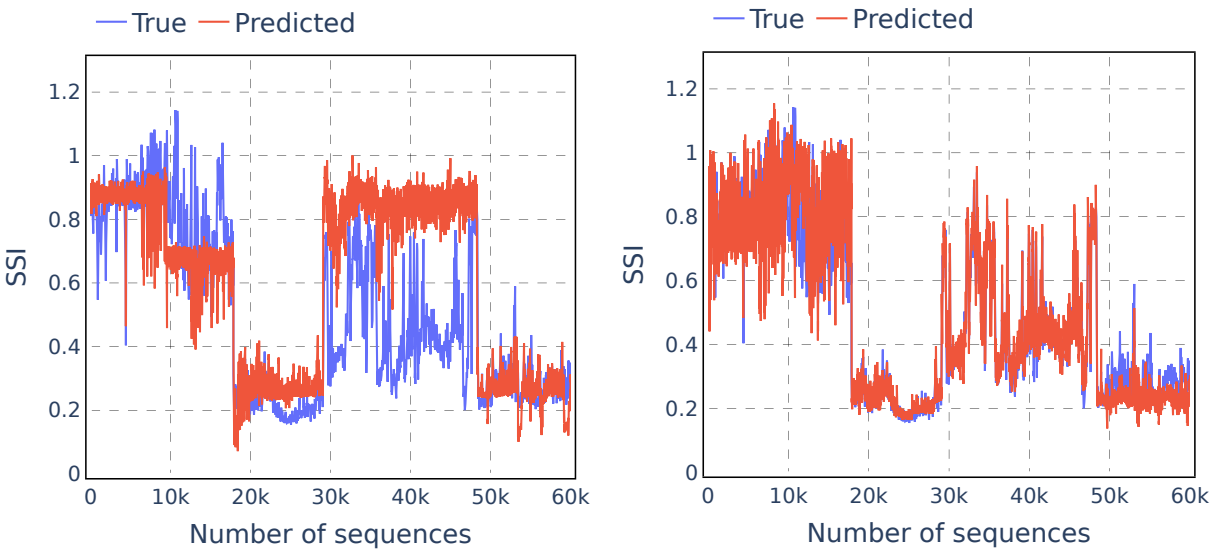
(d) DAIN



(e) LN

Figure 4.5: Comparison of the predicted SSI over time for well B using TL, applying different normalization methods:(a) F-by-F Norm (Feature-by-feature normalization); (b) Sequence normalization; (c) DAIN; and (d) LN.

new model demonstrates improved prediction accuracy compared to the original one (source model). Table. 4.1 shows overall results of transfer learning contribution deployed on the 4 wells. As demonstrated in Table. 4.1, the MAE decreases after the model fine-tuning. In addition to that, the transfer learning application is fast (30 seconds to retrain the model), and the quantity of data used is relatively small compared to the quantity needed to train a new model.



(a) Without TL

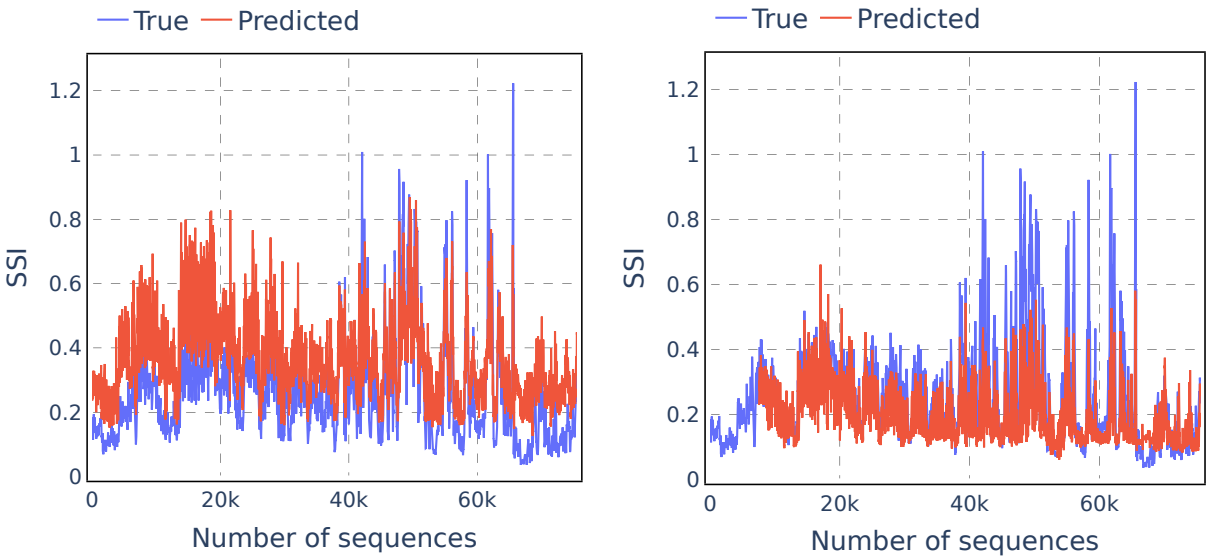
(b) With TL

Figure 4.6: Comparison LSTM model predictions on part of well D: (a) Without TL; (b) With TL.

Well	Without TL	With TL	Improvement (%)
A	0.13	0.08	38.46
B	0.12	0.09	25
C	0.11	0.07	36.36
D	0.1	0.06	40

Table 4.1: MAE values of SSI prediction using the baseline model with LN as the normalization method, with and without transfer learning. Wells A, B, C, and D correspond to wells 4, 6, 7, and 9, respectively, described in Table 3.1.

Despite the improvement provided by TL, the MAE metric hides some shortcomings of the retrained model. To illustrate this, Fig. 4.7 and Fig. 4.8 present the model SSI predictions for well A and the corresponding MAE evolution over time, respectively, with and without the application of TL. The first 10 % of well A's drilling data, used for model retraining, only contain sequences with low SSI (≤ 0.5) (Fig. 4.9b). As a consequence, the model fails to predict high SSI values, as seen in Fig. 4.7b and Fig. 4.8. This is crucial as high SSI correspond to severe vibrations and potential damage. To avoid this undesirable effect, data augmentation techniques to add sequences with high SSI (≥ 0.5) are applied, as detailed in the next section.



(a) Without TL (b) With TL

Figure 4.7: Comparison of the LSTM model predictions on well A: (a) Without TL; (b) With TL.

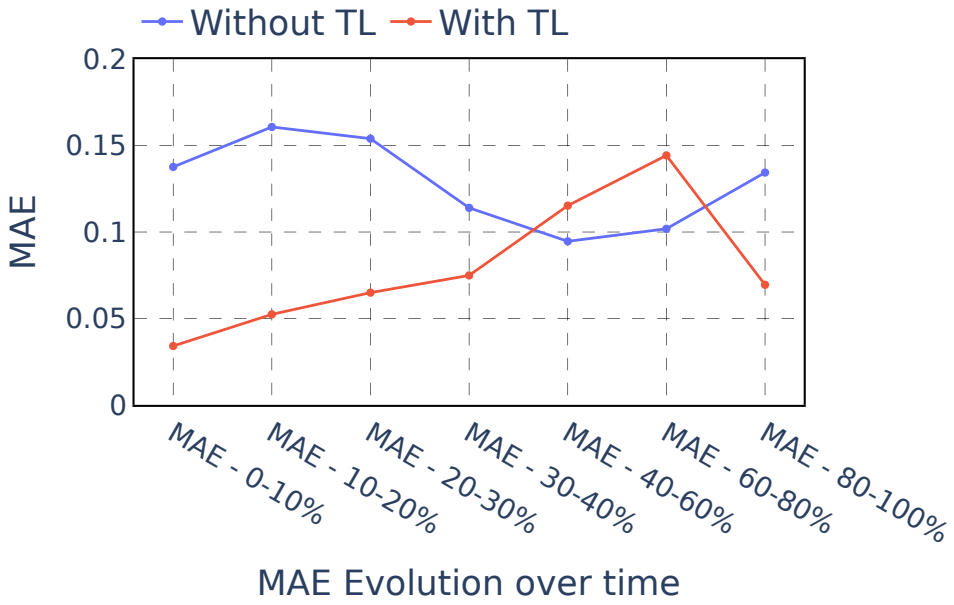
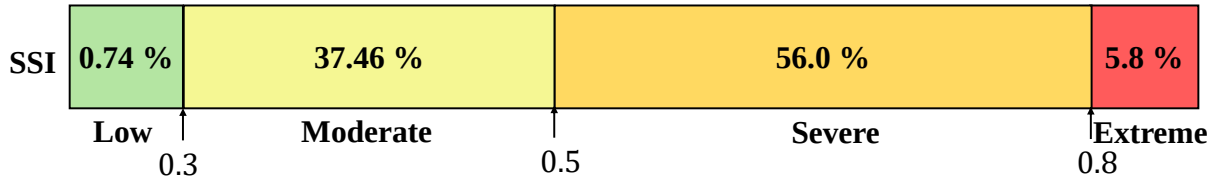


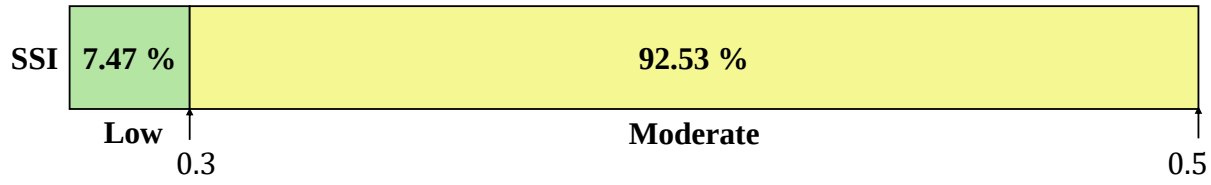
Figure 4.8: MAE evolution over time for the baseline model with LN as the normalization method, with and without TL for Well A. The TL is applied only once, using the first 10 % of well A's drilling data. For the remaining 90 % of the data, the SSI is predicted for each 10 % or 20 % segment, and the mean of MAE of each segment is calculated.

4.6.3 Impact of data augmentation

Fig. 4.10 presents the model predictions after applying TL with data augmentation (see Section. 4.4), either by incorporating synthetic sequences (Fig. 4.10a) or by adding sequences from



(a) Well D



(b) Well A

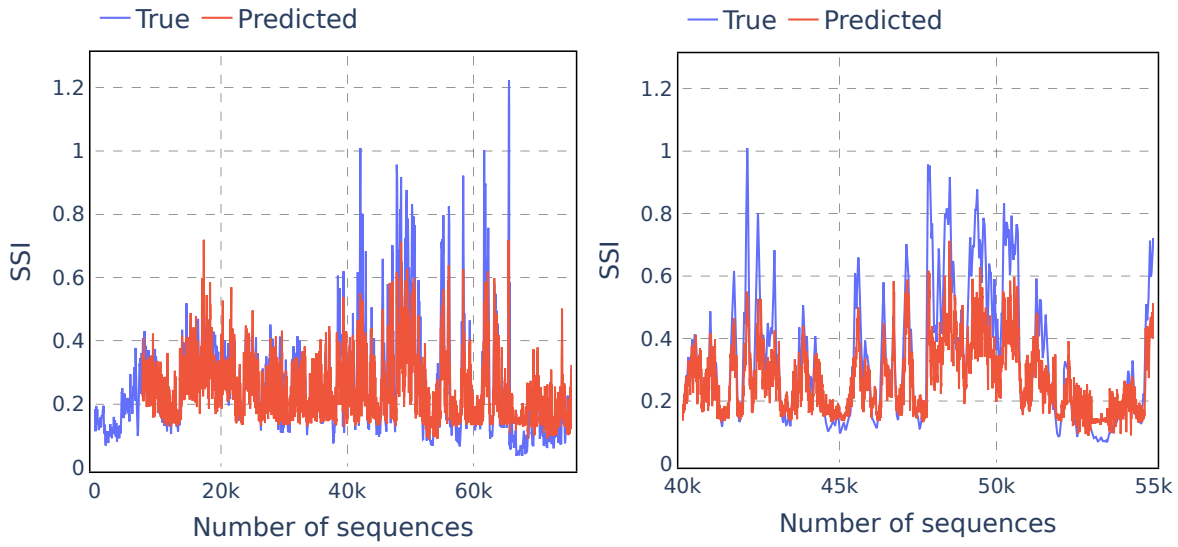
Figure 4.9: Distribution of true ssi values, calculated using downhole data, within the first 10 % of available drilling data from Wells A and D, used for transfer learning.

other wells (Fig. 4.10b). In the case of adding data from other wells, 2000 sequences with high SSI (SSI$\geq$ 0.5) from wells B, C, and D are combined with the first 10% of well A drilling data (approximately 7500 sequences). For synthetic data augmentation, only 350 sequences are generated, due to the sequence generation time.

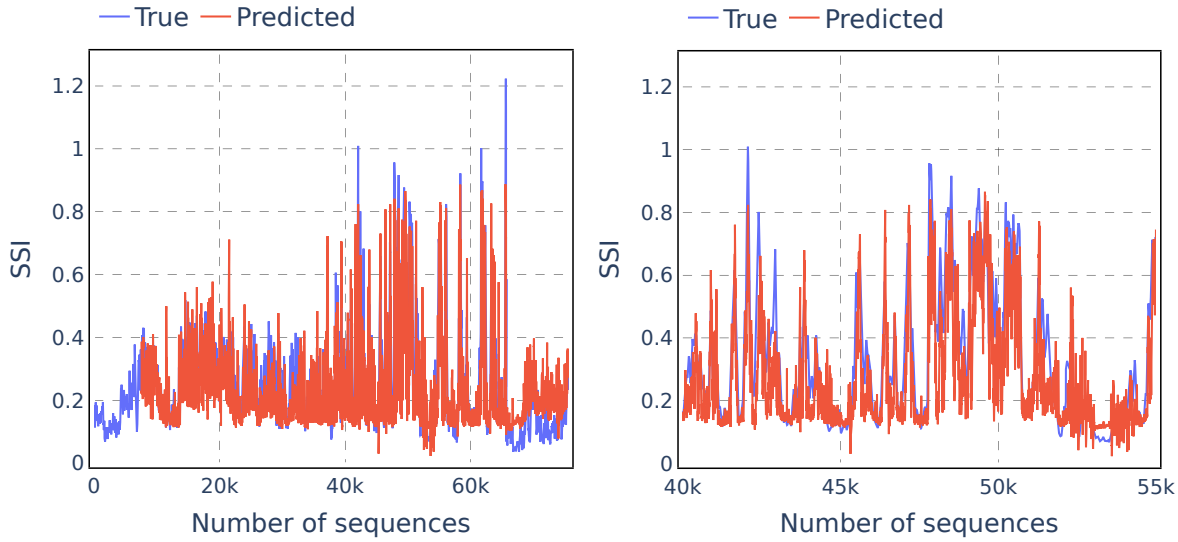
Fig. 4.11 shows the MAE evolution over time for the baseline model in three scenarios applied to test Well A: (1) with TL using only the first 10 % of Well A's drilling data, (2) the first scenario with additional synthetic sequences, and (3) the first scenario with additional sequences from other wells. The results indicate that incorporating sequences from other wells improves the prediction of high SSI values more than using synthetic sequences, as illustrated in Fig. 4.10 for the zoomed interval and Fig. 4.11 for the sequence between 40–80% of the total data. The MAE across all the test well decreases from 0.08 when transfer learning is applied without additional data, to 0.06 with the addition of sequences from other wells, and to 0.07 with the addition of synthetic sequences. This result is mainly caused by the size of the dataset from other wells, which is larger than the relatively small set of generated synthetic sequences.

4.6.4 Addition of physical features

The physical features introduced in Section 4.5 are computed individually for each test well. The hybrid model is trained using 60-second sequences of surface measurements along with the additional physical features as inputs. The performance of the hybrid model on the test



(a) With the addition of synthetic data (left: all sequences; right: zoom)



(b) With the addition of sequences from other wells (left: all sequences; right: zoom)

Figure 4.10: Comparison of TL performance on well A after data augmentation: (a) With the addition of synthetic data; (b) With the addition of sequences from other wells (including a zoomed-in view of a short sequence).

wells is presented in Fig. 4.12. In three of the four test wells, the hybrid model outperforms the purely data-based model, yielding a lower MAE.

On average across all four wells, the data-based model achieves a mean MAE of 0.11, whereas the hybrid model reduces it to 0.09. This suggests that the physical features provide information not captured by surface measurements alone, such as drill string characteristics and contact forces, which influence the severity of downhole torsional vibrations and help improve prediction accuracy.

For the one well where the hybrid model does not outperform the data-based model, this can be explained by the fact that the surface measurements already capture most of the variability affecting the SSI. In this case, adding physics-based features may introduce redundant

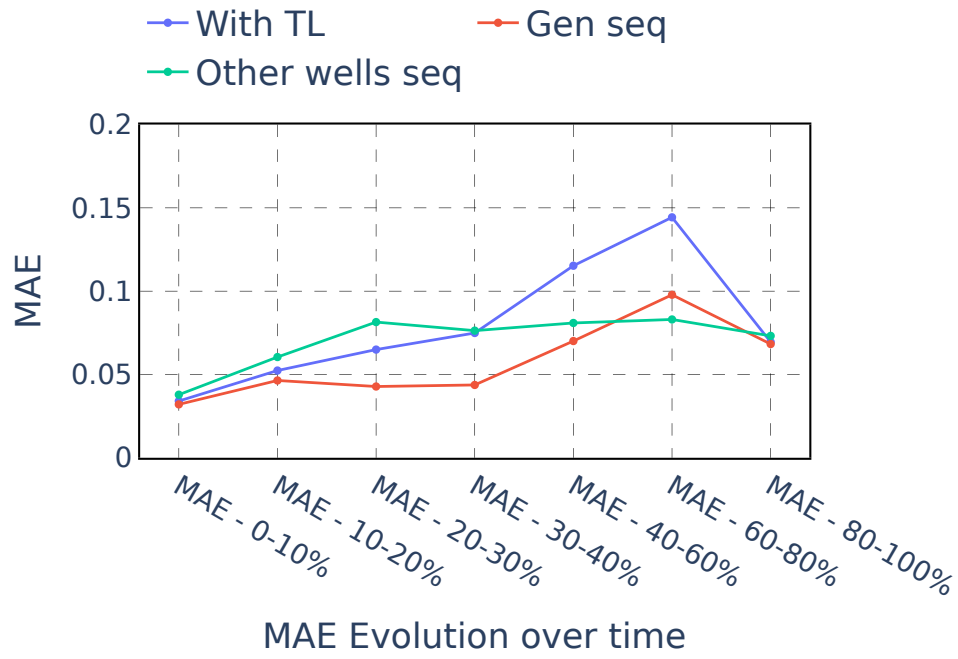


Figure 4.11: MAE evolution over time for baseline model applied to Well A: with TL: with TL using only the first 10 % of Well A's drilling data; Gen Seq: the first case with additional synthetic sequences; and Other Wells Seq: the first scenario with additional sequences from other wells. The SSI is predicted for each 10% or 20% segment, and the mean MAE of each segment is reported.

information or noise, slightly reducing the model's performance.

Throughout the remainder of this dissertation, the term "baseline model" refers to the architecture composed of five LSTM layers, each followed by a LN layer, and ending with a dense output layer with a single neuron. Unless otherwise specified, this model does not incorporate transfer learning and relies solely on surface measurements, without incorporating physical features.

The exclusion of physical features is due to practical constraints. Computing them requires complete information about the well trajectory, the composition of the BHAs, and the integration of these details into the software used for physical calculations. For some of the wells studied in this dissertation, the composition of the BHAs is not available, making it impossible to derive the physical features.

4.7 Comparison of predicted SSI with TVI

As discussed in Section 3.2.4, attenuation during signal propagation and aliasing introduced by data preprocessing can cause surface torque-based indicators, such as the TVI (Section 3.2.2) or power spectrum analysis (Section 3.2.3), to underestimate the severity of downhole torsional vibrations. An example of this is shown in the sequence presented in Fig. 3.5. This behavior is not limited to a single sequence, analysis over the entire drilling data set (Fig. 3.6) consistently shows that surface torque underestimates downhole vibration severity. This has motivated the use of a machine learning model to predict downhole SSI from surface measurements, rather

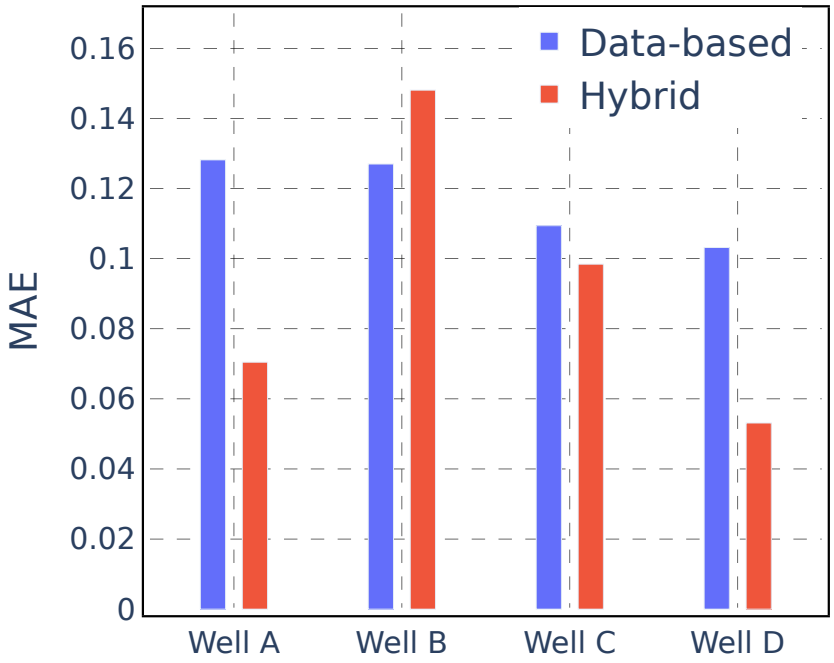


Figure 4.12: MAE values for the hybrid model (with physical features) and the data-based model (without physical features) across the test wells.

than relying solely on torque.

In this section, we compare the predicted SSI with the calculated TVI and the true SSI obtained from downhole measurements. The goal is to assess whether the trained model can accurately predict high-severity downhole SSI despite attenuated surface signals.

The baseline model is used to predict the SSI. Fig. 4.13 shows the predicted SSI variations over time for the sequence presented in Fig. 3.5, focusing on the interval between 05:55 and 06:05 where the true SSI is high but surface torque variations fail to reflect it. As we can see, the predicted SSI closely follows the true SSI, with only minor deviations for sequences with very high SSI (i.e., SSI ≥ 1). These deviations are likely due to the limited number of high-SSI examples in the training data. Except for this small interval (05:54 to 06:04), the model accurately captures downhole stick-slip severity, even for vibrations attenuated or suppressed due to aliasing, as discussed in Section 3.2.4.

The analysis is performed also on the last three test wells listed in Table 3.1. Fig. 4.14 shows the distribution of true SSI, predicted SSI and TVI. Each 60-second sequence is used as input for the baseline model and for TVI calculation using Eq. (3.2). Predicted SSI clearly separates the different stick-slip severity classes, with some mispredictions. In contrast, TVI increases gradually across classes, but the differences are small, making it difficult to distinguish sequences. For example, a sequence with TVI = 0.18 is nearly indistinguishable from low, moderate, or severe stick-slip classes, highlighting the advantage of the ML-based SSI predictions.

4.8 LSTM vs Transformer

In recent years, many advancements have taken place in the field of machine learning. Among them, the Transformer architecture, introduced by Vaswani [1], has been developed as an ex-

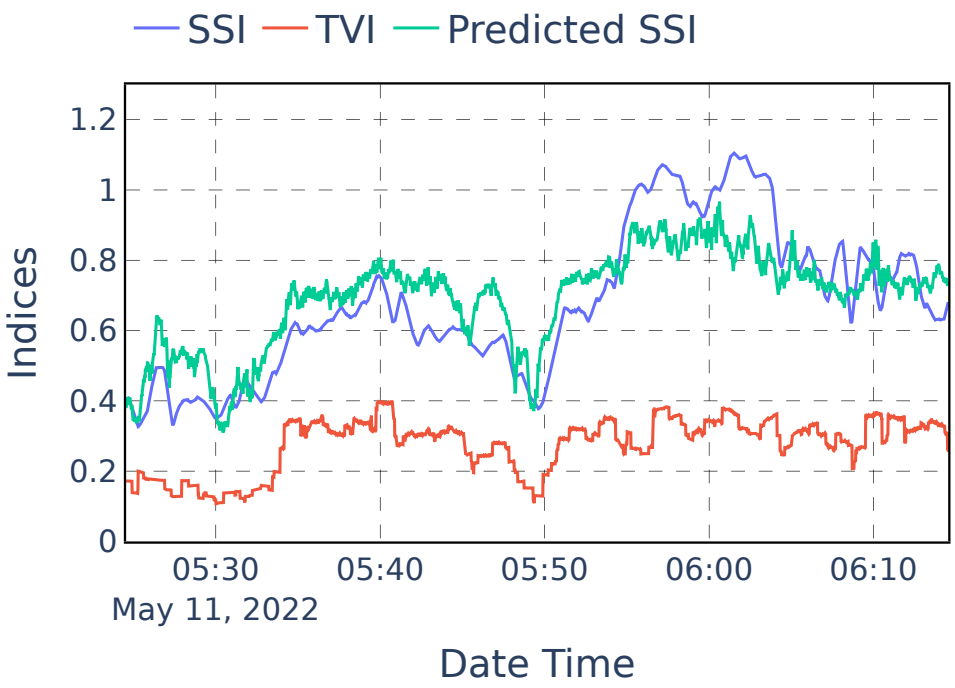


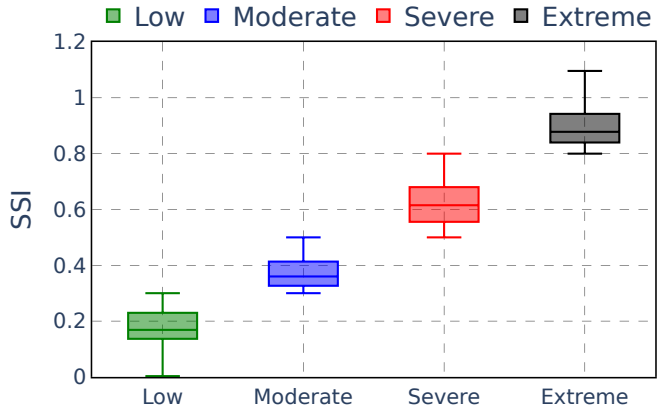
Figure 4.13: Predicted SSI using baseline model over time of sequence presented in Fig. 3.5.

tension of neural networks based on autoencoder frameworks. Transformers are particularly suited for processing sequential data, where the core component, is a self-attention mechanism, enables the model to capture both long-term and short-term dependencies within sequences. Transformers have achieved brilliant achievements in Natural Language Processing (NLP) [72, 73, 74], computer vision tasks, including image classification [75, 76, 77], image deraining [78], object detection [79, 80], action recognition [81, 82], speech recognition [83], as well as in time series analysis [84, 85, 86]. Given its proven effectiveness, a comparison is performed between the baseline model, based on LSTM and LN layers, and a Transformer-based model in the context of the SSI prediction problem.

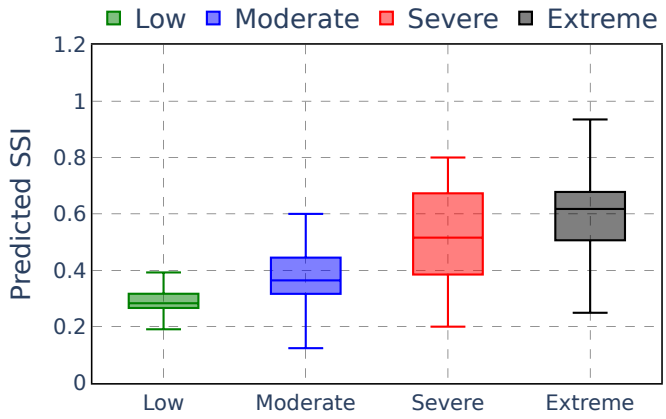
4.8.1 Transformer

Unlike traditional models such as LSTMs, which handle sequences step-by-step, the Transformer [1] uses self-attention mechanisms to capture dependencies across the entire input sequence simultaneously, regardless of the distance between elements. Since Transformers do not process sequences sequentially, positional encodings are employed to provide information about the relative positions of the elements in the sequence. The standard Transformer architecture consists of an encoder and a decoder (see Fig. 4.15), both built from multiple identical layers. The encoder processes the input sequence, while the decoder generates the output sequence, using the encoder's output. Each encoder layer consists of multi-head self-attention followed by a feedforward network. Each decoder layer also includes multi-head self-attention and a feedforward network, but with an additional cross-attention mechanism to attend to the encoder's output. The architecture also integrates residual connections and layer normalization to improve training stability and efficiency [1].

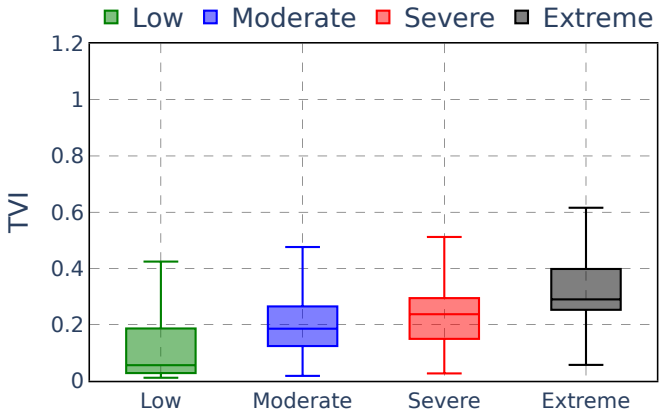
In our work, only the encoder component of the Transformer is used to predict the SSI



(a) SSI



(b) Predicted SSI



(c) TVI

Figure 4.14: Comparison of the true SSI, predicted SSI using baseline model and TVI over test wells. The categories for SSI are defined as follows: Low (SSI $\leq$ 0.3), Moderate (0.3 < SSI $\leq$ 0.5), Severe (0.5 < SSI $\leq$ 0.8), and Extreme (SSI > 0.8).

from sequences of surface measurements, as our model is a seq-to-one model. The architecture consists of a stack of five encoder layers, matching the number of LSTM layers in our baseline model. Each encoder layer follows the architecture presented in Fig. 4.15 (see Section 5.3.1).

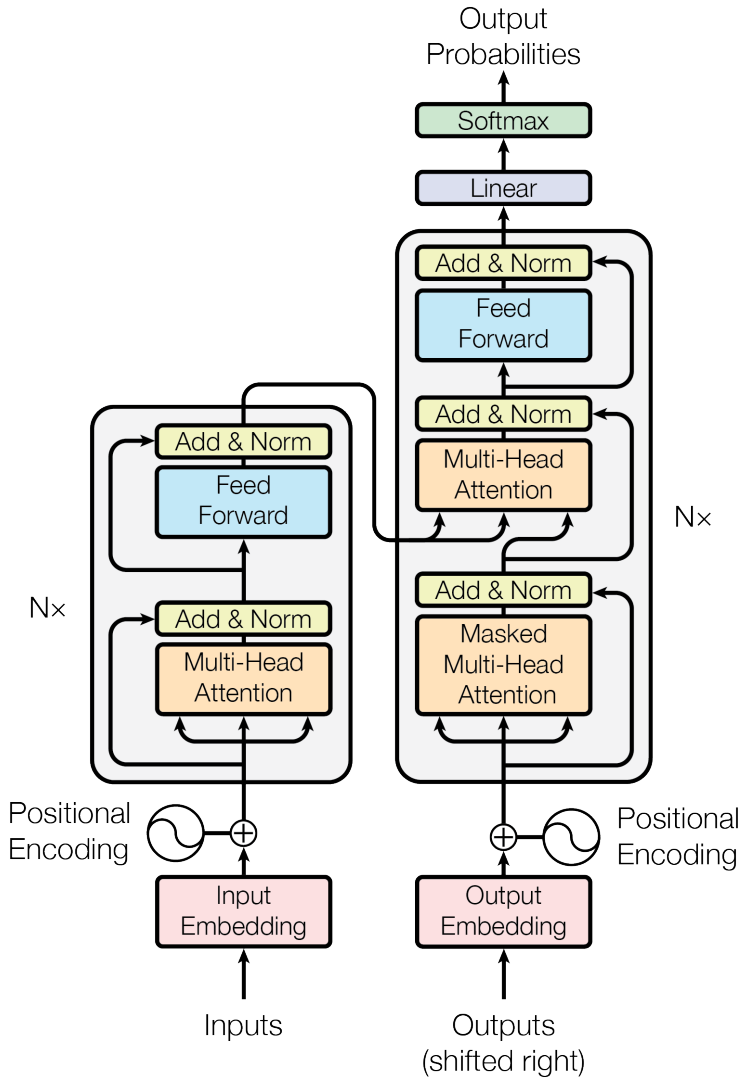


Figure 4.15: Architecture of the Transformer [1].

4.8.2 Results and discussions

For training, the first six wells listed in Table 3.1, which are originated from different fields, are selected, with 10 % of the training data set aside for validation. The remaining three wells are reserved for testing, as two of them are from the same field, with an additional distinct well included to further evaluate the model's generalizability. For the training process, Mean Squared Error (MSE) is used as loss function. Adam optimizer is employed with a learning rate of 10^{-3}. Each model is trained for 1000 epochs, with a batch size of 2048, and the model with the lowest SSI validation MSE is retained. For evaluation, Dynamic Time Warping (DTW) [87] between the actual and predicted SSI of test data, is used as performance metric. DTW is a computational technique used to measure the similarity between two time series that may vary in speed or timing. Given two sequences $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_m)$, we define D as a cost matrix $\in \mathbb{R}^{n \times m}$, where each element $D(i, j)$ represents the cost of aligning point x_i

with point y_j. The default used cost is the squared Euclidean distance:

$$d(i,j) = d(x_i, y_j) = (x_i - y_j)^2$$

Then, DTW is given by:

$$\text{DTW}(i,j) = d(i,j) + \min \begin{cases} \text{DTW}(i-1,j) \\ \text{DTW}(i,j-1) \\ \text{DTW}(i-1,j-1) \end{cases} \tag{4.10}$$

where the boundary conditions are initialized as:

- $\text{DTW}(1,1) = d(1,1)$
- $\text{DTW}(i,1) = d(i,1) + \text{DTW}(i-1,1)$, for $i = 2 \ldots n$
- $\text{DTW}(1,j) = d(1,j) + \text{DTW}(1,j-1)$, for $j = 2 \ldots n$

and the final DTW distance is $\text{DTW}(X,Y) = \text{DTW}(n,m)$.

The key advantages of DTW are its robustness to outliers, making it less sensitive to abnormal points, and its ability to stretch or compress parts of the time series to achieve optimal alignment. In contrast, MSE calculates point-by-point differences in magnitude between the two time series, without accommodating variations in timing or alignment. However, MSE is generally easier to interpret, as it measures the average of the squared differences between predicted and actual values, making it more straightforward than DTW. To facilitate comparisons, a normalized DTW by the number of corresponding well sequences is used.

The baseline model requires approximately four hours of training, while Transformer-based model takes around 21 hours, on an Nvidia GPU. Table. 4.2 shows the normalized DTW results for the three test wells, comparing the LSTM baseline and the Transformer-based models. Overall, the baseline model performs better and generalizes more effectively. Fig. 4.16 illustrates an example of true versus predicted SSI values for a sequence from the first test well (well 7 in Table 3.1).

As observed, the baseline model outperforms the Transformer in terms of prediction accuracy. This performance gap is mainly due to the limited amount of available data. While Transformers typically require large datasets to perform well [88], LSTMs are more suitable for smaller datasets like ours. With a larger dataset, the Transformer might eventually yield better results, but that is not the case in our current setup. Another factor is that Transformers, thanks to their attention mechanism, are able to capture long-range dependencies in sequences much more effectively than LSTMs, which often contributes to their superior performance. However, in the drilling context, such long-range correlations do not appear to be relevant for predicting the SSI. As discussed in Chapter 3, lithology is identified as one of the primary risk factors influencing stick-slip behavior. Since lithology changes during drilling, relying on long-range dependencies in sequences may not enhance prediction accuracy, as the relevant patterns tend to be more localized in time [55, 89].

Additionally, the Transformer requires significantly more training time, approximately five times longer than the LSTM, and demands substantial computational power and resources. For all these reasons, and given the limited size of our dataset, the LSTM baseline model is retained for the remainder of this work.

	Baseline	Transformer	%
Test well 1	0.136	0.156	**12.82**
Test well 2	0.086	0.095	**9.47**
Test well 3	0.107	0.147	**27.21**

Table 4.2: Normalized DTW calculated between the true and predicted SSI for the trained LSTM baseline and Transformer-based models on the three test wells. Test wells 1, 2, and 3 correspond, respectively, to wells 7, 8, and 9 in Table 3.1.

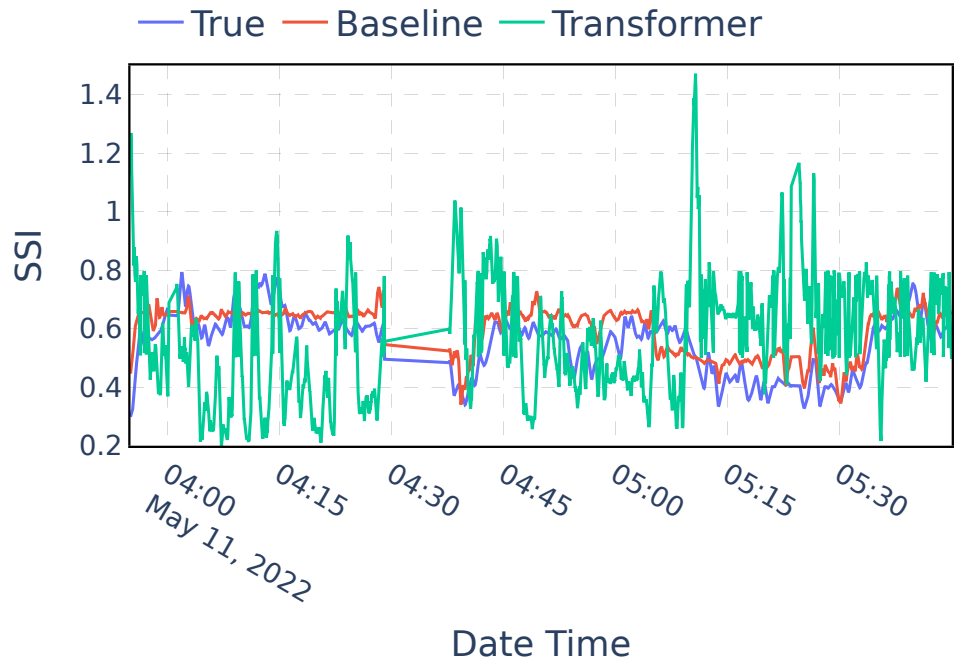


Figure 4.16: True and predicted SSI values over time, for both LSTM baseline model and Transformer-based model of a sequence from the first test well (well 7 in Table. 3.1).

4.9 Permutation Feature Importance

Understanding which input parameters have the greatest impact on the model's predictions is crucial for interpretability and domain comprehension. This section presents an analysis of feature importance, aiming to identify the input features that the model relies on most when estimating SSI.

To approximate feature importance, the Permutation Feature Importance (PFI) approach, as described by [90, 91, 92] is used. This approach involves first calculating the baseline model performance on the test data as is. Then, values of each feature are shuffled across all samples in a random manner, detaching it from the target variable but not altering any of the other features. The model is re-evaluated on this perturbed data, and the drop in performance is noted. A significant drop suggests that the shuffled feature is very important for accurate predictions.

Fig. 4.17 presents the feature importance results for the baseline model using test data. The most influential is surface torque, which captures variations in downhole bit rotation speed, as already discussed in Section 3.1. The next most important factors are total rotation speed (the

sum of surface and downhole motor speeds) and flow rate. Since flow rate directly controls the downhole motor speed, part of its effect is inherently reflected in total rotation speed. Although these features remain nearly constant within individual sequences, their values differ across sequences. Thus, the model appears to exploit their absolute levels rather than their within-sequence variations. For illustration, Fig. 4.18 shows the total rotation speed for one test well: it is constant over long intervals but exhibits abrupt changes at certain depths. From a physical perspective, these findings align with drilling dynamics. High total rotation speeds increase friction between bit and formation, leading to greater wear, heat, and stick-slip behavior [93, 94]. Flow rate also plays a critical role by improving lubrication between the drill string and borehole wall, thereby reducing vibrations and lowering the risk of stick-slip [95]. In addition, studies such as Berlioz et al. [96] show that increasing fluid density and viscosity can suppress lateral vibrations and dissipate energy, further stabilizing the drilling process. Flow rate also facilitates efficient cuttings removal, reducing obstructions that could otherwise amplify torsional resistance. This explains why flow rate emerges as a strong predictor of SSI.

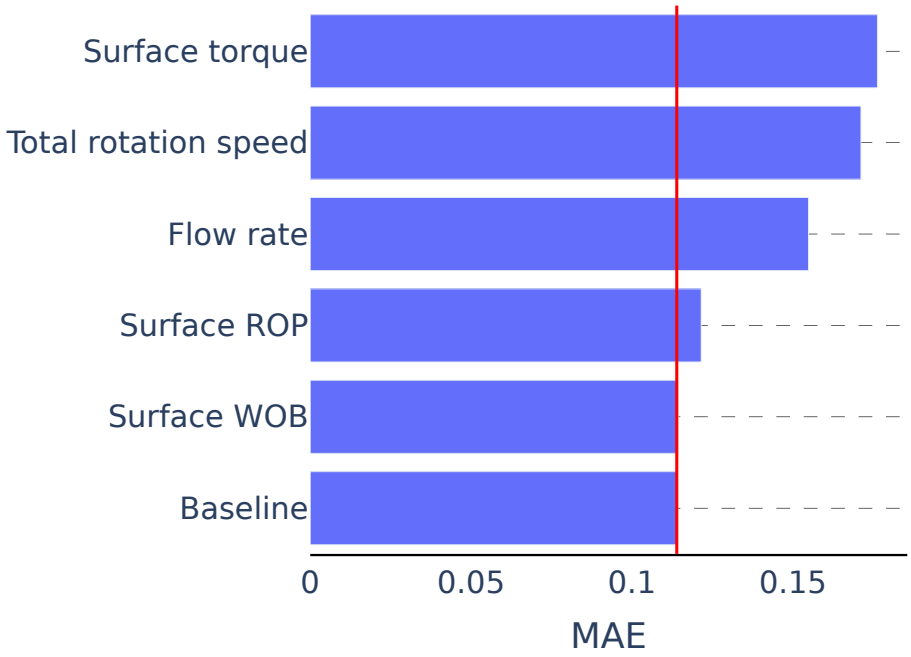


Figure 4.17: PFI of input surface measurements used by the baseline model to predict the SSI. "Baseline" refers to the baseline model's performance on the test data without any feature shuffling.

By contrast, ROP and surface WOB appear less influential in our model. Nevertheless, prior studies indicate their relevance: ROP has been linked to the onset of stick-slip vibrations [97, 98], and increasing WOB is known to raise frictional forces at the bit, thereby intensifying torque and stick-slip severity [99, 93, 100]. Interestingly, our results diverge from those of Hegde et al. [55], where WOB is identified as the most important predictor using a random forest classifier. This discrepancy may be attributed to differences in model type and feature importance methodology. Moreover, in our dataset, WOB and ROP exhibit little variability across the training wells, limiting the model's ability to learn the relationship between their variations and SSI.

The same analysis is performed for the hybrid model, which takes as input the surface

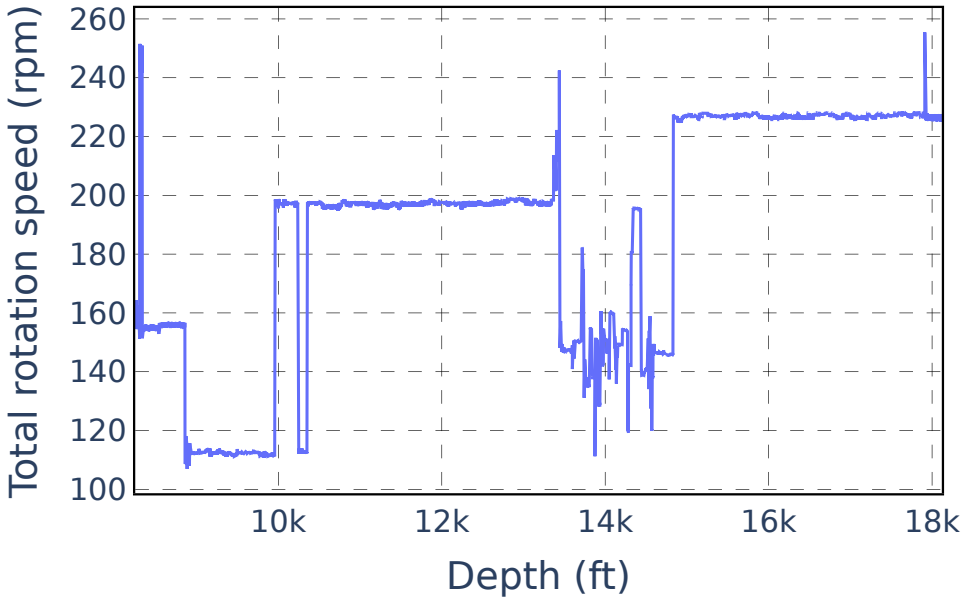


Figure 4.18: Depth variation of total rotation speed in a test well.

measurements and physical features (Section 4.5), with results shown in Fig. 4.19. For the surface measurements, the order of importance and interpretation remain consistent with those observed in the data-driven baseline model. Regarding the additional physical features, the total side force, which represents the normal contact force between the drill string and the borehole, emerges as highly influential. In contrast, the other added physical features remain nearly constant across training and test data. Consequently, permuting these values does not substantially alter the data distribution, which explains their lower importance scores.

4.10 Real-time application

The baseline model (data-based) is integrated into a real-time monitoring software (LiveScan). Fig. 4.20 presents an example of real-time application on a test well (well 8 in Table. 3.1), without application of TL. During the well execution, the process consists in pulling 1-Hz Wellsite Information Transfer Standard Markup Language (WITSML)[1] surface data, including surface torque, surface WOB, ROP, flow rate, and total rotation speed (the input features of the baseline model). The time-based data goes into an operation detection process to filter the rotary drilling data. The filtered data is then divided into 60-second sequences and fed into the model to predict SSI in real-time. The graphs in Fig. 4.20 represent from the left to the right: bit and hole depth, surface torque, true SSI calculated using Eq. (3.1) with 1-Hz measured downhole data, and predicted SSI.

[1] WITSML is a data format used to enable the real-time and historical transfer of wellsite drilling data between systems, companies, and devices in the oil and gas industry.

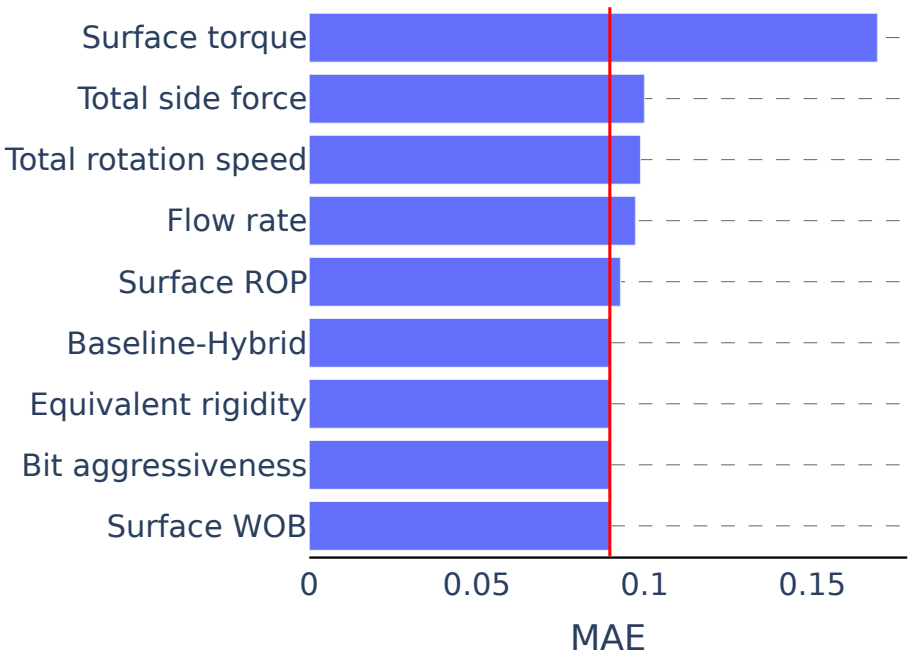


Figure 4.19: PFI of input surface measurements and physical features discussed in Section 4.5, used by the hybrid model to predict the SSI. "Baseline-Hybrid" refers to the model's performance on the test data without any feature shuffling.

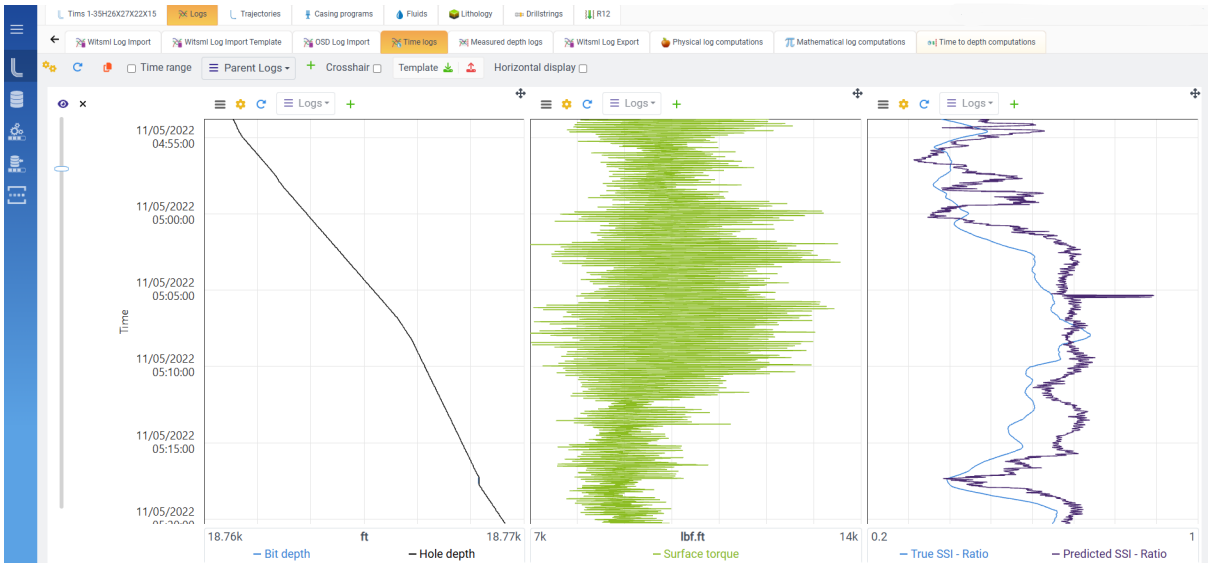


Figure 4.20: Example of real-time application of the proposed model on a test well (Well 8 in Table 3.1). From left to right: bit and hole depth, surface torque, true SSI calculated using Eq. (3.1) from 1-Hz measured downhole data, and predicted SSI obtained with the data-based baseline model.

4.11 Conclusion

In this chapter, three predictive machine learning approaches are proposed to predict SSI using 60-second sequences of 1-Hz surface drilling data as inputs: (i) training a data-based LSTM

model on multiple wells and applying it directly to a new well; (ii) application of transfer learning using a small portion of the first subject well's drilling data; and (iii) training a hybrid machine learning model using physical features in addition to the surface measurements as input. To enhance model performance, several data normalization techniques are evaluated: features-by-feature normalization, sequences normalization, DAIN and layer normalization. The models are trained on recorded historical industrial drilling data. Among these, layer normalization yields the best results. The findings highlight the potential of data normalization to ease the generalization of machine learning models. Transfer learning significantly boosts performance on new wells, and incorporating physics-based features further improves prediction accuracy compared to purely data-driven approaches. Finally, the LSTM baseline model is compared with a Transformer-based model, which requires considerably more training data and computation time. Therefore, the LSTM baseline model is retained as the preferred approach.

Chapter 5

Domain Generalization

Résumé du chapitre

Pour améliorer la généralisabilité du modèle de prédiction de l'SSI de référence présenté au Chapitre 4, plusieurs techniques de généralisation de domaine sont étudiées. Une comparaison approfondie est réalisée afin d'évaluer ces méthodes sur des données temporelles dans le contexte du forage, avec un accent particulier sur la prédiction des valeurs continues de SSI. Plus précisément, le modèle de référence est comparé aux approches de Generalisation de Domaine Adversariale (Adversarial Domain Generalization (ADG)) et de Minimisation du Risque Invariant (Invariant Risk Minimization (IRM)). L'étude examine également l'impact de l'TL sur les performances du modèle dans différents environnements de puits (voir Section 4.3). Les résultats de ce travail ont été publiés dans la revue *Transactions on Machine Learning Research (TMLR)* [28].

Chapter Summary

To improve the generalizability of the baseline SSI prediction model introduced in Chapter 4, several domain generalization techniques are investigated. A comprehensive comparison is performed to evaluate these methods on time series data in the drilling context, with a particular focus on predicting continuous SSI values. Specifically, the baseline model is compared with Adversarial Domain Generalization (ADG) and Invariant Risk Minimization (IRM). The study also examines the impact of TL on model performance across different well environments (see Section 4.3). The results of this work have been published in the *Transactions on Machine Learning Research (TMLR)* journal [28].

5.1 Review

In this section, we review recent approaches proposed to improve the generalization of machine learning models across different domains. Among these, recent advances have focused on aligning feature distributions between source and target domains through Domain Adaptation (DA) techniques, aiming to reduce domain discrepancies and improve target domain performance using existing source and target data. Some approaches achieve this by reweighting or selecting samples from the source domain [101, 102], while others transform the feature space to map the source distribution onto the target [103, 104]. A key factor in these methods is how the similarity between distributions is measured. One approach matches distribution means in a reproducing kernel Hilbert space [105], while Fernando et al. [106] propose mapping the principal axes of the distributions, and Ganin and Lempitsky [107] introduce Adversarial Domain Adaptation (ADA), which modifies feature representations rather than relying on reweighting or geometric transformation. This approach, which employs a deep and discriminately trained classifier to measure distribution separability [108, 109, 110], is extensively used across fields like task and text classification [111, 112], image classification for crack detection [113], medical image analysis [114], sentiment analysis [107, 115, 116, 117] or Named Entity Recognition (NER) [118].

Notably, DA can only be applied when target domain data is available during training. In cases where this is not feasible, a more challenging and realistic approach, known as Domain Generalization (DG) [119], is used. The primary objective of DG is to train a model using one or multiple different but related source domains so that it can generalize effectively to unseen target domains. While ADA was originally proposed for domain adaptation, the adoption of this reasoning has motivated adaptations of this approach for DG [120, 119, 121, 122]. Adversarial Domain Generalization (ADG) has been widely tested, particularly on image datasets for object recognition tasks [123], fault diagnosis [124, 125], anti-spoofing [126, 127], depersonalized cross-subject vigilance estimation problem [128], and fault identification [129].

While ADG has been extensively tested on image data, this dissertation focuses on testing and comparing their effectiveness on time series data. Specifically, the baseline model, introduced in Chapter 4, is compared with two approaches: ADG and Invariant Risk Minimization (IRM), for predicting the severity of downhole torsional vibrations (SSI), using 60-second sequences of 1-Hz surface drilling data in a real drilling context.

5.2 Methodology

Generalization in machine learning refers to the capability of a model to perform effectively well on an unknown *target* domain that differs from the *source* domain on which it is originally trained. In our context of drilling, a training set comprising data from multiple wells is considered, with each well being defined by distinct characteristics. Some of these characteristics are known, such as length, trajectory, and bit diameter, whereas others are unknown, uncertain or changing, such as rock characteristics or bit fatigue. The wells originate from different sites across various geographical locations and correspond therefore to distinct domains having their own unique data distribution. Our goal is to train a model capable to achieve minimal prediction error when applied to test data from a new well, which constitutes a different domain with a data distribution not encountered during training. Unlike traditional machine learning approaches,

which assume that training and testing data originate from the same distribution, the concept of model generalization focuses on achieving robust performance across various domains without prior knowledge of the target domain.

During training, labeled data from one or several source domains is typically available. For the target domain, various situations may be encountered depending on the nature of the available data:

- Only unlabeled samples from the target domains: this scenario is called unsupervised domain adaptation.
- Unlabeled samples from the target domains plus few labeled target samples: this scenario is referred to as semi-supervised domain adaptation.
- No samples from the target domain during training, meaning that the specific target domain in which the model will ultimately be deployed is completely unknown. This situation is called DG, which is the focus of this chapter.

To improve model generalizability, different learning strategies can be employed. These strategies involve the choice of training objectives and loss functions that guide the learning process, enabling the model to extract features that remain stable despite domain shifts. In this dissertation, ADG and IRM are evaluated.

5.2.1 Adversarial Domain Generalization

Let us denote by $\mathscr{X}$ the instance set of measurements conducted on the wells to be used as features. From a mathematical perspective, a *domain* is defined as a specific distribution $\mathscr{D}$ on the instance set $\mathscr{X}$. In our context, each well constitutes its own domain on $\mathscr{X}$. Suppose we have access to N_S wells with labeled data, characterized by the *source domains* $\{\mathscr{D}_S^i\}_{i=1}^{N_S}$. Our objective is to construct a regression algorithm using data from the source domains that generalizes well to new domains.

To specify a learning problem, two ingredients are required: a distribution $\mathscr{D}$ on the instance set $\mathscr{X}$, and an unknown target function $f : \mathscr{X} \to \mathbb{R}$ that we seek to approximate. A classical approach for approximating f is to map instances into a feature space $\mathscr{Z}$ using a representation function $\mathscr{R} : \mathscr{X} \to \mathscr{Z}$ and then select a function h from a hypothesis class $\mathscr{H}$ that utilizes these features to perform the regression task. A distribution on the feature space $\mathscr{Z}$ is induced by the representation function $\mathscr{R}$, which is denoted by $\tilde{\mathscr{D}}$. Furthermore, the function $\tilde{f} : \mathscr{Z} \to \mathbb{R}$ is denoted such that $f(x) = \tilde{f} \circ \mathscr{R}(x)$ for all $x \in \mathscr{X}$.

In our case, N_S labeled dataset $\{\mathscr{D}_S^i\}_{i=1}^{N_S}$, are available where

$$\mathscr{D}_S^i = \{(x_{S_i}^1, f(x_{S_i}^1)), \ldots, (x_{S_i}^{m_i}, f(x_{S_i}^{m_i}))\}$$

consists of m_i measurements sampled i.i.d from the source distribution $\mathscr{D}_S^i$. Our objective is to select a function h from the hypothesis class $\mathscr{H}$, which minimizes the expected risk ε_T on the target domain, defined as:

$$\varepsilon_T(h) = \mathbb{E}_{z \sim \tilde{\mathscr{D}}_T}[L(h(z), \tilde{f}(z))] = \mathbb{E}_{x \sim \mathscr{D}_T}[L(h \circ \mathscr{R}(x), f(x))], \tag{5.1}$$

where L is the loss function used to predict the difference between the prediction $h(z)$ and the target value $\tilde{f}(z)$ and is usually selected to be the MSE, $\mathscr{D}_T$ denotes the original distribution

of the target data, while $\tilde{\mathscr{D}}_T$ represents the distribution of the target data transformed by the representation function $\mathscr{R}$. The main challenge here is that labeled data are not available for the target distribution, making the prediction of the target error impossible to estimate directly. As a consequence, the training process must rely solely on the available labeled data from the N_S source domains, to adapt the training procedure in order to ensure good generalization. Theoretically, it can be shown that to minimize the expected risk ε_T on a new domain, one effective approach is to minimize the empirical risk on the available training source domains, while simultaneously learn a representation that aligns the features across the source domains [123]. The objective is to construct a shared representation that generalizes well across source domains while maintaining strong regression performance.

These approaches are theoretically justified under the following assumptions [130]:

- *Similarity between features distributions across domains*: The representation $\mathscr{R}$ should result in similar features for all source domains. To quantify this similarity, the $\mathscr{H}$-divergence is commonly used [131, 132, 133, 134]. Given two domain distributions $\mathscr{D}_S^X$ and $\mathscr{D}_T^X$ over X, and a hypothesis class $\mathscr{H}$, which is a discrete or continuous set of binary classifiers $\eta : X \to \{0,1\}$, the $\mathscr{H}$-divergence between $\mathscr{D}_S^X$ and $\mathscr{D}_T^X$ is defined as:

$$d_{\mathscr{H}}(\mathscr{D}_S^X, \mathscr{D}_T^X) = 2 \sup_{\eta \in \mathscr{H}} \left| \Pr_{X \sim \mathscr{D}_S^X}[\eta(X) = 1] - \Pr_{X \sim \mathscr{D}_S^T}[\eta(X) = 1] \right| \tag{5.2}$$

  In essence, the $\mathscr{H}$-divergence relies on the capacity of the hypothesis class $\mathscr{H}$ to distinguish between examples generated by $\mathscr{D}_S^X$ from examples generated by $\mathscr{D}_T^X$. As observed by Ben-David et al. [132], this analysis also applies to the multiclass case [1]. In practice, Ben-David et al. [132] suggested that the $\mathscr{H}$-divergence $d_{\mathscr{H}}(\mathscr{D}_S^i, \mathscr{D}_S^j)$ between the source domains $\mathscr{D}_S^i$ and $\mathscr{D}_S^j$ can be estimated by training a classifier C to distinguish between domains i and j based on the features produced by the representation $\mathscr{R}$:

$$d_{\mathscr{H}}(\mathscr{D}_S^i, \mathscr{D}_S^j) \simeq 1 - 2\,\mathrm{err}_{\mathrm{cl}}(C), \tag{5.3}$$

  where $\mathrm{err}_{\mathrm{cl}}(C)$ is the classification error, estimated based on the empirical datasets $\mathscr{D}_S^i$ and $\mathscr{D}_S^j$ with respective sizes m_i and m_j as:

$$\mathrm{err}_{\mathrm{cl}}(C) = \frac{1}{m_i + m_j} \sum_{k=1}^{m_i+m_j} L_{\mathrm{cl}}\big(C(z_k), \mathbf{1}_{\mathscr{D}_S^i}(z_k)\big). \tag{5.4}$$

  In Eq. (5.4) L_{cl} is a classification loss, usually the cross-entropy loss and $\mathbf{1}_{\mathscr{D}_S^i}(z)$ is the indicative function taking the value 1 when z comes from dataset $\mathscr{D}_S^i$. A high classification loss suggests that the classifier struggles to differentiate between the source domains. Conversely, a small $\mathscr{H}$-divergence implies high similarity between the distributions $\mathscr{D}_S^i$ and $\mathscr{D}_S^j$.

- *Existence of a suitable mapping function*: There must exist a function h^* that can accurately map features z to their respective labels, regardless of the domain.

[1] However, to obtain the same results when $|Y| > 2$, it is necessary to assume that $\mathscr{H}$ is a symmetric hypothesis class. Specifically, for all $h \in \mathscr{H}$ and any permutation of labels $c : Y \to Y$, it holds that $c(h) \in \mathscr{H}$.This assumption is typically met by most commonly used neural network architectures.

In our case, for the SSI prediction, both assumptions are validated. Regarding the first assumption, there is always a degree of similarity between feature distributions, even when the input feature distributions differ. This is because drilling operations typically rely on standardized tools and techniques, which help maintain consistency in surface measurement patterns across wells. Consequently, the overall structure of input feature distributions remains comparable across domains.

For the second assumption, prior studies and data analysis have demonstrated a correlation between surface measurements, particularly surface torque, and the severity of stick-slip, supporting the assumption that a suitable mapping function exists (see Section 3.4.2). Furthermore, machine learning models have proven effective in capturing these relationships in similar applications, including detection of other downhole vibrations. For example, our work presented in Chapter 4, demonstrates the feasibility of developing a mapping function for SSI prediction. Even if the mapping function is not perfect, a regression model can still approximate it with a reasonable degree of accuracy.

Domain adversarial training is commonly used for learning domain-invariant features. This approach is originally introduced by Ganin and Lempitsky [107, 131] in the context of domain adaptation. As described in Section 5.2.1, minimizing the target error ε_T on a new well amounts to reducing both the expected risks ε_{S_i} on each source domain and the discrepancy between source domains. To that end, a representation $\mathscr{R}$ is learned, that aligns the source domains but still produces features that remain relevant for the regression task at hand [107]. To achieve this, adversarial training is used to learn domain-invariant features by simultaneously training two components: a generator and a discriminator in an adversarial manner. The input x is processed by an embedding function or representation $\mathscr{R}$ (generator) to learn a domain invariant feature representation, which is used as input to the regression model h for mapping to the SSI. To ensure alignment between the source domains, a domain classifier C (discriminator) is trained to distinguish between the source domains based on their embedded features. The goal is to maximize the domains classifier's loss, indicating that the embedded features from the source domains are sufficiently similar, making it difficult for the classifier to distinguish between them. For a dataset constituted of m observations $\{(x_i, f(x_i)) \sim \mathscr{D}_S^k\}_{i=1}^m$, $k \in \{1, \ldots, N_s\}$, the final loss function is:

$$\min \frac{1}{m} \sum_{i=1}^{m} L\Big(h(\mathscr{R}(x_i)), f(x_i)\Big) - \lambda \, L_{\text{cl}}\Big(C(\mathscr{R}(x_i)), \mathbf{1}_{\mathscr{D}_S^k}(x_i)\Big), \tag{5.5}$$

where the second term is used to penalize the distance between the embedded features for the source domains, L_{cl} is the classification loss (cross-entropy loss) computed across the source domains, λ is a weighting coefficient, and $\mathbf{1}_{\mathscr{D}_S^k}$ is the indicative function that takes the value 1 if the element x_i belongs to the source domain $\mathscr{D}_S^k$, and 0 otherwise.

5.2.2 Invariant Risk Minimization

Invariant Risk Minimization (IRM) is a DG technique aimed at learning a new data representation where the correlations remain invariant across different training environments, similar to ADG [135]. Specifically, the model learns a representation $\mathscr{R}(x)$, where x represents the input features. The goal is for this representation to capture the underlying structure of the data invariant across the different environments. IRM seeks to identify a domain-invariant predictor

$g = p \circ \mathscr{R}$ by searching for a regression function p that is simultaneously optimal for all domains $\{\mathscr{D}_S^i\}_{i=1}^{N_S}$, which lead to the optimization problem:

$$\begin{aligned} \hat{\mathscr{R}}, \hat{p} = \arg \min_{\substack{\mathscr{R}: \mathscr{X} \to \mathscr{Z} \\ p: \mathscr{Z} \to \mathbb{R}}} \sum_{i=1}^{N_S} \varepsilon_{\mathscr{D}_S^i}(p \circ \mathscr{R}) \\ \text{subject to } p \in \arg\min_{\bar{p}} \varepsilon_{\mathscr{D}_S^i}(\bar{p} \circ \mathscr{R}), \forall i = 1, \ldots, N_S, \end{aligned} \tag{5.6}$$

where $\varepsilon_{\mathscr{D}_S^i}(p \circ \mathscr{R})$ is the empirical risk associated with the i-th domain. Eq. (5.6) is a challenging bi-level optimization problem. However, as demonstrated in [135], when a linear classifier p is fixed, any predictor $g = p \circ \mathscr{R}$ can be written as $g = p_*.\mathscr{R}_*$ for a scalar p_*. Without loss of generality, this scalar can be absorbed into $\mathscr{R}$, such that $\mathscr{R} = p_*.\mathscr{R}_*$, resulting in $g = 1.\mathscr{R}$. Thus, $\mathscr{R}$ itself becomes the invariant predictor. This reformulation simplifies the problem as follows:

$$\hat{\mathscr{R}} = \arg \min_{\mathscr{R}: \mathscr{X} \to \mathbb{R}} \sum_{i=1}^{N_S} \varepsilon_{\mathscr{D}_S^i}(\mathscr{R}) + \alpha \|\nabla_{p|p=1.0} \varepsilon_{\mathscr{D}_S^i}(p.\mathscr{R})\|^2, \tag{5.7}$$

where α is a penalty weighting coefficient used to balance the Empirical Risk Minimization (ERM) term and invariance of the predictor $1 \cdot \mathscr{R}$ across the different domains. The reader is referred to the original article [135] for a detailed derivation of this result.

5.3 Models characteristics

5.3.1 Baseline model

The baseline model consists of a single component, takes 60-second sequences of 1-Hz surface measurements as input and predicts the SSI, as introduced in Chapter 4. The architecture featuring a series of five LSTM layers, each followed by a LN layer, and concludes with a dense output layer containing a single neuron responsible for outputting the predicted SSI, as illustrated in Fig. 5.1.

5.3.2 Adversarial Domain Generalization model

For ADG , the global model architecture proposed by Ganin et al. [131] is adapted. It comprises three main components (see Fig. 5.2):

- **Generator (or feature extractor):** This component processes 60-second sequences of surface measurements, generating a D-dimensional embedded feature vector $z \in \mathbb{R}^D$. The generator parameters are indicated by θ_G, and the relationship is expressed as $z = G(x, \theta_G)$, where G is the same as $\mathscr{R}$ in Eq. (5.1). The generator architecture comprises a series of LSTM layers combined with LN layers, mirroring the baseline model (see Fig. 5.1) except for the final output dense layer.

- **Discriminator (or domain classifier):** The discriminator takes the embedded features vector z as input and classifies it based on annotated data from the source domains. Its parameters are represented by θ_C, $C(z, \theta_C)$ (see Eq. (5.4)). The discriminator is structured as a fully connected neural network, with a Gradient Reversal Layer (GRL) as input layer

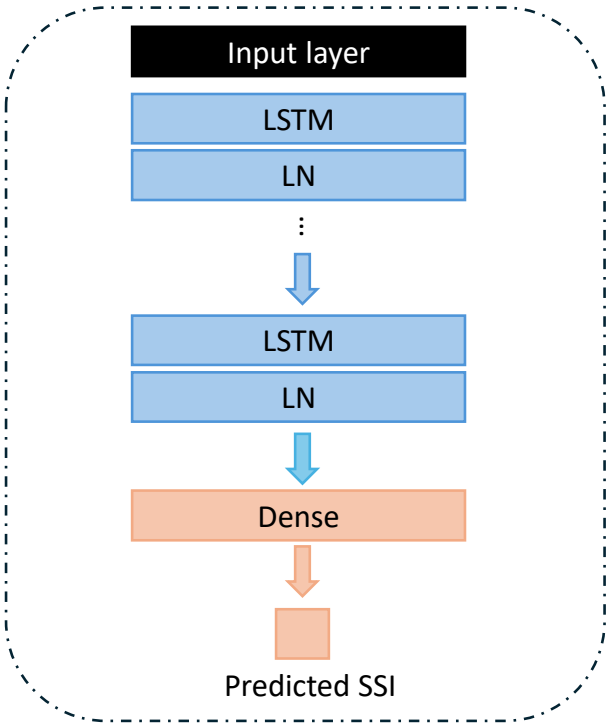


Figure 5.1: Architecture of our LSTM baseline model.

to reverse the direction of the gradients during backpropagation (multiply the classification loss by $-\lambda$ (see Eq. (5.5))).

- **SSI-predictor:** This component utilizes the embedded features from the generator to predict the SSI for each sequence. The parameters for this mapping are denoted by θ_{SSI} and the regression output is $h(z, \theta_{\text{SSI}})$ (see Eq.(5.1)). The SSI-predictor is also a fully connected neural network.

During the training phase, our objective is to minimize the SSI prediction loss for the training wells while optimizing the generator parameters to make the embedded features z (where $z = G(x, \theta_G)$) domain-invariant, which involves aligning the source distributions $\{G(x_i, \theta_G), x_i \sim \mathscr{D}_{s_i}\}$ for $i \in \{1, 2, ..., N_s\}$,where N_s is the number of training wells. To achieve this, the generator parameters θ_G are optimized to maximize the domain classifier loss, thereby promoting domain invariance and making the feature distributions across domains as similar as possible. Simultaneously, the domain classifier and the SSI-predictor parameters, θ_C and θ_{SSI} respectively, are optimized to minimize their corresponding losses. The corresponding optimization problem is given by

$$\hat{\theta}_C, \hat{\theta}_{\text{SSI}}, \hat{\theta}_G = \text{argmax}_{\theta_C} \text{argmin}_{\theta_G, \theta_{\text{SSI}}} E(\theta_G, \theta_C, \theta_{\text{SSI}}) = L(\theta_G, \theta_{\text{SSI}}) - \lambda L_{\text{cl}}(\theta_G, \theta_C) \tag{5.8}$$

where L is the loss function used for the SSI prediction (MSE), while L_{cl} is the classification loss (cross-entropy loss), computed across the N_s source domains. The parameters θ_C of the domain classifier are optimized to minimize the domain classification loss (Eq. (5.9)), the parameters θ_{SSI} of the SSI-predictor are optimized to minimize the SSI prediction loss, while the feature

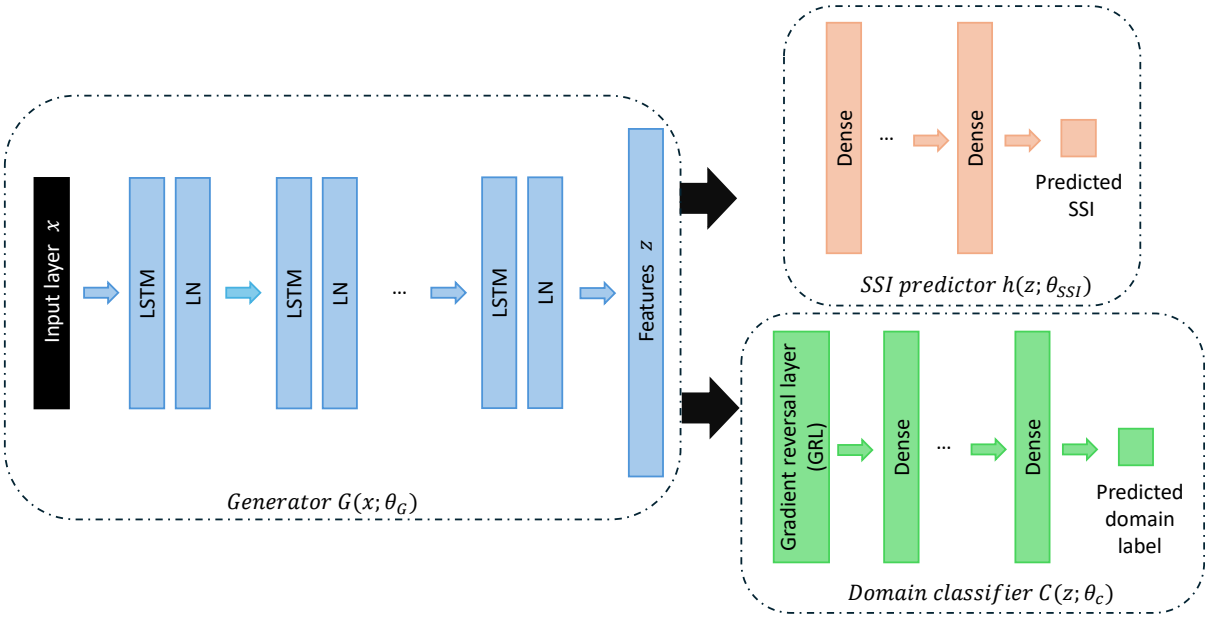


Figure 5.2: Architecture of ADG model. The generator is composed of a sequence of LSTM and LN layers; the SSI predictor and domain classifier are fully connected neural networks; a GRL is used to reverse the direction of the gradients during backpropagation.

mapping parameters θ_G are adjusted to both minimize the SSI prediction loss and maximize the domain classification loss (Eq. (5.10)). The parameter λ is a weighting coefficient used to control the trade-off between the losses [107].

$$\hat{\theta}_C = \text{argmax}_{\theta_C} E(\hat{\theta}_G, \theta_C, \hat{\theta}_{\text{SSI}}) \tag{5.9}$$

$$(\hat{\theta}_G, \hat{\theta}_{\text{SSI}}) = \text{argmin}_{\theta_G, \theta_{\text{SSI}}} E(\theta_G, \hat{\theta}_C, \theta_{\text{SSI}}) \tag{5.10}$$

5.3.3 Invariant Risk Minimization model

The IRM model adopts the same architecture as the ADG model, with the primary difference being the exclusion of the domain classifier component. Specifically, the IRM consists of the generator G, made up of a series of LSTM and LN layers, which processes 60-second sequences of surface measurements as input. The output from the generator is then fed into the SSI predictor h (same as ADG), a fully connected neural network, which is responsible for predicting the SSI (see Fig. 5.2). An additional trainable variable β is introduced to represent the linear classifier p in Eq. (5.11) and to compute the regularization term. Finally, the IRM loss function is:

$$\sum_{i=1}^{N_S} L(h(G(x_{\mathscr{D}_S^i}, \hat{\theta}_G), \hat{\theta}_{\text{SSI}}), f(x_{\mathscr{D}_S^i})) + \alpha \|\nabla_{\beta|\beta=1.0} L(\beta . h(G(x_{\mathscr{D}_S^i}, \hat{\theta}_G), \hat{\theta}_{\text{SSI}}), f(x_{\mathscr{D}_S^i}))\|^2 \tag{5.11}$$

Where L is the MSE loss, $x_{\mathscr{D}_S^i}$ is the data from the i-source domain, $f(x_{\mathscr{D}_S^i})$ represent the true SSI of $x_{\mathscr{D}_S^i}$, and α is a penalty weighting coefficient.

5.4 Experiments

In this section, the results of training a machine learning model to predict the SSI using 60-second sequences of surface measurements as input are presented. The model is trained using different training wells (source domains) and tested on completely different wells (target domain) in four different approaches, depending on the availability of source and target data:

- **Baseline model**: A traditional deep learning model consisting of a single component that takes the 60-second sequences of surface measurements as input and predicts the SSI as output (see Section 5.3.1).
- **ADG model**: As explained in Subsection 5.3.2, the model consists of three components: a generator, SSI-predictor and domain classifier.
- **IRM model**: This generalization technique incorporates a regularization term into the loss function. It processes 60-second sequences of surface measurements as input for the generator, and the output is then used as input to the SSI predictor. The architecture is illustrated in Subsection 5.3.3.
- **TL**: In contrast to the previous techniques, TL is applied during the testing phase. Once a model is trained (either the ADG, IRM, or baseline model), if labeled surface data sequences from the test well are available during testing, TL technique can be applied to improve the model's performance.

5.4.1 Data processing

For training, the first three wells, in Table.3.1, which originate from the same field and represent a substantial number of sequences, are selected for the training process. The last three wells are reserved for testing. The remaining wells (4, 5, and 6) are used alternatively for training and validation (see Section 5.4.2).

5.4.2 Grid search for hyperparameters tuning

To tune the model hyperparameters, a grid search method is employed with the ADG model. This approach involves exhaustively searching through a specified subset of the hyperparameter space for the training algorithm. The hyperparameters evaluated include the regularization parameter for the generator (covering bias, kernel, and recurrent regularization), the number of hidden layers in the generator, and the loss function weighting coefficient λ for the ADG (Eq. (5.5)). The potential values for the regularization coefficient are $(10^{-3}, 10^{-4}, 10^{-5})$, the number of hidden layers in the generator are set to $(4, 6, 8)$, while the weighting coefficient λ varies among $(1, 10, 100, 1000)$. In our experiments, validation data is used to determine the optimal hyperparameter values. Since our goal is to develop a model that generalizes effectively across sources, validation data is selected from wells that are entirely distinct from those used in training. Therefore, in each training session, from the first six wells listed in Table 3.1, four wells are used for training and the remaining two for validation. The last three wells (from two different rigs) are reserved for testing after the hyperparameters have been selected. Selecting the two wells for validation among the first six wells is challenging, as this choice significantly

influences both model performance and hyperparameters selection: when wells with characteristics similar to the majority of training wells are chosen, the SSI validation error tended to be low. In contrast, choosing wells with different characteristics resulted in a higher validation error. To address this, three different validation data cases are tested, as outlined in Table 5.1. In each case, the first three wells in Table 3.1, which are from the same field, are fixed as training data, and wells 4, 5, and 6, from 3 different rigs, are used alternately to select two wells for validation in each scenario.

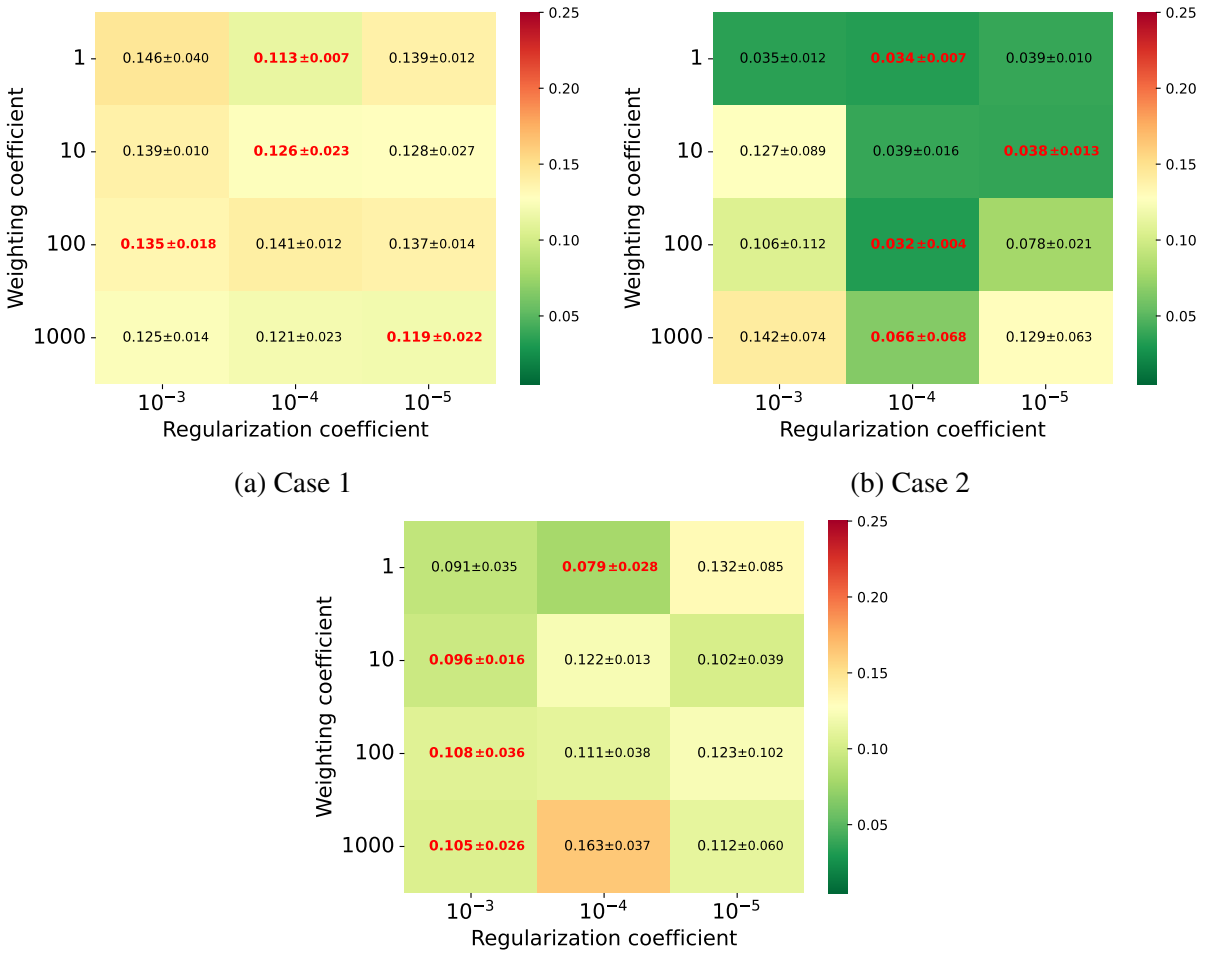
Table 5.1: Validation data selection for the three tested cases.

	Well 4	**Well 5**	**Well 6**
Case 1	**Validation data**	Training data	**Validation data**
Case 2	**Validation data**	**Validation data**	Training data
Case 3	Training data	**Validation data**	**Validation data**

5.4.3 Regularization coefficient

Due to limited computational capacity, hyperparameters are not all optimized simultaneously. Instead, the number of hidden layers in the generator is fixed at 6 while varying the regularization coefficient and the weighting coefficient for the ADG across the different validation data cases described in Table 5.1. The architectures for the domain classifier and SSI predictor are fixed at 5 dense layers, each containing $[60, 40, 20, 10]$ neurons for the first four layers, and 4 neurons (representing the number of training wells) and 1 neuron (for SSI) for the last layers, respectively. The generator architecture comprised 5 LSTM layers combined with 5 LN layers (including 2 input layers, 6 hidden layers, and 2 output layers), with 64 neurons per layer and bias, kernel, and recurrent regularization applied uniformly across all layers. Each architecture search is conducted for 500 epochs, utilizing a constant learning rate of 10^{-3} on an Nvidia GPU. For each combination of hyperparameters, and for each validation data case, the model is trained using three different initializations. For each initialization, the trained model is used to calculate the MSE and normalized DTW (see Section 4.8.2). The average of the three initializations is then computed.

Fig. 5.3 illustrates the average MSE on validation data across the three tested initializations for each of the three cases. As we can see, results vary across cases, with case 2 exhibiting the lowest MSE validation error. This outcome is likely due to the training dataset's sequence configuration, as for case 2, wells 4 and 5 are selected for validation, while well 6, with the highest sequence number, is included in the training dataset. Fig 5.4 displays the mean MSE and normalized DTW on validation data, calculated over the three tested cases. Notably, the error variations remain consistent whether using MSE or DTW. The results indicate that the lowest validation SSI prediction error is attained with a regularization coefficient of 10^{-4} for most of the tested weighting coefficients. Therefore, this hyperparameter is set at 10^{-4} for the remainder of this chapter.

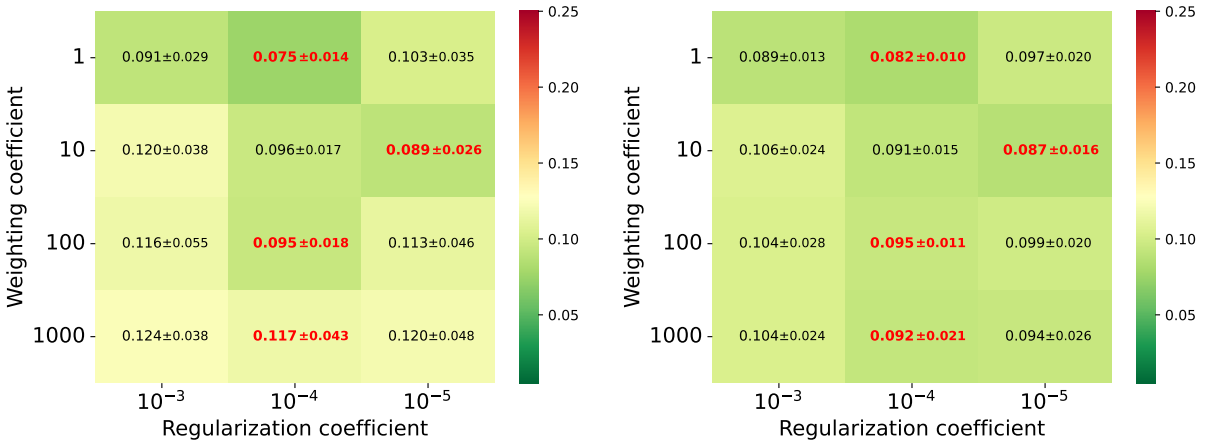


(a) Case 1

(b) Case 2

(c) Case 3

Figure 5.3: Average MSE of SSI validation data predictions for the three tested cases (Table. 5.1) with varying regularization and weighting coefficient over three different intializations: The ADG model is trained for each combination of hyperparameters using three distinct initializations. The validation SSI error is calculated for each one, and the average error is then computed across the three initializations.



(a) MSE (b) DTW

Figure 5.4: Average MSE and normalized DTW of SSI validation data predictions for the three tested cases with varying regularization and weighting coefficient in the ADG model.

5.4.4 Number of generator hidden layers and weighting coefficient

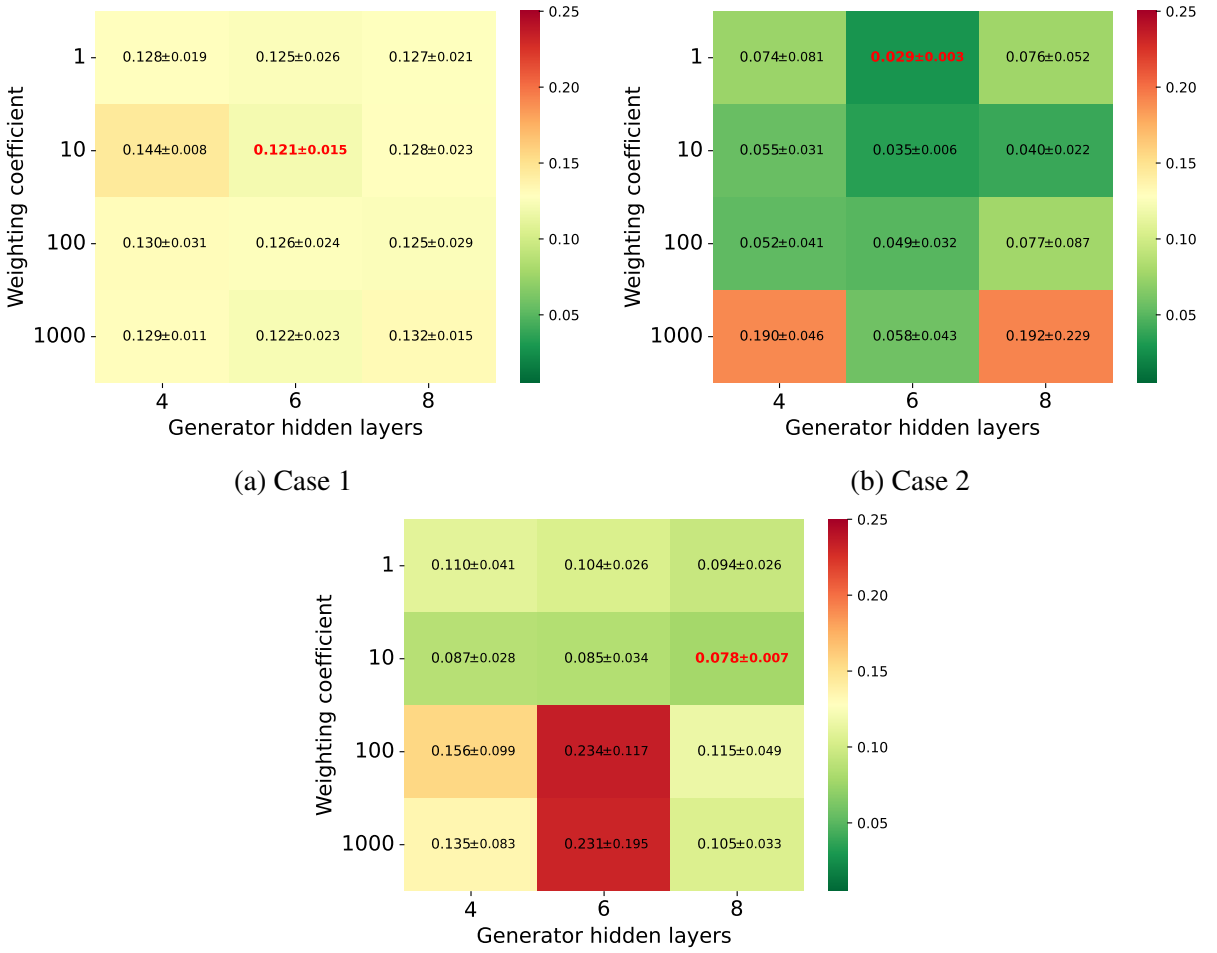
To select the optimal number of hidden layers for the generator and the weighting coefficient λ (see Eq. (5.8)), a grid search is conducted using three different initializations across the three validation cases (see Table. 5.1), following the same approach as for the regularization coefficient. The tested values for the generator's hidden layers are $(4,6,8)$ and for the weighting coefficient λ is set to $(1,10,100,1000)$. The optimal parameters are chosen based on the SSI validation error. Similar to the regularization coefficient selection, Fig 5.5 shows the average MSE on validation data across the three initializations for each case, while Fig 5.6 presents the mean MSE and normalized DTW on validation data, averaged across the three cases. The hyperparameter configuration with the lowest SSI validation error is chosen, which includes 6 hidden layers for the generator and a weighting coefficient of 10. These hyperparameters are fixed for the remainder of this chapter.

Remark: For the penalty weighting coefficient α of the IRM model, five different values are tested [0.01, 0.1, 1, 10, 1000] and selected the model with the lowest SSI prediction error on validation data, which occurred with a penalty coefficient of 1. This IRM model is then used for comparison with the ADG and baseline models for the rest of this chapter.

5.5 Results and discussions

5.5.1 Comparison of ADG, IRM, and baseline models

With the selected hyperparameters, a comparison is made between the ADG and IRM techniques applied to time series for SSI prediction against the baseline model (Section 5.3.1). For training, the first six wells are designated as training data and the last three as test data (Table 5.1), with 10% of the training data reserved for validation. Each model is trained for 1000 epochs, with a batch size of 2048, and a fixed learning rate of 10^{-3}, and the model that achieves the lowest SSI validation MSE is retained. The training is repeated five times, each time with a different initialization, and the average test SSI prediction error is calculated across



(a) Case 1

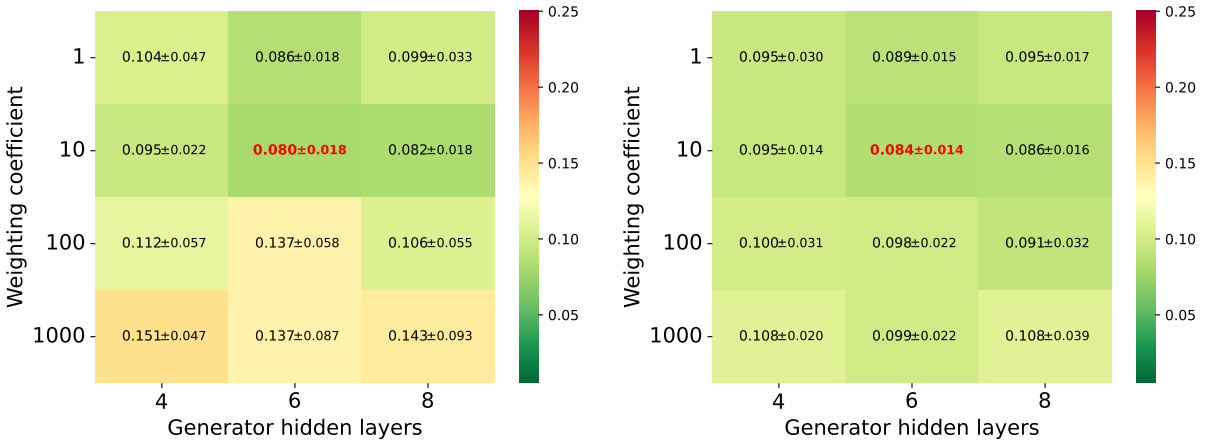
(b) Case 2

(c) Case 3

Figure 5.5: Average MSE of SSI validation data predictions for the three tested cases with varying generator hidden layer number and weighting coefficient over three different intializations: The ADG model is trained for each combination of hyperparameters using three distinct initializations. The validation SSI error is calculated for each one, and the average error is then computed across the three initializations.

the five runs. The ADG model requires about six hours of training, whereas the IRM and baseline models each take approximately four hours.

Table. 5.2 shows the normalized DTW results for the three test wells, over the five runs, for the ADG, IRM and baseline models. The ADG and IRM models demonstrate better generalization compared to the baseline, achieving improvements of 10.86 % and 8.42 %, respectively. However, the ADG shows slightly better results than the IRM. Fig. 5.9 and Fig. 5.10 illustrate examples of true and predicted SSI sequences for the three tested models across the three test wells. As shown in Fig. 5.8a and Fig. 5.8b, all models generally perform well in predicting SSI for most sequences. However, there are also sequences where they all fail to capture SSI variations, such as in Fig. 5.9a, where they overestimate the SSI, and Fig. 5.9b, where they underestimate the SSI. Notably, in some cases (Fig. 5.10), the ADG and IRM provide more accurate SSI predictions than the baseline. This indicates that the generator effectively projects the training data into a domain-invariant feature space, whether with ADG or IRM, enabling the



(a) MSE (b) DTW

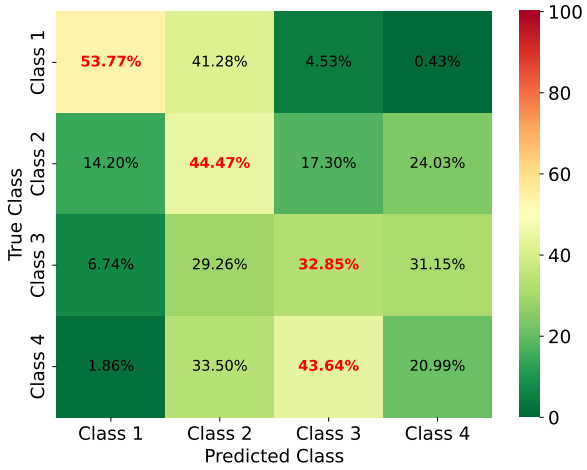
Figure 5.6: Average of the MSE and normalized DTW of SSI validation data predictions for the three tested cases with varying generator hidden layer number and weighting coefficient for the ADG model.

SSI predictor to make more reliable predictions using the newly projected features. Fig. 5.10b and Fig. 5.10c show sequences where the ADG model predicts the SSI with slightly greater accuracy than the IRM model, with both models consistently outperforming the baseline.

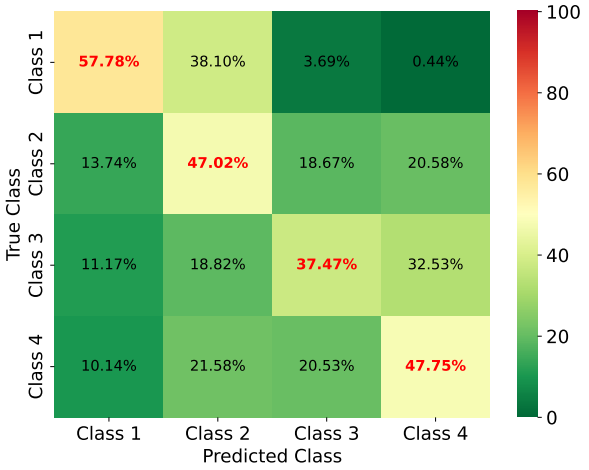
To further evaluate the results, Fig. 5.7 presents the confusion matrices for the three tested models, averaged over the three test wells. The SSI is classified into four distinct categories based on the severity of stick-slip: sequences without stick-slip (SSI $\in [0, 0.3[$) are assigned to the first class; sequences with low stick-slip (SSI $\in [0.3, 0.5[$) fall into the second class; sequences with moderate stick-slip (SSI $\in [0.5, 0.7[$) are grouped into the third class; and sequences with severe stick-slip (SSI ≥ 0.7) are classified into the fourth class. As shown, the confusion matrices reveal that all models correctly identify the majority of sequences for each class, except for the last class, which corresponds to sequences with severe stick-slip, the most damaging scenario. The baseline model incorrectly assigns most of these sequences to the third class instead of the fourth, a misclassification not seen in the ADG and IRM models. The IRM and ADG models show similar performance, with the IRM model predicting SSI more accurately for sequences with low SSI, while the ADG model performs better for sequences with high SSI. The reasons for sequence misprediction are discussed in Subsection 5.6. In conclusion, the IRM and ADG models exhibit similar performance, with a slight advantage for the ADG, but both outperform the baseline model.

5.5.2 Transfer learning application

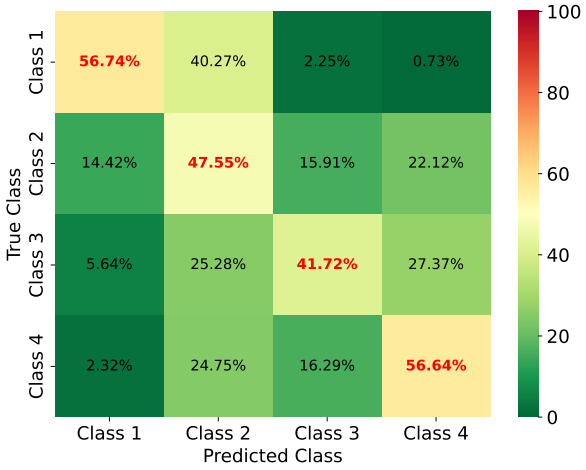
To enhance model performance on a specific target well, a TL technique can be applied if a small amount of labeled surface drilling data for that well is available (see Section 4.3). In our case, the contribution of TL is tested on each of the three test wells using the first 10 % of labeled surface drilling data with both, the ADG and baseline models. The source model, either the trained ADG or baseline model, is then fine-tuned to adapt to the target well by retraining only the weights and biases of the first two layers in the generator and SSI predictor for the ADG model, and the first two layers in the baseline model. The same optimizer and learning



(a) Baseline

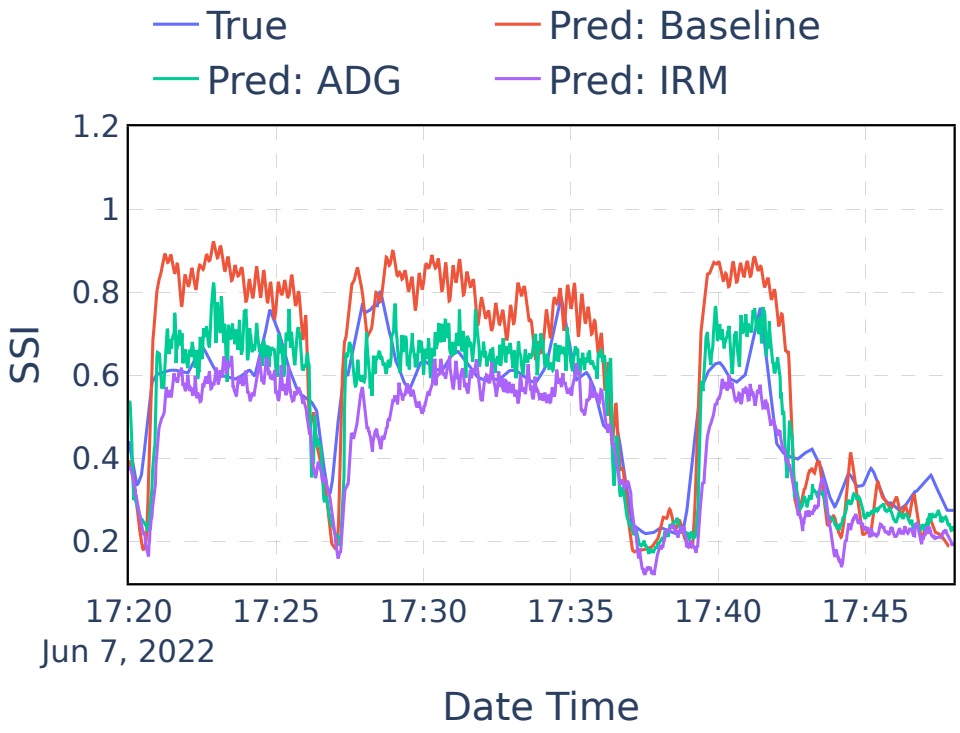


(b) IRM

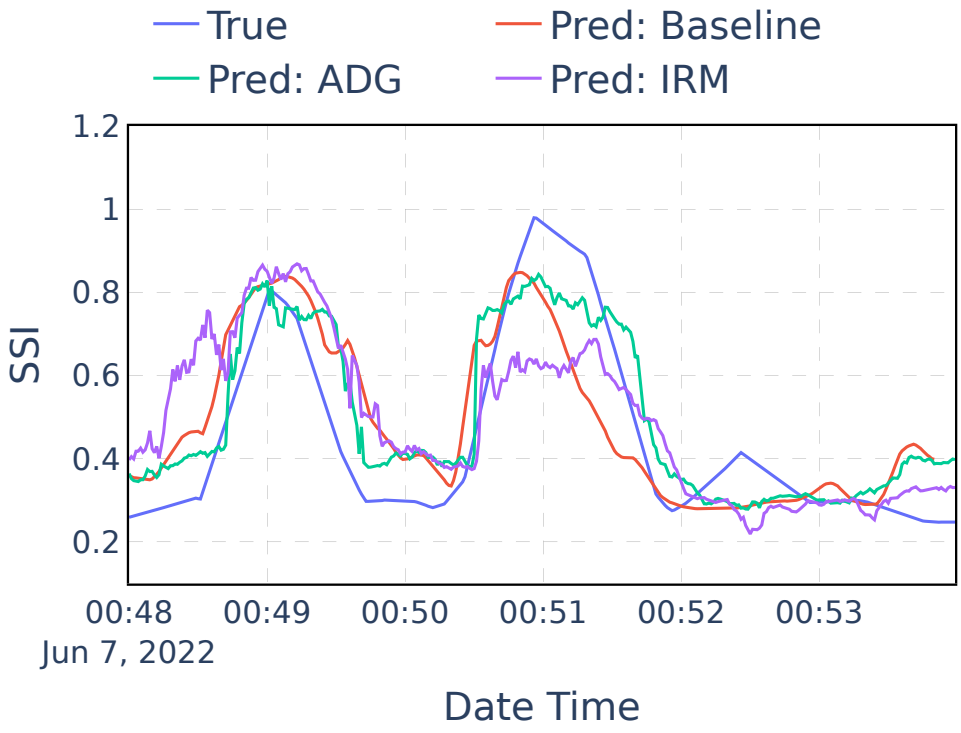


(c) ADG

Figure 5.7: Confusion matrices illustrating the performance evaluation of the baseline, IRM and ADG models over the three test wells. The true and predicted SSI values are classified into four categories: sequences with SSI $\in [0, 0.3[$ are assigned to the first class; SSI $\in [0.3, 0.5[$ are classified into the second class; if the SSI $\in [0.5, 0.7[$ belong to the third class; and sequences with SSI ≥ 0.7 are assigned to the fourth class.

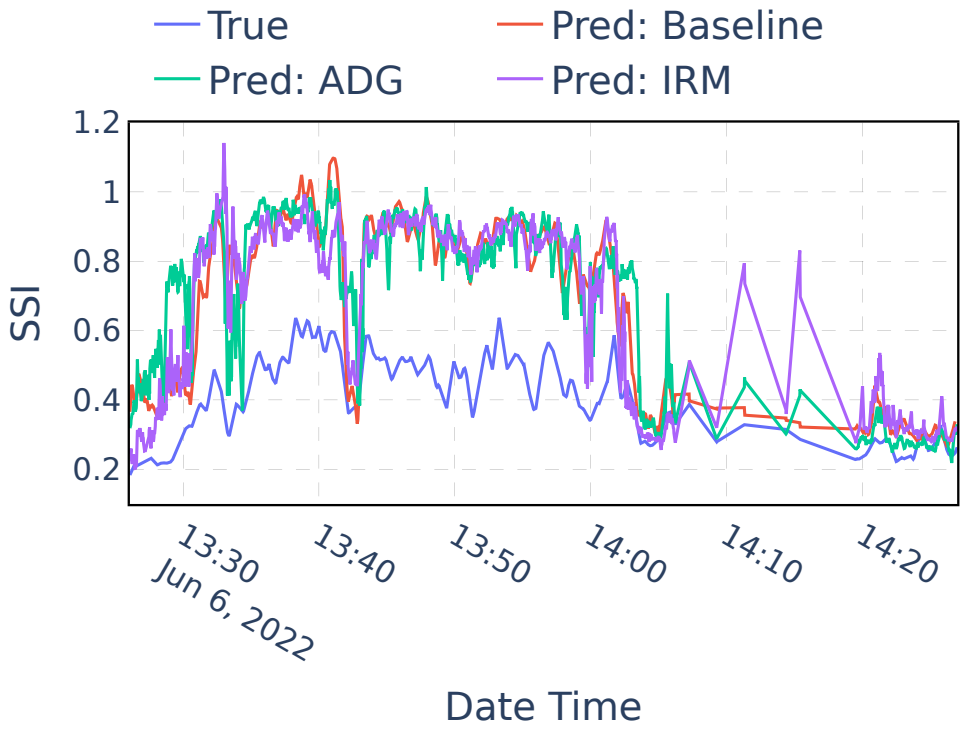


(a) Sequence 1

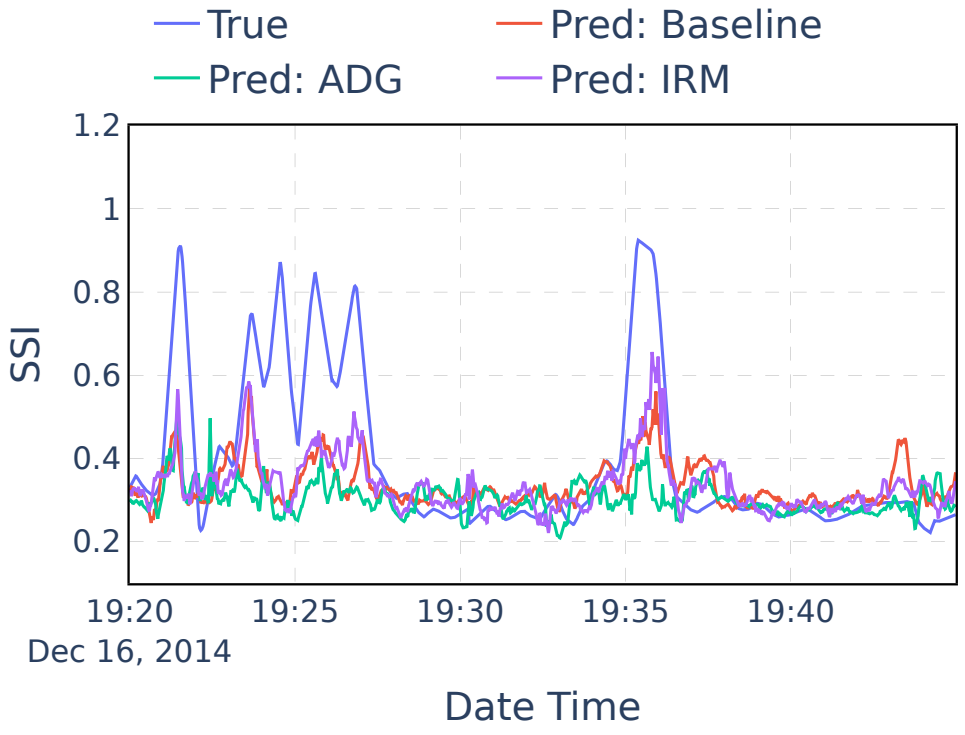


(b) Sequence 2

Figure 5.8: True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. In the two sequences, all models accurately predict the SSI.

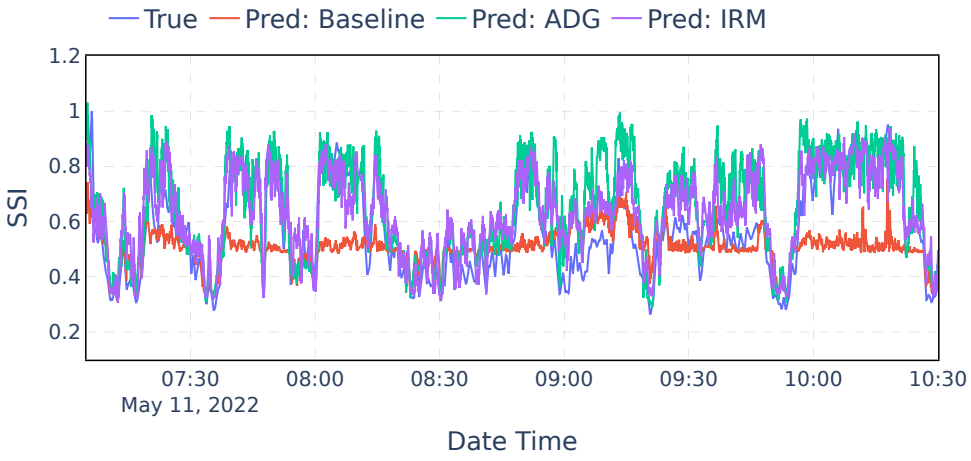


(a) Sequence 3

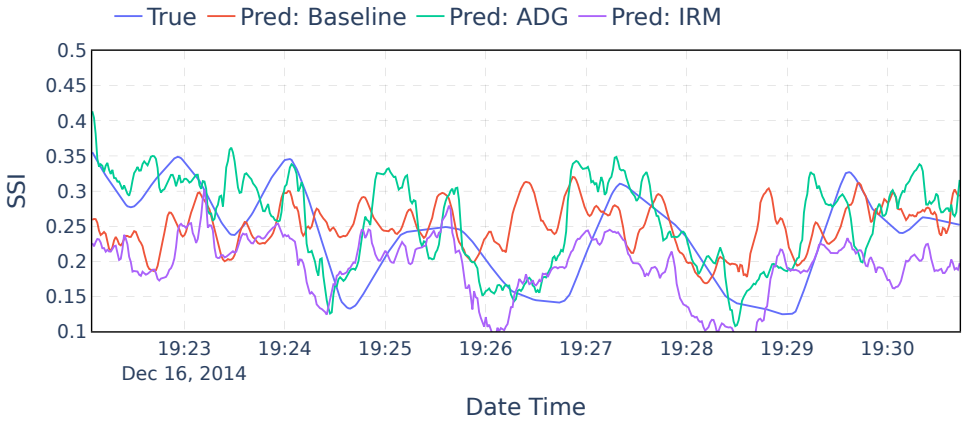


(b) Sequence 4

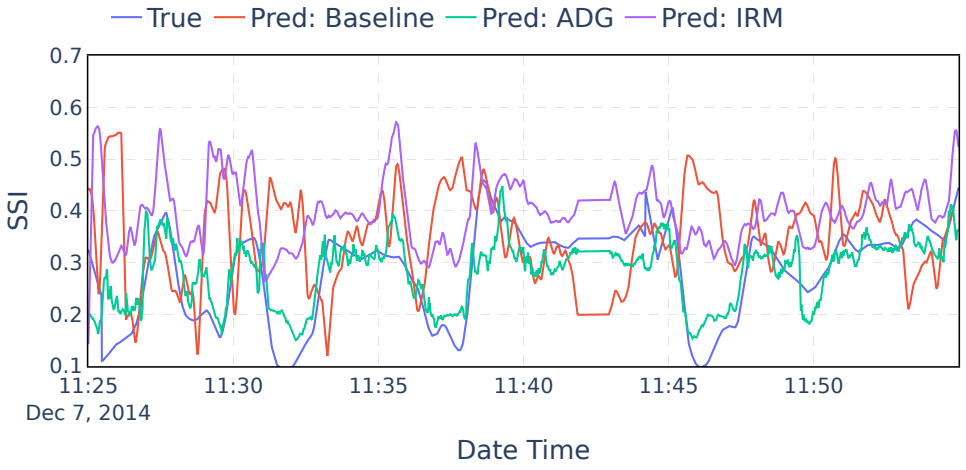
Figure 5.9: True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. In the two sequences, all models fail to capture the variations in SSI.



(a) Test well 1



(b) Test well 2



(c) Test well 3

Figure 5.10: True and predicted SSI values over time for the three trained models: baseline, IRM, and ADG, based on sequences from the three test wells. For the three test sequences, the ADG and IRM provide more accurate SSI predictions than the baseline. The last two sequences the ADG model predicts the SSI with slightly greater accuracy than the IRM model.

Table 5.2: Normalized DTW of the trained ADG, IRM and baseline models on the three test wells: Each model is trained with five different initializations, and the error is reported as the average across these five runs.

	Baseline	ADG	ADG vs Baseline (%)	IRM	IRM vs Baseline (%)	ADG vs IRM (%)
Test well 1	0.136	0.122	**10.29**	0.120	**11.76**	**-1.66**
Test well 2	0.086	0.075	**12.79**	0.080	**6.97**	**6.25**
Test well 3	0.107	0.097	**9.34**	0.100	**6.54**	**3.00**

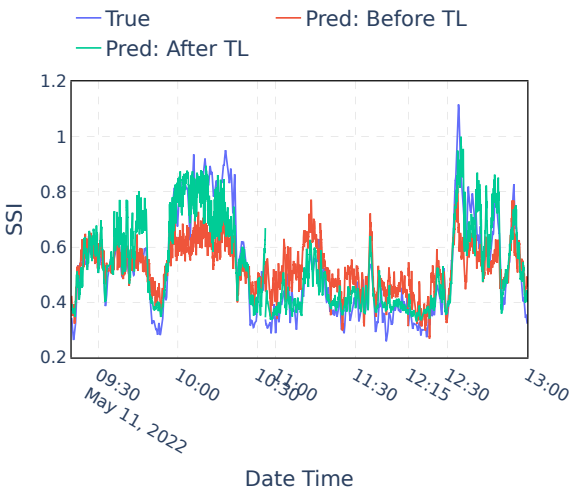
rate as in the source model training are used, but with a significantly reduced number of epochs, resulting in a lighter computational load, with a retraining time of approximately 1.5 minutes.

Table. 5.3 presents a comparison of model performance before and after applying TL for both ADG and baseline models. For each test well, the source model (either ADG or baseline) is partially retrained using the first 10 % of labeled surface drilling data. The results in the table show that TL enhances model performance in both ADG and baseline models, with a decrease in normalized DTW following retraining. Additionally, this TL approach is efficient in terms of time and requires only a small amount of data compared to what would be needed to train a new model.

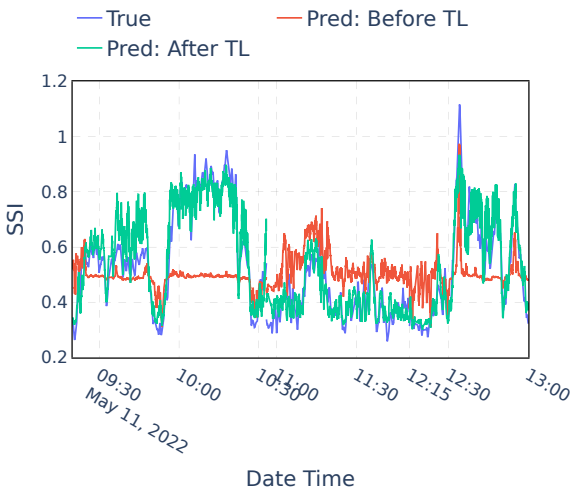
Table 5.3: Comparison of model performance, for both ADG and baseline models, before and after applying TL, using normalized DTW as the evaluation metric across the three test wells: Each source model undergoes partial retraining with the first 10 % of labeled surface drilling data from each test well.

	ADG			Baseline		
	Without TL	**With TL**	**%**	**Without TL**	**With TL**	**%**
Test well 1	0.122	0.095	**22.13**	0.136	0.110	**19.11**
Test well 2	0.075	0.058	**22.66**	0.086	0.06	**30.23**
Test well 3	0.097	0.088	**9.27**	0.107	0.100	**6.54**

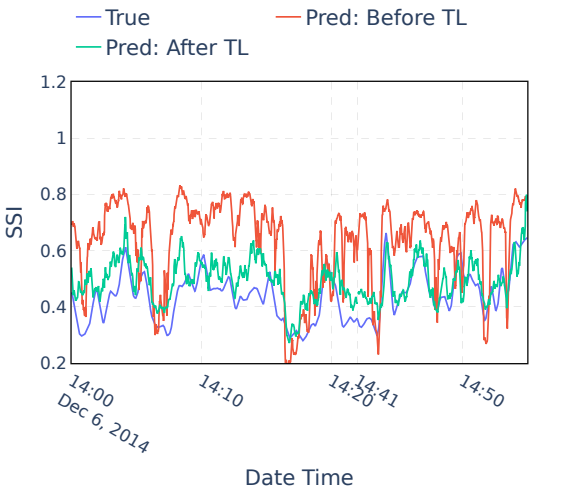
Fig. 5.11 shows the actual and the predicted values of SSI over time for the three test wells, comparing results with and without the application of TL. As we can see, after TL is applied, the new model predicts the SSI better than the source one for both models. Additionally, even after TL application, the ADG model outperforms the baseline model, with a lower normalized DTW across all three test wells (see Table. 5.3).



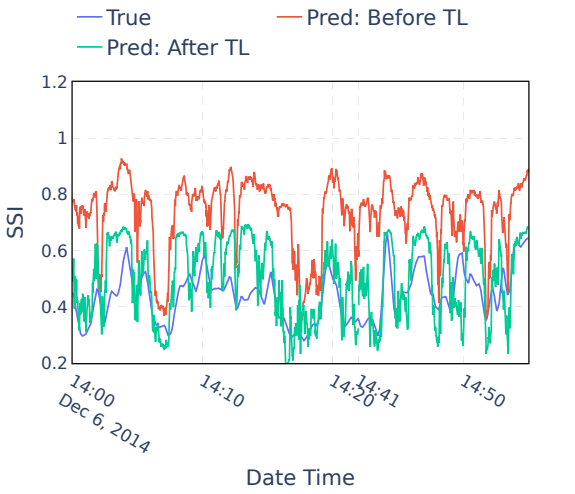
(a) Test well 1: ADG model



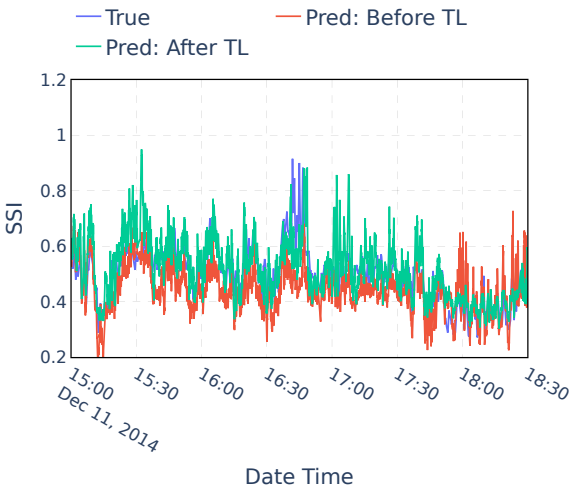
(b) Test well 1: Baseline model



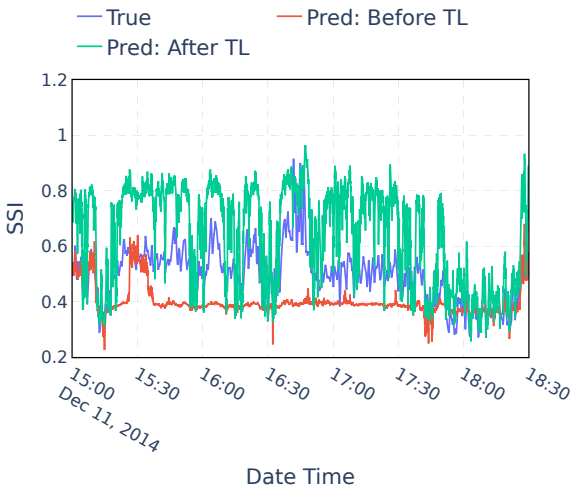
(c) Test well 2: ADG model



(d) Test well 2: Baseline model



(e) Test well 3: ADG model



(f) Test well 3: Baseline model

Figure 5.11: True and predicted SSI values over time, for both ADG and baseline models applied on sequences from the three test wells before and after the application of TL.

5.6 Causes of misprediction of SSI

The three models, baseline, ADG and IRM, before or after the application of TL, mispredict the SSI for certain sequences. In this section, some potential reasons for these mispredictions are analyzed, which generally cannot be addressed by domain generalization approaches.

5.6.1 Synchronization problem

In drilling operations, a time delay arises between surface and downhole measurements, and this delay is not constant but varies over time. It may stem from the propagation time required for downhole vibrations to reach the surface, or from environmental factors such as downhole temperature and pressure, which can affect the tool's internal time stamps [136]. As a result, one second may appear longer or shorter, and the sampling process can become nonuniform. Consequently, achieving perfect synchronization between surface and downhole measurements is especially challenging, particularly during long-duration drilling.

To evaluate our trained models, the SSI is predicted using surface measurements and the predictions are compared with the true SSI, which is calculated using synchronized downhole measurements (Eq. (3.1)). However, due to imperfections in synchronization, discrepancies can occur, leading to errors in evaluation, as illustrated in the following example. The synchronization issue is present in most domains and cannot therefore be addressed with domain generalization techniques. Fig. 5.12 shows a long test sequence that exhibits synchronization issue between surface and downhole data, with the surface data being significantly shifted. In this sequence, the 60-second red and blue sequences are used to predict their SSI using the trained ADG model. The model mispredicts the SSI of the red sequence, where the predicted SSI is high (0.77), due to surface torque variations, while the true SSI, calculated from the mismatched downhole measurements, is much lower (0.1) due to the absence of downhole vibrations. This misestimation is due to data misalignment and not to the trained model itself. For the blue sequence, the predicted SSI is 0.84 and the true SSI is 0.92, indicating that the model predicts the SSI fairly well. This is because surface torque variations are observed, likely caused by downhole vibrations, even though they do not correspond to the true sequence.

5.6.2 Undetected stick-slip

In some instances, downhole stick-slip can occur without generating vibrations detectable at the surface. This suggests that the downhole vibrations are attenuated as they propagated to the surface, due to friction and viscous damping, making them undetectable with low-frequency (1 Hz) surface measurements. This phenomenon is more commonly observed in lateral wells, particularly within the horizontal section. Fig. 5.13 illustrates the variation in surface torque and bit rotation speed during a test sequence in the horizontal section of a test well. As shown, severe stick-slip occurs at the bit (with a true SSI of 0.87), but no surface torque variation is observed. This probably explains why the trained model fails to predict the SSI using 1-Hz surface data for this type of sequence (with a predicted SSI of 0.14).

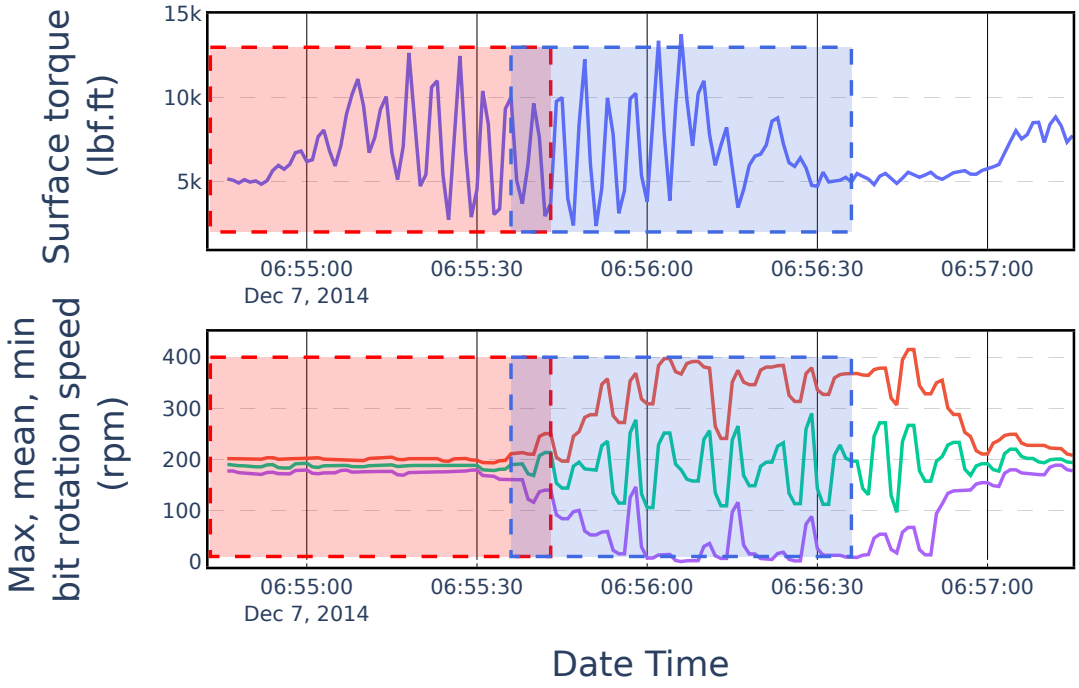


Figure 5.12: Variation in surface torque (surface measurement) and bit rotation speed (downhole measurements) for a test sequence. This sequence exhibits a synchronization issue between surface and downhole measurements. The predicted SSI for the 60-second red and blue sequences are high (0.77 and 0.84 respectively) due to torque variations. However, when using the mismatched sequence, the true SSI values for the red and blue sequences are 0.1 and 0.92 respectively. Consequently, the model mispredicts the SSI of the red sequence due to the synchronization problem.

5.6.3 Mislabeled data

To label surface data sequences, the true SSI is computed from downhole bit rotation speed measurements (Eq. 3.1). However, this labeling approach can sometimes produce misleading values. Specifically, certain sequences may display sudden variations (peaks) in the downhole speed that are not actual stick–slip events, yet they still yield a high SSI. For example, Fig. 5.14 shows a sequence where a sharp fluctuation in downhole bit rotation speed produces a high true SSI of 0.72, even though no stick–slip occurs. In such cases, the predicted SSI can be much lower (0.21), which may actually reflect the absence of stick–slip. Nevertheless, for evaluation purposes, the prediction is compared against an incorrectly inflated true SSI. This discrepancy is not due to a weakness of the trained model but rather a limitation of the labeling process itself.

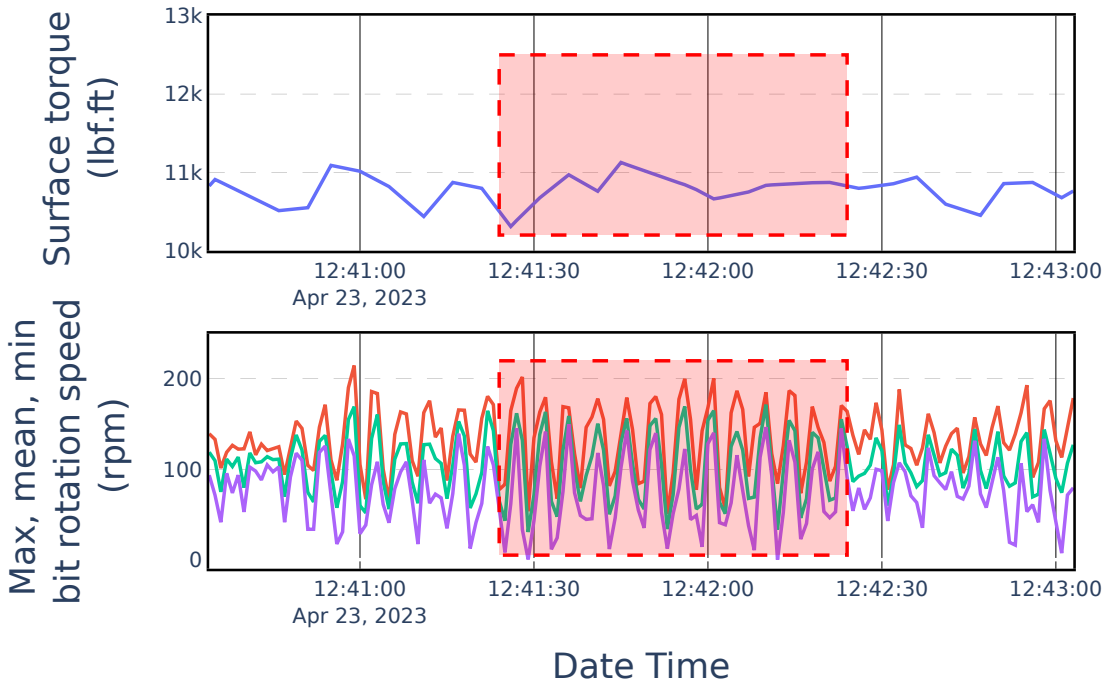


Figure 5.13: Variation in surface torque (measured at the surface) and bit rotation speed (measured downhole) during a test sequence. Stick-slip behavior occurs at the bit, but no surface torque variation is observed (due to friction). The predicted SSI using the trained ADG model for the 60-second red sequence is 0.14 and the true SSI is 0.87.

5.6.4 Domain mismatch

Another potential cause of ADG or IRM issues is domain mismatch, which arises when the model is not trained on enough data that adequately represents the target domain. Consequently, the model has difficulty generalizing effectively. To enhance generalization, the model may need to be trained on a more diverse dataset, which would improve its performance when exposed to new, unseen data.

5.7 Conclusion

In this chapter, the application of domain generalization techniques to time series data for predicting the SSI in drilling operations is explored. Our approach used 60-second sequences of 1-Hz surface drilling data as inputs. The performance of the ADG model with the IRM model, a traditional baseline model, and a TL approach are compared.

To optimize key parameters including the regularization coefficient, the architecture of the model, and the weighting coefficient for the ADG and the IRM models, a grid search methodology is employed. Our results demonstrate that ADG and IRM models enhance the generalization capabilities of the SSI prediction model, achieving improvements of 10 % and 8 % over the

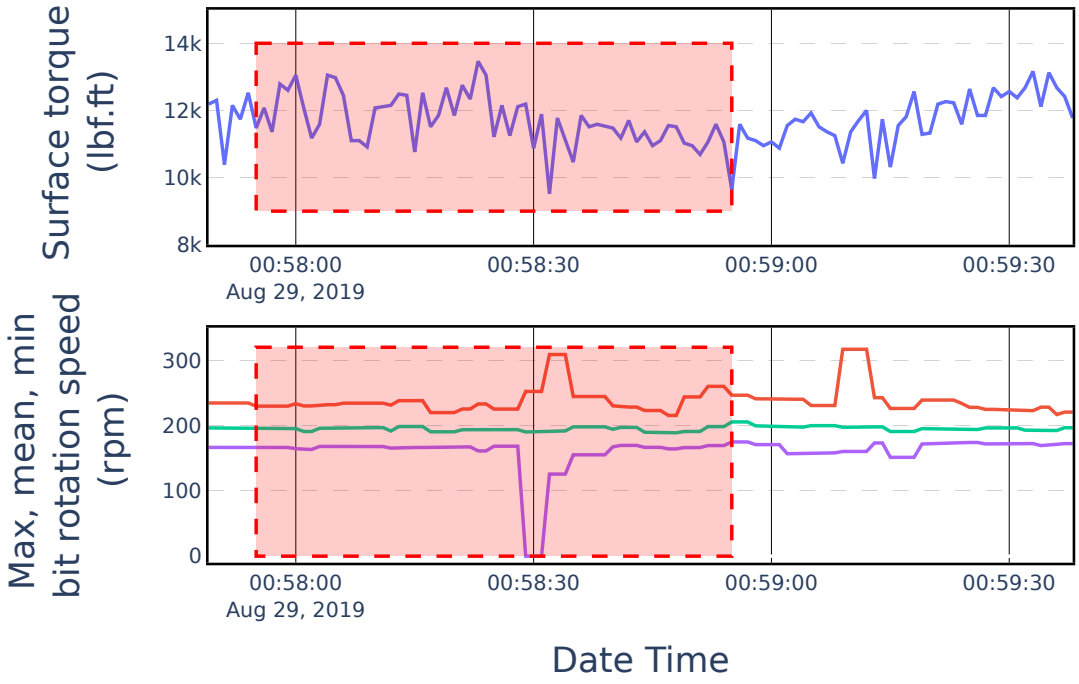


Figure 5.14: Variation in surface torque (measured at the surface) and bit rotation speed (measured downhole) during a test sequence. In this example, a peak in downhole bit rotation speed within the 60-second red segment results in a high true SSI (0.72), even though no stick-slip occurs. This makes the sequence mislabeled.

baseline model, respectively. Notably, the ADG model slightly outperforms the IRM model. Additionally, the implementation of TL on a pre-trained model (whether ADG or baseline) demonstrated improved performance. Even after TL application, the ADG model outperforms the baseline model, which highlights the generator's ability to map the training data sequences into a space where the SSI predictor cannot differentiate among them.

The proposed model can readily be used to detect stick-slip on new wells with no need for downhole sensors to be installed. In particular, severe events are detected around 60 % of the time (see Fig. 5.7). This constitutes a tool for drilling operators to monitor drilling vibrations in real-time, and reduce drilling costs.

Part II
Downhole velocity and friction coefficients prediction from low-frequency surface data

In the second part of this thesis, the focus shifts to the prediction of downhole bit velocity and the friction coefficients between the drill string and borehole, using low-frequency (1 Hz) surface measurements. While the first part of this work is devoted to predicting the Stick–Slip Index (SSI), the objective here is broader: to estimate the downhole velocity itself, from which the SSI can subsequently be deduced. Because direct downhole measurements of velocity and friction coefficients are rarely available in practice, owing to technical and economic constraints, developing reliable estimation methods based solely on surface data becomes essential.

To address this challenge, we investigate two approaches that combine data and physical modeling in the estimation process: Physics-Informed Neural Networks (PINNs) and the Projected Salted Kalman Filter (PSKF). PINNs, an offline method, rely on deep learning combined with governing equations to enforce consistency, whereas PSKF, a control-theory-based method, enables real-time estimation. Both methods leverage a simplified physical model but differ in how they incorporate physics and handle uncertainty.

This part of the thesis is organized as follows. Chapter 6 introduces the problem, outlines the challenges encountered in existing methods from the literature, presents our proposed solutions, describes the physical model employed, and discusses the hypotheses we impose. Chapter 7 focuses on PINNs and details their application to the drilling problem, including the training strategy, the use of transfer learning, and their evaluation on both simulated and field data. Chapter 8 then presents the Salted Kalman Filter (SKF) and its enhanced version, the PSKF, highlighting their real-time capability, the challenges addressed, and their validation on both simulated and field data. Finally, Chapter 9 provides a comprehensive comparison of both approaches, using simulated and field data from different wells. This includes a benchmark against a MATLAB toolbox method, considered state-of-the-art for simulated data, as well as a comparison with downhole measured data for field tests. Finally, it discusses their respective strengths, limitations, and perspectives for future improvements.

Chapter 6

Bit rotation speed and friction coefficients estimation - Introduction

Résumé du chapitre

Ce chapitre présente le problème traité dans la seconde partie de cette thèse : le développement d'une méthode en temps réel et généralisable pour l'estimation de la vitesse de rotation au fond du puits ainsi que des coefficients de frottement dynamique et statique, à partir de mesures de surface échantillonnées à basse fréquence (1 Hz) dans des puits horizontaux. Pour aborder ce problème, des modèles physiques simplifiés de la tige de forage sont utilisés et expliqués dans ce chapitre. À partir de ces modèles, deux approches distinctes, basées respectivement sur l'apprentissage automatique et la théorie du contrôle, sont appliquées à la même tâche d'estimation. Ces méthodes sont brièvement présentées ici et seront discutées plus en détail dans les chapitres suivants.

Chapter Summary

This chapter introduces the problem addressed in the second part of this dissertation: the development of a real-time, generalizable method for the estimation of downhole rotational speed and both dynamic and static friction coefficients using low-frequency (1 Hz) surface measurements in lateral wells. To tackle this problem, simplified physical models of the drill string are employed and explained in this chapter. Building on these models, two distinct approaches, rooted in machine learning and control theory are applied to the same estimation task. These methods are briefly outlined here and discussed in greater details in the following chapters.

6.1 Problem statement

The second part of this dissertation focuses on estimating bit rotation speed and friction parameters during drilling operations, using surface measurements, two key factors in improving drilling efficiency. Monitoring variations in bit rotation speed is crucial for achieving the optimal ROP and minimizing energy loss, thereby maximizing drilling efficiency and reducing operational time and cost. Furthermore, downhole measurements, including bit rotation speed, are not always available as BHAs are not always equipped with sensors. Usually when they do exist, they are not always available in real-time, but only collected after tripping operations [1]. Alternatively, it can be transmitted to the surface in real-time using pressure pulses in the drilling fluid, though this method has low data transmission rate and accuracy, limiting its uselfuness. Another option is using a wired connection integrated into the drill pipes, which provides high-speed real-time data transmission but is significantly more expensive due to the embedded transmission system. It is therefore very rarely installed.

Friction parameters also play a crucial role throughout the drilling process. As the drill string is tripped in or out, or rotated on- or off-bottom, the friction forces must be considered. Qualitative assessments and quantitative estimates of the torque, drag, vibration intensity, and drill ahead prediction are critically important in many phases of the well construction. During the design phase, these calculations are affected by the friction coefficients, which play an important role on the wellbore quality and in the drill string mechanics calculations [137, 138, 139]. Underestimating friction coefficients may lead to a significant underestimation of frictional drag resulting into a poor definition of the operation limits or to undrillable wellbore trajectories. Real-time estimated friction factor logs are used to identify hole cleaning problems, stuck pipe, differential sticking, formation changes and mud lubrication problems [140, 141]. Furthermore, the variation of the friction coefficients along the wellbore can generate torsional drill string vibrations [142, 143]. The dependency of the friction forces on the slip velocity, or rotation speed in the case of torsional dynamics, is known to be a source of stick-slip vibrations (see Section 3.1) [30]. Due to the cyclic nature of stick-slip, different problems can arise such as drill pipe fatigue problems [144], drill string components failures [144], and bit damage [145]. In addition to that, the estimation of friction coefficients is highly complex due to their dependencies on various factors such as temperature, drill string materials, rock formation, and drilling fluid. This complexity renders them one of the most uncertain factors in most mechanical computations concerning the drill string [146].

Given these challenges, our objective is to develop real-time general methods to estimate both downhole rotational speed as well as the static and dynamic friction coefficients between the drill string and the borehole using only surface measurements (surface torque and surface rotation speed) at 1 Hz, providing a practical, cost-effective alternative to downhole sensors.

6.2 Review

There is an extensive literature on the combined use of surface measurements and a simple model of torsional stick-slip oscillations to predict bit rotation speed and friction coefficients, such as models used for analysis and control, either distributed [147, 148, 149] or finite-dimensional [150, 151]. For friction coefficients estimation, torque and drag models, both

[1] Tripping out is the action of removing the drill string from the wellbore, e.g. to change a piece of equipment.

soft [152] and stiff-string approaches [153] are tested. However, the model must be calibrated before being used for each new well and each new estimation requires a manual trial-and-error procedure. Control configuration methods are developed in [154] in order to track a desired speed and to obtain the equivalent friction torques. Nonetheless, the evaluations are focused only on simulated data. A soft sensor to estimate the friction coefficients and angular velocity using only surface measurements while drilling off-bottom is presented in [155]. The method is extended for on-bottom drilling in [156]. Based on a distributed torsional model, it relies on linear observer design [157] to construct estimates of the unmeasured states and parameters. This yields a method implementable in real-time but that fails to converge in some cases, when linear correction terms and parameter update laws are insufficient. In Wu and Hareland [140], a Finite Element Analysis (FEA) approach is presented to calculate the friction coefficient and downhole WOB. The friction coefficient is determined using off-bottom data, and then the downhole WOB is estimated through the FEA model. However , this method predicts only the dynamic friction coefficient, neglecting the static friction component, and does not estimate the downhole rotational speed. As will appear, our approach adopts a different trade-off: we explicitly tackle the dynamic, nonlinear aspects of the problem, while neglecting the distributed behavior.

Parameter identification for nonlinear systems typically relies on optimization-based methods using least squares or related cost functions [158, 159], or on black-box structures [160]. For real-time applications, the Kalman filter and its variants [161, 162, 163] are widely used, but generally do not perform well in the presence of strong nonlinearities. Some proper observers with different orders, to predict the bit rotational speed, rock stiffness and torque-on-bit, in real-time are designed and presented by Sadeghi et al. [164]. However , the method is tested only on simulated data, and does not take into account friction forces between the pipe and the borehole. In fact, the drill string is assumed to be vertical and straight, which limits the applicability of the results to real drilling conditions.

Machine learning models for approximating nonlinearities have been proposed too. Such a data-driven approach predicting the friction coefficients using the measured top drive torque and angular velocity as input is developed in [165], where a neural network regressor is trained using a thousand of simulated training points with different ranges of depths and drill string dimensions. However, the trained model does not perform well on data outside of the training set, typically corresponding to new wells. To estimate the bit behavior, spectrogram analyses of surface and downhole measurements are proposed in [166]. Bayesian inference combined with deep neural networks for state estimation is proposed by Mestav et al. [167]. Zhang et al. [168] propose a machine learning model for real-time prediction of downhole rotation speed ranges using only surface-measured data, enhanced by an ensemble framework to improve the robustness of stick-slip detection and quantification. All input features such as rotary torque and top drive speed are collected at the surface, with the exception of the training labels, which consist of downhole rotation speed data acquired from Measurements While Drilling (MWD) tools [2] after the completion of a BHA run. The surface data is sampled at 2 Hz. For the downhole rotation speed range prediction, an XGBoost model is trained on historical datasets. The training labels include minimum, average, and maximum rotation speed values, and the model outputs the predicted downhole rotation speed range of the BHA, defined by the minimum and maximum downhole rotation speed. While purely data-driven models can face challenges related to large data requirements, limited physical interpretability, and generalization to unseen

[2]Measurements While Drilling (MWD) refers to the technology and tools used to collect real-time data from downhole sensors during drilling operations.

scenarios, they nonetheless provide valuable insights and serve as a strong foundation when combined with physics-based approaches.

To address these challenges, our goal is to develop a real-time generalizable method that integrates surface measurements with a physical model, specifically, the torsional drill string dynamics described in the next section, to predict downhole rotation speed and friction coefficients for lateral wells. This approach focuses on overcoming the limitations of previous work, particularly regarding model generalizability, handling of the nonlinearities, prediction of static and dynamic friction coefficients, and validation using field data.

6.3 Drill string torsional dynamical model and friction parameters

This section introduces the torsional dynamic model based on [69]. In view of estimating friction parameters and performing simulations, the model can be approximated as a series of n discrete masses, as depicted on Fig. 6.1. This lumped-parameter model is widely used to imitate drill string behavior and analyze stick-slip vibrations, offering a good balance between accuracy and computational efficiency [34, 169, 170, 171]. As noted by Volpi et al. [172], even a low number of degrees of freedom in this lumped parameter approach could successfully capture the overall behavior of a continuous model discretized with the finite element method.

6.3.1 Lumped torsional drilling model

We assume that the drill string is in torsion only (i.e. we ignore the axial and lateral dynamics), rotated from the surface by a motor with rotary speed ω_t and inertia I_{Top}. At the bottom extremity, torque created by the rock cutting process (TOB) is applied to the drill string, and is considered as a function of the bit rotation speed ω_b. Along the well trajectory, the drill string may be in contact with the borehole. Each contact force creates a friction torque T^f applied to the drill string. This torque is described by a friction law that depends on the drill string rotary speed at the contact point. In this lumped model, each mass behaves as a torsional pendulum with a linear spring of torsional stiffness K_i, a torsional damping C_i, and a mass inertia I_i, which are connected to the top drive with inertia I_{Top} [173]. The expressions for I_i, K_i, and C_i are derived by discretizing each mass i into smaller segments (elements) and computing the equivalent lumped parameters by aggregating the contributions from all segments associated with that mass. The contributions of additional components, such as stabilizers and shock absorbers, are explicitly considered in this process. Hence, the lumped parameters for each mass i are given by:

$$I_i = \sum_{j=1}^{n_i} \rho_i \, J_j \, L_j, \quad C_i = \sum_{j=1}^{n_i} 2 \, \pi \, \mu \, L_j \, \frac{r_{j_o}^3}{r_{j_o} - r_{j_i}}, \quad K_i = \frac{1}{\sum_{j=1}^{n_i} \frac{L_j}{G J_j}}, \quad J_i = \frac{\pi}{2} \left[r_{j_o}^4 - r_{j_i}^4 \right] \tag{6.1}$$

where, n_i is the number of segments used to discretize mass i, ρ_i is its density, μ is fluid viscosity, r_{j_o} and r_{j_i} are the outer and inner radius of segment j, L_j is the length of segment j, J_j is its polar moment of inertia, and G is shear modulus.

Define $\vec{\phi} = (\phi_{\text{Top}}, \phi_1, \ldots, \phi_n)^T$ the angular displacement vector for the n masses from the

surface to the bit. As a result, we have the following matrix equation:

$$\mathbf{M}\ddot{\vec{\phi}} + \mathbf{C}\dot{\vec{\phi}} + \mathbf{K}\vec{\phi} = -\vec{T}^f - \vec{T}^b \tag{6.2}$$

where:

- $\mathbf{M} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the diagonal inertia matrix with entries I_i, $i = 0, ..., n$, with $I_0 = I_{Top}$.
- $\mathbf{C} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the diagonal damping matrix with entries C_i, $i = 0, ..., n$. We note that the drill string conventionally behaves elastically without internal damping. In this model, the damping terms come from external interactions as viscous mud-structure friction [69].
- $\mathbf{K} \in \mathbb{R}^{(n+1)\times(n+1)}$ is a block-diagonal stiffness matrix, whose block entries are the n sub-matrices $K_i \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ describing the stiffness of each element.
- T_i^f for $i \in [1, .., n]$ represents the friction torque for the i-th mass, and T^b represents the TOB due to the interaction between the rock and the bit. Therefore:

$$\vec{T}^b = (0, \ldots, 0, T_n^b)^T$$
$$\vec{T}^f = (0, T_1^f, \ldots, T_n^f)^T$$

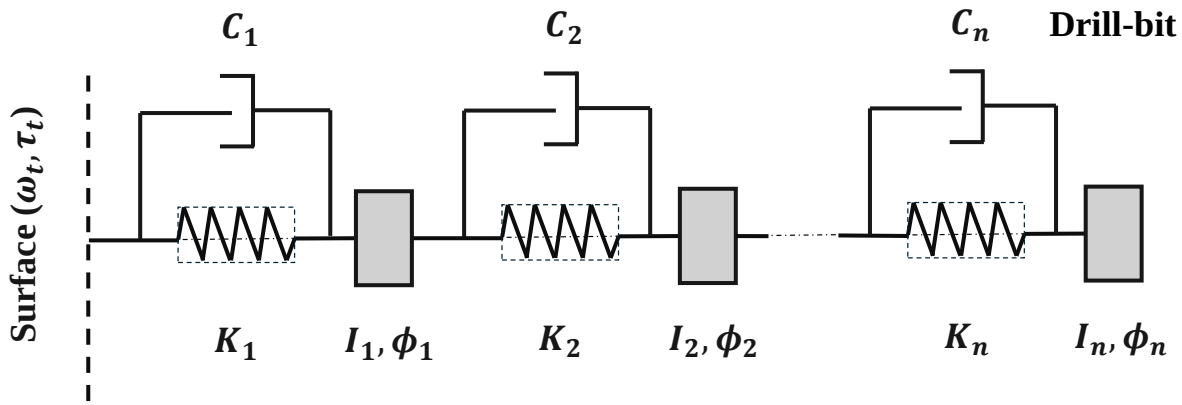


Figure 6.1: Drill string mass-spring system, where ω_t and τ_t denote the surface rotational speed and surface torque, respectively. Each mass behaves as a torsional pendulum characterized by a angular displacement ϕ_i, linear spring with torsional stiffness K_i, damping coefficient C_i, and rotational inertia I_i, computed using Eq. (6.1).

6.3.2 Friction model

Several nonlinear functions have been proposed to describe the variation of friction law as a function of the drill bit rotation speed ω_b and the drill string contact point rotary speed ω_i, respectively [174, 43]. Most of them are characterized by the presence of the so-called *Stribeck effect*, which describes the decrease in friction as the sliding velocity[3] increases. This behavior

[3]The sliding velocity refers to the relative tangential speed between two surfaces in contact.

is marked by a significant difference between the static and dynamic friction values [175], and can lead to instability or stick-slip vibrations. For this purpose, the friction law can be modeled as:

$$T_i^f = R_i F_i^N \Gamma(\omega_i, \mu_s, \mu_d) \tag{6.3}$$

where R_i is the drill string outer radius at the i-th contact point, $\omega_i = \dot{\phi}_i$ is the rotation speed at the node i, F_i^N is the normal contact force computed using a torque & drag model at the i-th contact point (see Section 4.5) [155, 69, 176, 71], μ_s is the static friction coefficient, where $\mu_s F_i^N$ represents the maximum friction force that must be overcome to initiate motion of the i-th mass. μ_d is the dynamic friction coefficient, also called the kinetic friction coefficient, and characterizes the friction force acting once relative motion has started. Γ, referred to as the dry friction function [3], captures the variation of the friction force with respect to the sliding velocity.

To calculate T_i^f for each mass i, the rotation speed ω_i of that mass is used. However, in calculating T_n^f for the last mass, it is assumed that the downhole motor is positioned just before the bit. Consequently, friction is considered to occur mostly on the surface of the last mass before the downhole motor, which rotates at velocity $\omega_n = \omega_b$, rather than on the surface moving with velocity $\omega_b + \omega_{\text{Motor}}$.

In this dissertation, two models of dry friction are used and presented in the following two subsections.

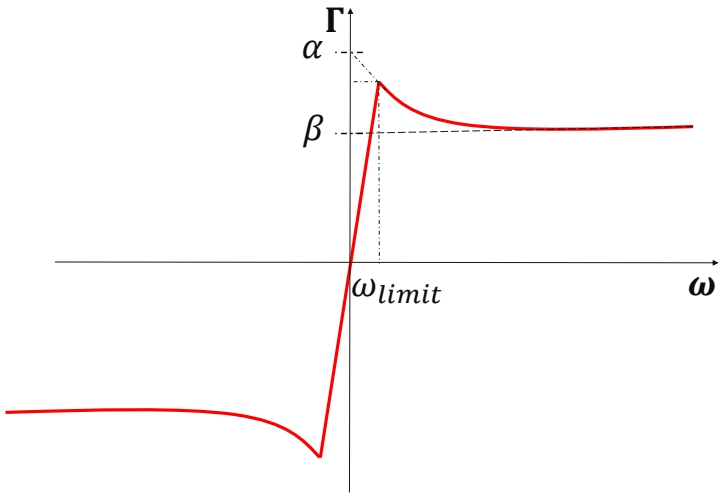
6.3.2.1 Smooth friction model

The smooth friction model, described in Fig. 6.2a, describes the dry friction function as a smooth single-valued function of ω on the whole domain. The Stribeck effect is modeled by a continuous exponential decrease of the friction coefficient during sliding. The dry function Γ of the smooth friction model can be written as follow:

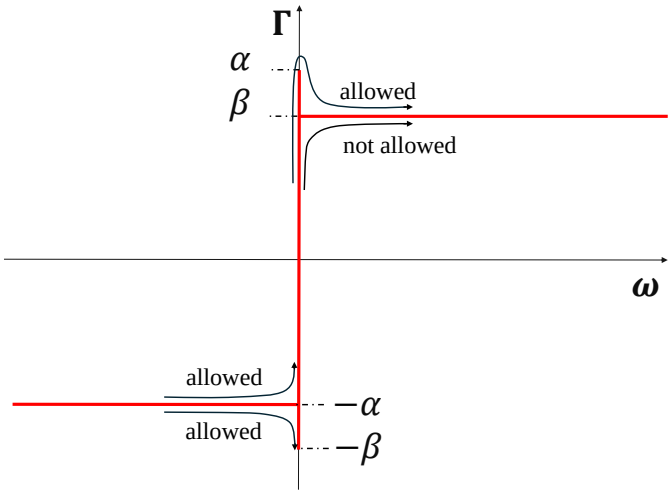
$$\Gamma(\omega, \alpha, \beta) = \begin{cases} \text{sign}(\omega)\left[\beta + (\alpha - \beta)\, e^{-|\omega|/\omega_c}\right], & \text{if } |\omega| \geq \omega_{\text{limit}}, \\ \dfrac{\omega}{\omega_{\text{limit}}} \Gamma(\omega_{\text{limit}}, \alpha, \beta), & \text{otherwise.} \end{cases} \tag{6.4}$$

where ω_c is a characteristic rotary speed, ω_{limit} is the limit rotary speed used to avoid the discontinuity of the law. This smooth model exhibits a steep slope for small velocity, reaching a peak value close to α when ω_{limit} is small. This upward ramp is followed by an exponentially decaying segment that asymptotically approaches a nearly constant value β.

The smooth friction model offers both advantages and limitations. Its use leads to a smooth ordinary differential equation for the system's dynamical equation Eq. (6.2). However , the disadvantages are twofold. First, it fails to accurately capture the stick phase, as the friction force becomes zero only at zero velocity. As a result, the drill string will always slip in the presence of an external force. Therefore, the stick phase can be approximated as the region along the upward ramp of the friction curve. Second, the steep slopes of both the upward and downward ramps lead to a highly stiff system of differential equations. This stiffness poses challenges for numerical integration and can make simulations computationally demanding [3].



(a) Smooth friction model



(b) Stiction friction model

Figure 6.2: Friction models [2, 3].

6.3.2.2 Stiction friction model

Unlike the smooth friction model, which can be interpreted as a regularized approximation of the stiction model, the stiction friction model replaces the upward and downward ramps with a set-valued function at $\omega = 0$, as illustrated in Fig. 6.2b. The Stribeck effect is then captured by explicitly modeling the difference between the static and dynamic friction coefficients [177, 178].

The most distinctive feature of the stiction friction model is that the transition from slip to stick is permitted for any value of the friction force within the interval $\left[-\alpha F_i^N, \alpha F_i^N\right]$. In contrast, the transition from stick to slip can only occur when the external force F_{ext}[4] exceeds the static friction threshold $F_s = \alpha F_i^N$. This mechanical interpretation, however , may conflict with

[4]In our case, F_{ext} consists of the torsional forces exerted by the sections upstream and downstream.

the mathematical framework in which such systems are described as differential inclusions [3]. These admissibility conditions are illustrated in Fig. 6.2b, where arrows labeled *allowed* and *not allowed* indicate the permitted and forbidden transitions. If the system with the stiction friction model is interpreted as a differential inclusion, we obtain:

$$\Gamma(\omega,\alpha,\beta) = \begin{cases} \beta & \text{if } \omega > 0 \\ [-\alpha,\alpha] & \text{if } \omega = 0 \\ -\beta & \text{if } \omega < 0 \end{cases} \tag{6.5}$$

The solution of this differential inclusion (6.5) exists as $\Gamma(\omega,\alpha,\beta)$ is upper semi-continuous, non-empty and bounded and has a closed convex image using Theorem 1.

Theorem 1 (Existence of solution of a differential inclusion)**.** *Let F be a set-valued function. Assume that F is upper semi-continuous and that $F(t,x)$ is closed, convex, and bounded for all $t \in \mathbb{R}$ and $x \in \mathbb{R}^n$. Then, for each $x_0 \in \mathbb{R}^n$, there exists a $\tau > 0$ and an absolutely continuous function $x(t)$ defined on $[0,\tau]$, which is a solution of the initial value problem*

$$\dot{x}(t) \in F(t,x(t)), \quad x(0) = x_0.$$

However , this formulation is not mechanically consistent, as a transition from stick to slip is possible for $|F_{\text{ext}}| < F_s$. This leads to a lack of uniqueness in the solutions. Specifically, when the system starts from a zero-velocity initial condition ($\omega(0) = 0$) and the external force satisfies $F_d < |F_{\text{ext}}| < F_s$, with $F_d = \beta F_i^N$, two outcomes are possible: the system may either remain stuck or begin to slip. Both scenarios are valid solutions to the dynamic equations, despite representing different physical behaviors.

In conclusion, the stiction friction model cannot be treated as a differential inclusion, as its solution concept conflicts with the mechanical understanding of friction. As a result, systems modeled with stiction friction are generally not suitable for integration using standard event-driven or time-stepping methods.

6.3.3 Torque-On-Bit model

For the Torque-On-Bit (TOB) variation, the mathematical formulations vary from one author to another [179, 180, 181]. In our model, we adopt an exponentially decaying law:

$$T_n^b(\omega) = \begin{cases} \text{sign}(\omega)\left[\tau_d + (\tau_s - \tau_d)\, e^{-|\omega|/\omega_c}\right], & \text{if } |\omega| \geq \omega_{\text{limit}}, \\ \text{sign}(\omega)\left[\dfrac{|\omega|}{\omega_{\text{limit}}} T_n^b(\omega_{\text{limit}})\right], & \text{if } |\omega| < \omega_{\text{limit}}, \end{cases} \tag{6.6}$$

where τ_s is the static torque, τ_d the dynamic torque, ω_c is a characteristic rotary speed, ω_{limit} is the limit rotary speed.

The torque T_n^b is applied only to the last mass (the bit), as it represents the resistive torque due to the bit-rock interaction. Since the downhole motor is assumed to be positioned immediately upstream of the bit, the torque T_n^b acts directly on the bit, which rotates at a velocity of $\omega_b + \omega_{\text{Motor}}$. Hence, T_n^b is expressed in terms of $\omega_b + \omega_{\text{Motor}}$ rather than ω_b alone.

6.4 Simplified torsional dynamical model

For an idealized drill string, we assume:

(i) The bit remains centered along the borehole axis, i.e. no spurious lateral motions of the bit.

(ii) Coupling effects with axial and lateral vibrations are disregarded.

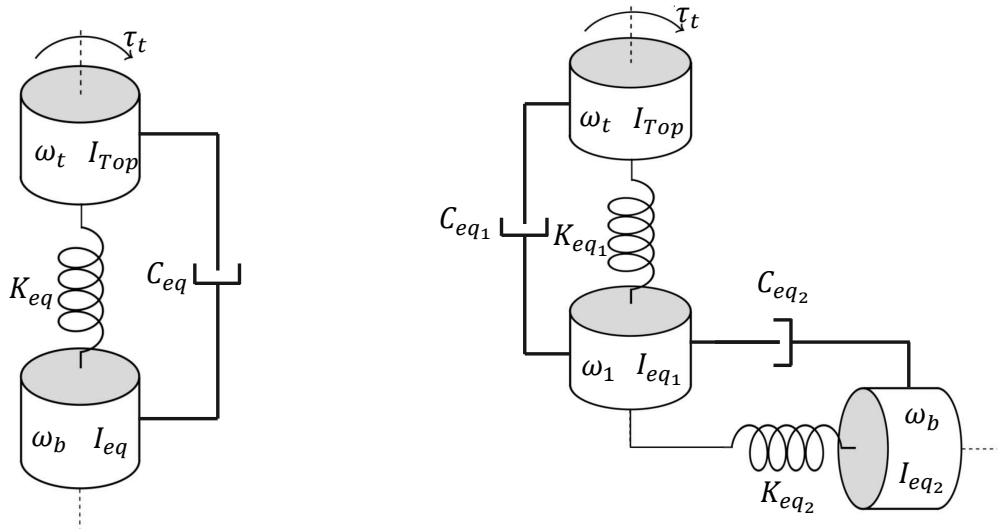
(iii) The motor's rotational speed is directly added to the bit's angular velocity.

(iv) The vertical section of the drill string is always considered to be in the slip phase, while only the lateral section can enter the stick phase.

(v) All frictional torques between the pipes and the borehole wall in the lateral section are lumped with the TOB.

Based on these assumptions, two simplified lumped-mass models are proposed [182, 183]:

- **One-mass lumped model (Fig. 6.3a):** the entire drill string is represented as a single equivalent mass.
- **Two-mass lumped model (Fig. 6.3b):** the drill string is separated into two inertial components. The first mass represents the drill pipes in the vertical section along with the drive system inertia (I_{Top}), while the second mass corresponds to the lateral section of the drill string, including the BHA inertia augmented by the inertia of the drill pipes in that section.



(a) One-mass lumped model

(b) Two-mass lumped model

Figure 6.3: The two simplified lumped-mass models considered in this work.

This modeling choice is motivated by three main reasons. First, a simple lumped-mass model, when combined with actual drilling parameters, can effectively reproduce complex nonlinear drilling dynamics. For instance, Cunha Lima et al. [184] develop a simplified single-degree-of-freedom model to study the torsional stability of the drill string, for better understanding the causes of stick–slip phenomenon and the way to mitigate it. Building on this work, similar mechanical models are proposed by [185, 99, 186, 187, 34] to investigate the influence of various drilling parameters on stick–slip vibrations.

Second, measured frequencies of torsional and stick–slip oscillations are often found to be close to the estimated first natural torsional frequency of the drill string assembly [175, 43, 170, 188]. Richard [189] compares the stick-slip frequency to the dominant frequency of the angular velocity signal measured while rotating the bit off-bottom, which is assumed to correspond to the drill string's natural torsional frequency. Their findings show that self-excited torsional vibrations may also occur at fundamental frequencies lower than the first natural torsional frequency, depending on operational conditions [190]. As a result, a physical one-mass or two-mass model is sufficient to capture stick-slip dynamics. In practice, much of the added complexity in more advanced models does not significantly improve the overall accuracy of downhole velocity or friction coefficients estimation. This is primarily due to the variability of well conditions, and to the limitations of available measurements, which are typically insufficient to keep all parameters of a complex model accurately calibrated over time.

Third, the objective is set to develop a solution capable of estimating downhole velocity and friction coefficients using low-frequency surface data. The solution is intended to be generalizable, suitable for real-time application, and directly applicable to field data. Given that surface data are typically recorded at 1 Hz, the industrial standard frequency, more complex physical models require substantially more information to produce accurate results, as additional parameters must be estimated and a larger number of equations satisfied. Other reasons specifically related to each of the proposed methods are detailed in the following chapters.

The one-mass simplified lumped physical model of the drill string can be expressed as follows:

$$\begin{aligned} I_{\text{Top}}\,\dot{\omega}_t &= \tau_t - K\,\Delta \\ \dot{\Delta} &= \omega_t - \omega_b \\ I\,\dot{\omega}_b &= -C\omega_b + K\Delta - T^f(\omega_b, \mu_s, \mu_d) - T^b(\omega_b + \omega_{\text{Motor}}, \tau_s, \tau_d) \end{aligned} \tag{6.7}$$

where ω_t and τ_t are the applied angular velocity and measured torque at the surface (top drive), respectively. The parameters I, K, and C denote the equivalent inertia, torsional stiffness, and damping coefficient of the drill string modeled as a single mass, and are computed using Eq. (6.1). I_{Top} is the inertia of the top drive system, ω_b is the angular velocity of the drill string, and ω_{Motor} is the rotational speed of the downhole motor. The variable Δ represents the relative angular displacement between the top drive and the bit. T^f represents the friction torque and can follow either the stiction friction model (Section 6.3.2.2) or the smooth friction model (Section 6.3.2.1). T^b represents the TOB and is given by Eq. (6.6).

The corresponding two-mass simplified lumped model is given by:

$$\begin{aligned} I_{\text{Top}}\ \dot{\omega}_{\text{t}} &= \tau_t - K_1\ \Delta_1 \\ \dot{\Delta}_1 &= \omega_t - \omega_1 \\ I_1\ \dot{\omega}_1 &= -C_1\omega_1 + K_1\Delta_1 - K_2\Delta_2 - \mu_d R_1 F_1^N \\ \dot{\Delta}_2 &= \omega_1 - \omega_b \\ I_2\ \dot{\omega}_b &= -C_2\omega_b + K_2\Delta_2 - T^f(\omega_b, \mu_s, \mu_d) - T^b(\omega_b + \omega_{\text{Motor}}, \tau_s, \tau_d) \end{aligned} \tag{6.8}$$

The parameters I_1, I_2, K_1, K_2, C_1, C_2, F_1^N, and F_2^N denote the equivalent inertias, torsional stiffnesses, damping coefficients, and normal contact forces associated with the two lumped masses, ω_1 and ω_b are the angular velocities of the first and second masses. The variables Δ_1 and Δ_2 represent the relative angular displacements between the top drive and the first mass, and between the first mass and the bit, respectively. Finally, as stated above, the vertical section of the drill string, represented by the first mass, is always considered to be in the slip phase. Consequently, the friction torque applied to the first mass is always equal to $\mu_d F_1^N$. The second mass, corresponding to the lateral section, can enter the stick phase, where interactions with the borehole wall and the bit are more pronounced.

6.5 The objective

The main objective of the second part of this dissertation is to develop a real-time, generalizable model that can handle the nonlinearity of friction and is capable of predicting the downhole rotational speed, as well as the static and dynamic friction coefficients between the drill string and the borehole, for lateral wells using only surface measurements (surface torque and surface rotation speed) at 1 Hz. This work focuses on addressing the limitations of the methods discussed in Section 6.2 by exploring two different modeling approaches, each grounded in a different paradigm. These approaches share a common goal: to leverage both surface measured data and physical laws in order to improve the estimation of downhole parameters.

- **Machine learning** (Chapter 7): A Physics-Informed Neural Network (PINN) [14], which leverages both measured data and physics laws into the learning process, is employed. This approach improves the model's ability to generalize and ensures consistency with known physical behavior. It is an offline method.
- **Control theory** (Chapter 8): the Salted Kalman Filter (SKF) [23], an extension of the classical Kalman filter designed for hybrid systems, is explored. It allows for real-time estimation under uncertainty, making it a promising candidate for inferring downhole parameters dynamically. Unlike the PINN, SKF is an online estimation technique.

Each method offers different advantages and disadvantages in terms of interpretability, computational cost, adaptability to changing conditions, and suitability for real-time versus offline use. Detailed discussions of these aspects are provided in the subsequent chapters.

Chapter 7

Physics-Informed Neural Network

Résumé du chapitre

Ce chapitre présente le développement d'une approche d'apprentissage automatique généralisable pour estimer la vitesse de rotation au fond du puits et les coefficients de frottement en utilisant uniquement des mesures de surface à 1 Hz, sans nécessiter de données temps réel ou historiques du fond du puits. La méthode proposée repose sur un Réseau de Neurones Informé par la Physique (Physics-Informed Neural Network (PINN)), qui intègre directement un modèle dynamique torsionnel non linéaire simplifié dans la fonction de perte. Conçue comme une solution hors ligne et généralisable, cette approche peut être appliquée à tout puits dont la dynamique est capturée par le modèle simplifié. La méthode est évaluée sur des données simulées et réelles. De plus, une stratégie d'apprentissage par transfert est employée pour améliorer la vitesse d'entraînement.

Chapter Summary

This chapter presents the development of a generalizable machine learning approach for estimating downhole rotation speed and friction coefficients using only 1-Hz surface measurements, without requiring any real-time or historical downhole data. The proposed method is based on a Physics-Informed Neural Network (PINN), which embeds a simplified nonlinear lumped torsional dynamic model directly into the loss function. Designed as an offline and generalizable solution, this approach can be applied to any well whose dynamics are captured by the lumped model. The method is evaluated on both simulated and field data. Additionally, a transfer learning strategy is employed to improve training speed.

7.1 Review

In recent years, neural networks have been increasingly explored in drilling research for detecting downhole events and estimating downhole measurements using surface data, as detailed in Section 6.2. These purely data-driven models have shown promise but remain limited in several key aspects. Their effectiveness typically hinges on the availability of large, high-quality datasets, and they often fail to account for the underlying physical principles of the drilling process. As a result, such models may produce physically inconsistent outputs and exhibit poor generalization beyond their training distributions.

To overcome these limitations, the research community has turned to Physics-Informed Neural Networks (PINNs) [14], a class of models that integrate physical laws, usually expressed as partial or ordinary differential equations, directly into the training process. Unlike conventional neural networks that treat learning as a purely statistical task, PINNs enforce physical consistency by incorporating governing equations, initial conditions, and boundary conditions into the loss function in addition to the supervised part of the loss, where data is used. This hybrid learning approach enables them to estimate unknown quantities while remaining faithful to the physics of the system [191, 192].

PINNs have gained traction for solving high-dimensional PDEs [193] and are particularly appealing when data are sparse [194, 195]. They have been applied across various scientific and engineering fields, including gas flow modeling [4], nano-optics [196], supersonic aerodynamics [197], and materials science [198]. These successes highlight the versatility and potential of PINNs in bridging the gap between data and theory.

Despite their promise, PINNs are not without challenges. Training PINNs is computationally intensive due to their reliance on gradient-based optimization, often making them slower than classical numerical solvers like Finite Element Method (FEM) and Finite Difference Method (FDM) [199]. Moreover, deep architectures can suffer from vanishing gradients [200], and the optimization process may become trapped in local minima. Designing an effective loss function requires manual tuning and domain expertise, while selecting appropriate training data often involves trial and error [14]. These limitations hinder the use of PINNs in real-time or production-scale settings. To address these practical concerns, efforts are made to improve training efficiency, including better initialization strategies [201, 202, 203]. In our work, a transfer learning strategy is proposed (see Section 4.3) to significantly reduce training costs while maintaining acceptable accuracy, making PINNs more feasible for industrial applications.

In the oil and gas industry, the integration of physics into machine learning remains relatively underdeveloped. However, there is a clear trend toward embracing more advanced Physics-Informed Machine Learning (PIML) methods. For instance, in the geophysical domain, Moseley et al. [204] incorporate the second-order acoustic wave equation and corresponding boundary conditions directly into the loss function of a PINN to construct velocity models of increasing complexity, including homogeneous, layered and Earth-realistic models, while Gou et al. [205] use the first-order wave equation to estimate both velocity and density fields. Other notable applications include downhole pressure and temperature prediction in geothermal drilling [206], as well as bottom-hole pressure forecasting using a physics-constrained data-driven framework enhanced with particle swarm optimization [207].

Among these works, the approach of Redaud et al. [208] is particularly relevant to our objective. They propose a dual-architecture Transformer-based PINN designed to estimate downhole velocity and friction coefficients between the drill string and the borehole. Their model predicts

the distributed state variables $\omega(t,x)$ and $\tau(t,x)$ at any time t and spatial location $x \in [0,L]$, where L is the length of the drill string, using sequences of past surface and downhole measurements. The loss function used in this model consists of three components: a data loss term, a physics-based loss term, and a boundary condition loss term. The model is trained on a large synthetic dataset generated from simulations using a test well geometry, and evaluated on additional sequences generated with the same well trajectory and parameters as the training data. Simulated surface measurements used in the study are sampled at a high frequency of 20 Hz. However, this setup presents several limitations for practical applications. First, the model is validated only on simulated data and not on field measurements. Second, in real-world scenarios, surface data is typically available at a much lower frequency (e.g., 1 Hz). Third, the model is both trained and tested using data derived from the same well configuration, which raises concerns about its ability to generalize to other drilling environments. Lastly, the data loss term in the loss function relies on the availability of ground-truth torque and velocity distributions along the drill string, information that is rarely accessible in real drilling operations.

A common feature across most of these works is the structure of the loss function used to train the PINN, which typically consists of at least four components: a physical loss enforcing the governing equations, a data loss term that incorporates known values of the target variable at specific times or positions, an initial condition loss, and a boundary condition loss. These components collectively ensure that the model learns both from the data and the underlying physics. Nevertheless, in many practical scenarios, including ours, such comprehensive information is not available. Specifically, we lack direct observations of the variables to be estimated (e.g., downhole velocity and friction coefficients), as well as initial condition data. Instead, only access to 1-Hz surface measurements is available. In this context, the model must rely almost entirely on the physical loss and surface measurements to infer the hidden downhole dynamics. This constraint presents a significant challenge and distinguishes our approach from prior work, as it requires training the PINN without any direct supervision on the target variables.

In this chapter, we adopt a different trade-off by developing a PINN based on a nonlinear physical model, trained with only surface data from the sequence of interest. In other words, no prior dataset needs to be constructed. While this approach may require longer training times to converge, it eliminates the need for accurate initial estimates or any direct information about the target variables. The proposed method is general and well-suited for application across different wells, thus overcoming the limitations of traditional data-driven models. It is also robust, aiming to enhance drilling performance and mitigate stick-slip vibrations. The model is evaluated on industrial drilling data, demonstrating its ability to handle nonlinear dynamics and to converge reliably from arbitrary initializations.

7.2 Objective

This chapter introduces a PINN approach [209, 210] for estimating downhole bit rotation speed and friction coefficients in nonlinear torsional drilling systems, using only low-frequency surface measurements, torque and rotation speed, sampled at 1 Hz, the current industry standard. The core idea is to embed a simplified nonlinear lumped torsional model directly into the PINN's loss function, enabling the network to infer unmeasured quantities, such as static and dynamic friction parameters, from partial observations.

Based on two input sequences, the applied surface rotation speed, ω_t, and the measured surface torque, τ_t, we aim to estimate the unmeasured internal states, $\hat{\mathbf{x}}$, and the friction coeffi-

cients $\hat{p}$. These quantities are predicted using a neural network that takes time as input, and the estimation is performed by solving the following constrained optimization problem:

$$\hat{\mathbf{x}}_{\hat{\theta}},\, \hat{\theta},\, \hat{p} = \arg \min_{\substack{\mathbf{x}_\theta,\, \theta,\, p \\ \dot{\mathbf{x}}_\theta(t) = f(\mathbf{x}_\theta, u, p)}} \left(\int_0^T \| \omega_t(t) - h(\mathbf{x}_\theta(t)) \|^2 \, dt \right) \tag{7.1}$$

where $h(\hat{\mathbf{x}}_{\hat{\theta}}(t)) = \hat{\omega}_t(t)$ computes the predicted surface rotation speed from the estimated states $\hat{\mathbf{x}}_{\hat{\theta}}$, and $\hat{\theta}$ denotes the neural network's weights. In this formulation, the function f represents the nonlinear torsional model incorporated directly into the loss function, and $\hat{\omega}_t$ denotes the predicted surface rotation speed. This approach bridges the gap between purely data-driven methods, which rely on dense labeled datasets, and purely physics-based models, which require prior knowledge of friction coefficients as input.

Our method is designed to be generalizable across wells, requiring neither prior knowledge of friction parameters nor access to downhole measurements during training. The PINN is trained by minimizing a loss function composed of two terms: a physics-based term enforcing consistency with the governing differential equations, and a term to force alignment with available surface measurements. No supervised loss on downhole targets is required. The approach is validated on both simulated and field data, for which no ground truth values are known for friction coefficients. To the best of our knowledge, our work represents the first application of PINNs for parameter estimation in dynamical systems using industrial drilling measurements. For simulations, the ground truth is generated using a high-fidelity PDE-based model with a stiction friction law, while the PINN uses a simpler smooth friction model (see Section 6.3.2.1) in its embedded dynamics. To improve training efficiency, a TL strategy is also employed, significantly reducing computation time without compromising prediction accuracy.

It is important to note that the static friction coefficient becomes observable primarily during the transition from stick phases to slip phases. Since typical stick-slip cycles last between 5 and 20 seconds, only a few measurement points per cycle are available at 1 Hz. However, when stick-slip oscillations are present, both static and dynamic friction coefficients can be effectively estimated.

7.3 Physics-Informed Neural Network

PINNs are deep learning models designed to solve supervised learning tasks while incorporating physical laws described by nonlinear partial or ordinary differential equations of the form:

$$\frac{\partial^n y(t,x)}{\partial t^n} = f\left(t, x, y, \frac{\partial y}{\partial t}, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial t^2}, \frac{\partial^2 y}{\partial x^2}, \frac{\partial^2 y}{\partial t \partial x}, \dots\right)$$

These models are particularly suited for **supervised** tasks that require minimizing errors with respect to both observed data and physical constraints [14, 192]. The general form of the loss function is defined as:

$$\mathscr{L} = \lambda_{Data}\, \mathscr{L}_{Data} - \lambda_{Physics}\, \mathscr{L}_{Physics} - \lambda_{Initial}\, \mathscr{L}_{Initial} - \lambda_{Boundary}\, \mathscr{L}_{Boundary} \tag{7.2}$$

Here, $\mathscr{L}_{Data}$ corresponds to the data loss computed using available target values at sparse time points t and positions x (supervised part). The term $\mathscr{L}_{Physics}$ represents the physics loss derived from the differential equations governing the system. The terms $\mathscr{L}_{Initial}$ and $\mathscr{L}_{Boundary}$

are the losses associated with initial and boundary conditions, respectively. The coefficients λ_{Data}, $\lambda_{Physics}$, $\lambda_{Initial}$, and $\lambda_{Boundary}$ are weights that balance the contribution of each component in the total loss. These weights can either be fixed or adapted during training using self-adaptive weighting techniques, which automatically adjust them to improve convergence and performance [211, 212]. Assuming a mean squared error formulation for all terms and denoting $\dot{y} = f(t,x,y)$ as the physical model equations, the total loss becomes:

$$\begin{aligned}\mathscr{L} = &\frac{\lambda_{Data}}{N_{Data}} \sum |y(t,x) - \hat{y}(t,x)|^2 + \frac{\lambda_{Physics}}{N_{Physics}} \sum \left|\dot{\hat{y}}(t,x) - f(\hat{y}(t,x),t,x)\right|^2 \\ &+ \frac{\lambda_{Initial}}{N_{Initial}} \sum |y(0,x) - \hat{y}(0,x)|^2 + \frac{\lambda_{Boundary}}{N_{Boundary}} \sum \left|y(t,x_{Boundary}) - \hat{y}(t,x_{Boundary})\right|^2,\end{aligned} \tag{7.3}$$

where N_{Data}, $N_{Physics}$, $N_{Initial}$, and $N_{Boundary}$ denote the number of points used in the supervised data loss, physics-based loss, initial condition loss, and boundary condition loss, respectively. In this framework, time t and position x are used as inputs to the PINN, which outputs $\hat{y}$. This output is directly used to compute the data, initial, and boundary losses. However, to compute the physics loss, derivatives of $\hat{y}$ with respect to the inputs must be calculated. These derivatives are obtained using automatic differentiation [213], which enables PINNs to encode physical constraints effectively during training.

7.4 PINN for estimation in drilling

In our case, the target variables to be estimated are the friction coefficients between the drill string and the borehole, along with the bit rotational speed, based solely on 1-Hz surface measurements . For each sequence of surface torque τ_t and rotational speed ω_t over a given time interval, the PINN outputs two constant friction coefficients and a time-dependent sequence of downhole rotational speed corresponding to that interval. No direct or historical measurements of the bit rotational speed are used during training. Therefore, the total loss is composed exclusively of a physics-based term $\mathscr{L}_{Physics}$, which enforces consistency with the governing physical equations, and a term ensuring alignment with the surface measurements. The model is trained for **each sequence** of data, which typically contains 30 points for 30-second of 1-Hz surface torque and velocity.

7.4.1 PINN's physical model

To address the estimation problem, the PINN is tested using two simplified lumped torsional dynamic models, in the physical loss function:

- **Case 1: One-mass lumped model**: This model assumes that the entire drill string behaves as a single rotating mass. It is described by Eq. (6.7).
- **Case 2: Two-mass lumped model**: This model divides the drill string into two rotating masses, one representing the vertical section and the other the lateral section. It is described by Eq. (6.8).

The motivations for using these simplified models are detailed in Section 6.4. From a PINN-specific perspective, there are two main reasons for this choice. First, the more complex the

physical model, the longer the training time, which conflicts with our objective of developing a real-time estimation method. Second, complex models require high-frequency measurements and more training data about the target variables, which implies the use of the supervised loss term $\mathscr{L}_{Data}$ in Eq. (7.2). For instance, Redaud et al. [208] employ a PDE-based torsional dynamics model and rely on 20 Hz surface measurements along with historical simulated supervised data (i.e., drill string velocity for $x \in [0, L]$) to train their network. However, in our setting, the objective is to develop a model that operates in real-time and depends solely on a single sequence of 1-Hz surface data. These constraints necessitate the simplification of the physical model. Furthermore, in our experience, training PINNs with more detailed lumped models involving multiple masses, while feasible, demands significant tuning of the loss weighting parameters. In particular, different weights must be assigned to each equation of the dynamical model.

For both the one-mass and two-mass models, the smooth friction model described in Section 6.3.2.1 is used. The rationale behind this choice is related to the use of automatic differentiation (autodiff) [213], which is employed by PINNs to compute derivatives while training.

Automatic differentiation refers to the concept of building a general-purpose tool capable of computing the derivative of any function that is implemented as a program in a general-purpose programming language [214, 215]. These tools are designed to compute derivatives of differentiable functions, however, in practice, they are often applied to functions that are not differentiable everywhere. A common example is the ReLU activation function used in neural networks, which introduces non-differentiabilities. Despite this, autodiff systems are widely employed to compute gradients of loss functions involving such non-smooth operations. As discussed in Lee et al. [216], autodiff systems are considered valid even when applied to non-differentiable functions, provided that the non-differentiable points are rare, specifically, if they occur on a subset of the input domain with Lebesgue measure zero[1]. In such cases, the system is expected to compute correct derivatives almost everywhere, meaning for all inputs except for a set of measure zero.

However, in the case of stick-slip phenomena modeled using the stiction friction model described in Section 6.3.2.2, the function becomes non-differentiable during the stick phase, i.e., when $\omega_b = 0$. Unlike the ReLU case, these non-differentiabilities are not rare and can occur frequently. As a result, using automatic differentiation under such conditions becomes unreliable, as the derivatives do not exist at those points and their occurrence is not negligible. For this reason, the smooth friction model is preferred over the stiction model in our implementation. It ensures differentiability across the entire domain, allowing the PINN to compute gradients reliably and train effectively using automatic differentiation.

7.4.2 PINN's loss function

For both cases, using the one-mass and two-mass physical models, the PINN takes time t as input and predicts the state vector, defined as follows:

- **Case 1: One-mass lumped model** (Eq. (6.7)): $\hat{\mathbf{x}} = (\hat{\omega}_t, \hat{\Delta}, \hat{\omega}_b)$
- **Case 2: Two-mass lumped model** (Eq. (6.8)): $\hat{\mathbf{x}} = (\hat{\omega}_t, \hat{\Delta}_1, \hat{\omega}_1, \hat{\Delta}_2, \hat{\omega}_b)$

[1] A set is said to have *Lebesgue measure zero* if it can be covered by a countable collection of intervals whose total length can be made arbitrarily small. In other words, a set $S \subset \mathbb{R}^n$ has measure zero if, for every $\varepsilon > 0$, there exists a countable collection of n-dimensional boxes $\{I_k\}$ such that $S \subset \bigcup_k I_k$ and $\sum_k \text{Vol}(I_k) < \varepsilon$.

where $\hat{\Delta}$, $\hat{\Delta}_1$, and $\hat{\Delta}_2$ denote the predicted relative angular displacements:

- $\hat{\Delta}$ between the top drive and the bit,
- $\hat{\Delta}_1$ between the top drive and the first mass.
- $\hat{\Delta}_2$ between the first mass and the bit.

The friction coefficients to be estimated are treated as additional trainable variables and are therefore not included in the state vector as in Chapter 9. To facilitate parameter identification, the friction coefficients are reparameterized as the pair $(\hat{\mu}_s, \hat{r})$, where the dynamic friction coefficient is expressed as $\hat{\mu}_d = \hat{r}\hat{\mu}_s$. Here, $\hat{\mu}_s$ denotes the predicted static friction coefficient, and $\hat{r}$ represents the ratio between dynamic and static friction coefficients. Physically meaningful constraints require that

$$0 \leq \hat{\mu}_d = \hat{r}\hat{\mu}_s \leq \hat{\mu}_s \leq 1 \tag{7.4}$$

To naturally enforce these bounds within the PINN framework, the parameters are defined through sigmoid transformations:

$$\hat{\mu}_s = \sigma(\hat{\alpha}), \quad \hat{r} = \sigma(\hat{\beta})$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ maps any real input $x \in \mathbb{R}$ to the interval $[0,1]$. This reparameterization guarantees that $\hat{\mu}_s$ and $\hat{r}$ remain within physically admissible ranges during training.

The trainable parameters of the PINN thus include the network weights $\hat{\theta}$ and the physical parameters $\hat{p} = \{\hat{\alpha}, \hat{\beta}\}$. The loss function then can be written as:

$$\begin{aligned} \mathscr{L}(\hat{\theta}, \hat{p}) =& \lambda_f \sum_{i=1}^{N_f} \left\| \dot{\hat{\mathbf{x}}}_{\hat{\theta}}(t_i) - f\left(\hat{\mathbf{x}}_{\hat{\theta}}(t_i), \tau_t(t_i), \hat{p}\right) \right\|^2 \\ &+ \lambda_{Top} \sum_{i=1}^{N_{Top}} \left\| \omega_t(t_i) - h(\hat{\mathbf{x}}_{\hat{\theta}}(t_i)) \right\|^2 \end{aligned} \tag{7.5}$$

with $h(\hat{\mathbf{x}}_{\hat{\theta}}(t_i)) = \hat{\omega}_t(t_i)$ computes the predicted surface rotation speed from the estimated states $\hat{\mathbf{x}}_{\hat{\theta}}$. The first term is the *physics loss*, which enforces the torsional dynamics defined by either the one-mass or two-mass model depending on the case, and the second term ensures that the model learns the surface angular velocity ω_t. τ_t denotes the measured surface torque, ω_t and $\hat{\omega}_t$ represent the measured and predicted surface rotation speeds respectively. N_{Top} is the number of surface measurements, N_f the number of collocation points used to enforce the physical model, and λ_{Top} and λ_f are weighting coefficients balancing each loss term's contribution. The methodology for selecting these coefficients is presented in the next section.

Remark Notice that Eq. (7.5) corresponds to a reformulation of Eq. (7.1) where :

- The hard constraint that the solution must satisfy the dynamics is recast a soft constraint.
- The infinite-dimensional optimization problem of finding a trajectory is approximated by a finite-dimensional problem where the trajectory is parametrized by the neural network. Our approach is, in this sense, a form of collocation method for system identification.

In this formulation, the function f is defined as follows:
For the one-mass model (case 1):

$$f\left(\hat{\mathbf{x}}_{\hat{\theta}}, \tau_t, \hat{p}\right) = \begin{pmatrix} \frac{\tau_t - K\hat{\Delta}}{I_{\text{Top}}} \\ \hat{\omega}_t - \hat{\omega}_b \\ \frac{-C\,\dot{\hat{\omega}}_b + K\hat{\Delta} - T^f(\hat{\omega}_b, \hat{p}) - T^b(\hat{\omega}_b + \omega_{\text{Motor}}, \tau_s, \tau_d)}{M} \end{pmatrix} \tag{7.6}$$

For the two-mass model (case 2):

$$f\left(\hat{\mathbf{x}}_{\hat{\theta}}, \tau_t, \hat{p}\right) = \begin{pmatrix} \frac{\tau_t - K_1\hat{\Delta}_1}{I_{\text{Top}}} \\ \hat{\omega}_t - \hat{\omega}_1 \\ \frac{K_1\hat{\Delta}_1 - C_1\,\dot{\hat{\omega}}_1 - \mu_d R_1 F_1^N}{I_1} \\ \hat{\omega}_1 - \hat{\omega}_b \\ \frac{-C_2\,\dot{\hat{\omega}}_b + K_2\hat{\Delta}_2 - T^f(\hat{\omega}_b, \hat{p}) - T^b(\hat{\omega}_b + \omega_{\text{Motor}}, \tau_s, \tau_d)}{I_2} \end{pmatrix} \tag{7.7}$$

where T^f and T^b are the the smooth friction law and the TOB functions respectively (see Section 6.3.2 and 6.3.3).

7.4.3 Choice of loss weighting coefficients

Choosing appropriate weights for different components of the loss function is a key aspect of successfully training PINN. In the literature, several strategies have been proposed to tackle this challenge. One class of methods seeks to learn the weights along with the model parameters during training to automatically balance the contribution of each term [217, 218]. Another family of approaches introduces adaptive weighting schemes, where the weights are adjusted dynamically throughout training to improve convergence and robustness [211, 212]. More recently, Hua et al. [219] propose a variance-based strategy that adjusts weights based on the uncertainty of the model, allowing the network to focus on regions with higher prediction error variance.

While these approaches are designed to balance data-driven (supervised learning on target variables) and physics-based terms, our case is different. In our case, the total loss consists of two parts: (i) a physics-based loss enforcing the governing dynamic equations, and (ii) a surface velocity loss that ensures the model accurately learns the surface angular velocity ω_t. To structure the training process, we assign one weighting coefficient to each equation in the dynamic model f. This results in three loss weights for the one-mass model (Eq. (7.6)) and five for the two-mass model (Eq. (7.7)), with an extra one corresponding to the surface rotation speed learning term.

These weights are not arbitrary, they define the training priority. Since all loss terms are normalized and unitless, the highest-weighted equation is minimized first. Once its associated error is sufficiently reduced, the training process naturally shifts focus to the next terms. This strategy enables the model to progressively learn in stages, starting with what matters most.

In practice, this means the PINN first learns to reconstruct the surface rotation speed and extract relevant information from torque measurements. Only then can it effectively estimate the friction parameters and downhole dynamics. If the PINN misses this initial step, the entire estimation process breaks down. For this reason, the terms associated with surface measurements are assigned the highest priority during the initial training stages. As their error diminishes, the

network turns to refining the physical consistency through the other dynamic equations. This staged approach ensures both stability and effectiveness in the learning process.

Accordingly, in both the one-mass and two-mass PINN models, the surface velocity loss term and the first dynamic equation, which includes the surface torque τ_t through the term $\frac{\tau_t - K\hat{\Delta}}{I_{\text{Top}}}$, are assigned the highest weighting coefficients. Through extensive trial-and-error experiments, we determined that fixed weights of 100 for these two components led to the most robust and consistent performance across both model variants. The remaining equations are each assigned a weight of 1. In the one-mass model, this corresponds to the two last dynamic equations. In contrast, the two-mass model involves more dynamic equations, making the choice of weights more intricate. As a result, its optimization requires finer tuning to maintain convergence stability while still enabling accurate friction estimation.

7.4.4 PINN's architecture

Fully Connected Neural Networks (FCNNs) are the most commonly used architecture in the design of PINN, as demonstrated in numerous studies [220, 221, 222, 223]. Based on this prevailing trend in the literature, FCNN architecture is used for our PINN implementation.

As illustrated in Fig. 7.1, the proposed model is a fully connected neural network that takes time t as its sole input and outputs a state vector $\hat{x}_\theta(t) \in \mathbb{R}^n$, where $\theta = (W, b)$ denotes the weights and biases of the network. The network consists of five dense layers. Each hidden layer contains 64 neurons and uses the hyperbolic tangent (tanh) activation function. The final output layer is linear and includes n neurons corresponding to the dimension of the state vector. The choice of activation function plays a critical role in addressing the nonlinear nature of the problem [221, 224]. In PINNs, nonlinear activations are essential for approximating complex solutions to differential equations, as they introduce the required nonlinearity into the model. This enables the network to learn intricate patterns and decision boundaries that linear functions cannot capture. Among these, the hyperbolic tangent (tanh) is a common and effective choice due to its smoothness, differentiability, and proven stability in physics-informed learning tasks [14, 225].

The model is trained using the Adam optimizer. To promote efficient convergence, a learning rate scheduler is applied that reduces the learning rate when the loss stagnates during training.

7.5 Transfer learning for PINN

Using the approach described in Section 7.4, for any sequence of measurements $(\tau_t(t), \omega_t(t))$, the loss function defined in Eq. (7.5) is minimized to obtain the optimal parameters $(\hat{\theta}, \hat{p})$. Consequently, the PINN must be retrained from scratch for each new data sequence, which can be computationally expensive. To mitigate this training cost, transfer learning strategies can be employed (see Section 4.3).

As explained in [65], transfer learning focuses on transferring the knowledge from a related domain (called source domain) to improve the learning performance or minimize the number of labeled examples required in a target domain (see Section 4.3). The pre-trained source model is used as the starting point for another model with a second task of interest. The re-training can be performed on all or parts of the source model.

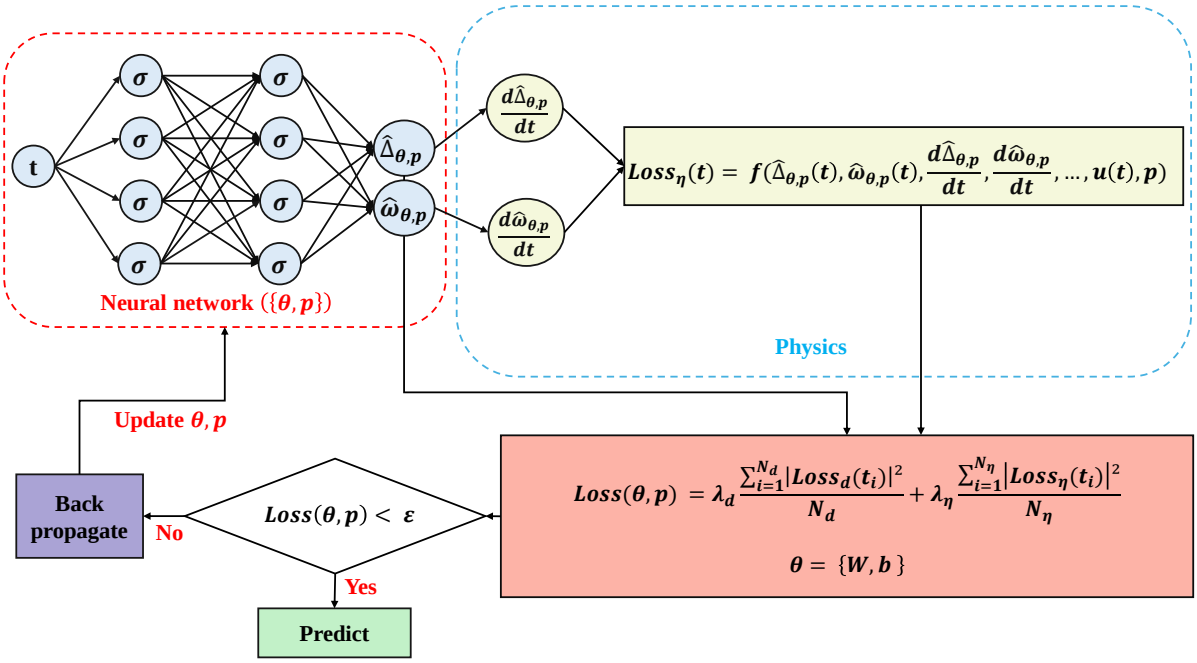


Figure 7.1: Schematic view of PINN [4]: t is the input time; $\hat{\omega}_{\theta,p}$ and $\hat{\Delta}_{\theta,p}$ are the predicted states by the PINN, p are the predicted friction coefficients, defined as additional trainable variables.

In our case, the source model is a PINN trained on an initial set of measurements $(\tau_t(t), \omega_t(t))$. When a new set of measurements $(\tau_{t,\text{new}}(t), \omega_{t,\text{new}}(t))$ becomes available, rather than training a new PINN from scratch to learn the corresponding parameters $\hat{p}_{\text{new}}$ and network weights $\hat{\theta}_{\text{new}}$, we employ a transfer learning strategy. This approach reuses the knowledge acquired by the source model to solve the nonlinear optimization problem. Specifically, the source model is fine-tuned on the new measurement sequences by retraining all trainable weights and biases, as well as the additional trainable variables representing the unknown friction coefficients $\hat{p}_{\text{new}}$. The same optimizer and learning rate scheduling as in the source model are maintained. This strategy significantly reduces computational cost while preserving robust performance.

7.6 Numerical results

In this section, the performance of the proposed methodology is illustrated for predicting the downhole bit rotation speed and friction coefficients (μ_s, μ_d) using only 1-Hz surface measurements as inputs, specifically the surface rotational velocity ω_t and surface torque τ_t. The methodology is first evaluated on simulated data generated from a PDE-based model [155], where the friction coefficients μ_s and μ_d are prescribed, allowing the estimated parameters to be compared with the ground truth. Finally, the approach is applied to field data sampled at 1 Hz from an actual drilling system.

The code is written in Python 3.10.9 and TensorFlow 2.11.0, and all the numerical examples have been run on AMD EPYC 7443 24-Core Processor.

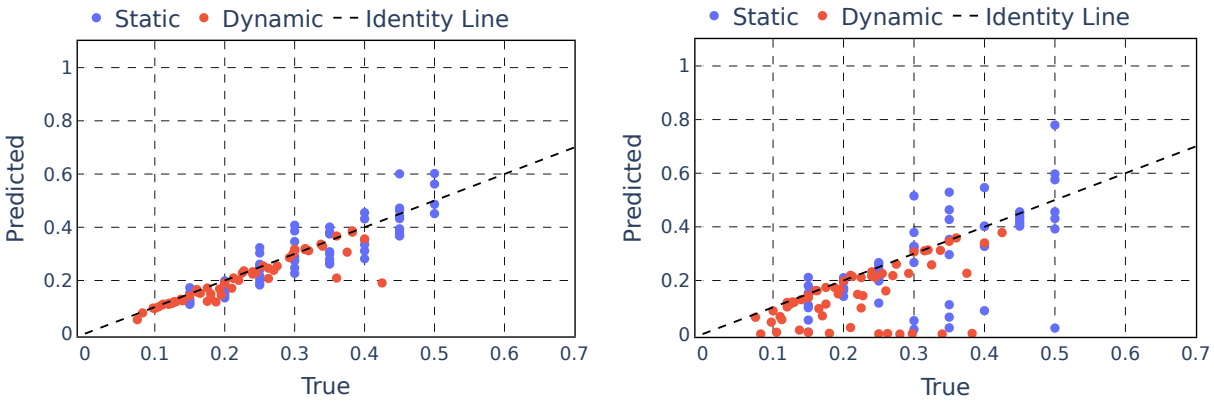
7.6.1 Application on simulated data

This section assesses the performance of the proposed PINN when applied to simulated data generated from a more complex and realistic physical model, specifically, a PDE-based torsional model, while the PINN itself continues to embed the simpler one-mass or two-mass lumped models in its loss function. To make the simulations more representative of real-world conditions, the stiction friction model (see Section 6.3.2.2) is used to generate the PDE-based simulations, while the smoother friction model is retained in the PINN's loss function to reflect the mismatch typically present between real-world complexity and simplified modeling assumptions. The evaluation is performed under the realistic constraint of a low measurement update frequency of 1 Hz, which reflects common limitations in field data acquisition.

The PDE-based model used to generate the simulation data is described in [155]. The trajectory of the test well employed for simulation is illustrated in Fig. 8.19. The parameters used in the one-mass and two-mass physical models embedded in the PINN are listed in Table 8.8. For the one-mass model, the parameters represent the equivalent behavior of the entire wellbore. In contrast, the two-mass model distinguishes between the vertical and lateral sections of the drill string, as discussed in Section 6.4. A total of 64 sequences are generated by randomly sampling the friction coefficients within the following intervals: $\mu_s \in [0.15, 0.5]$ and $\mu_d \in [0.075, 0.425]$. These simulated signals are then downsampled to 1 Hz to emulate realistic data resolution in drilling operations. For each simulation, corresponding to a specific pair of friction coefficients (μ_s, μ_d), a 30-second sequence is recorded at a sampling rate of 1 Hz. Each sequence includes the surface rotational speed ω_t, surface torque τ_t, and the bit rotational speed ω_b, which is used as ground truth for comparison with the predicted value $\hat{\omega}_b$. The PINN, for both the one-mass and two-mass lumped torsional dynamic models, is trained separately for each sequence using the loss functions defined in Eq. (7.6) and Eq. (7.7), respectively. Each sequence is treated independently, with no initialization from previously learned weights. On average, the training time per sequence is approximately 10 minutes for the one-mass PINN and about 12 minutes for the two-mass PINN. It is important to note that the initial values of the friction coefficients are randomly chosen for each training sequence.

Fig. 7.2 and Table 7.1 present the friction coefficient estimation results obtained from PDE-based simulated data. Fig. 7.2 compares the predicted and true values of the static and dynamic friction coefficients across 64 sequences for both the one-mass and two-mass PINN models. Correspondingly, Table 7.1 provides the MAE and MAPE for each friction parameter. The one-mass PINN, illustrated in Fig. 7.2a, demonstrates strong agreement with the ground truth, effectively estimating friction coefficients using only 1-Hz surface measurements and a smooth friction model. In contrast, the two-mass PINN, shown in Fig. 7.2b, exhibits less accurate predictions. This discrepancy arises from the increased complexity of the two-mass model, which incorporates additional physical equations in its loss function (see Eq. (7.7)) and requires careful tuning of multiple weighting coefficients. In addition, the higher number of parameters to be estimated significantly increases the complexity of the optimization problem and raises the risk of convergence to local minima.

Furthermore, Fig. 7.3 shows a comparison between the evolution of the simulated ground truth bit velocity ω_b and the predicted $\hat{\omega}_b$ given by the one-mass PINN for a test simulation. Fig. 7.3a displays the low-frequency (1 Hz) and high-frequency (100 Hz) downhole rotation speeds of the PDE-based generated bit rotation speed, with static friction coefficient $\mu_s = 0.15$ and dynamic friction coefficient $\mu_d = 0.12$. Fig. 7.3b presents the 1-Hz true bit rotation speed alongside the one-mass PINN predicted variations, while Fig. 7.4 presents their spectra at 1 Hz



(a) Case 1: One-mass PINN

(b) Case 2: Two-mass PINN

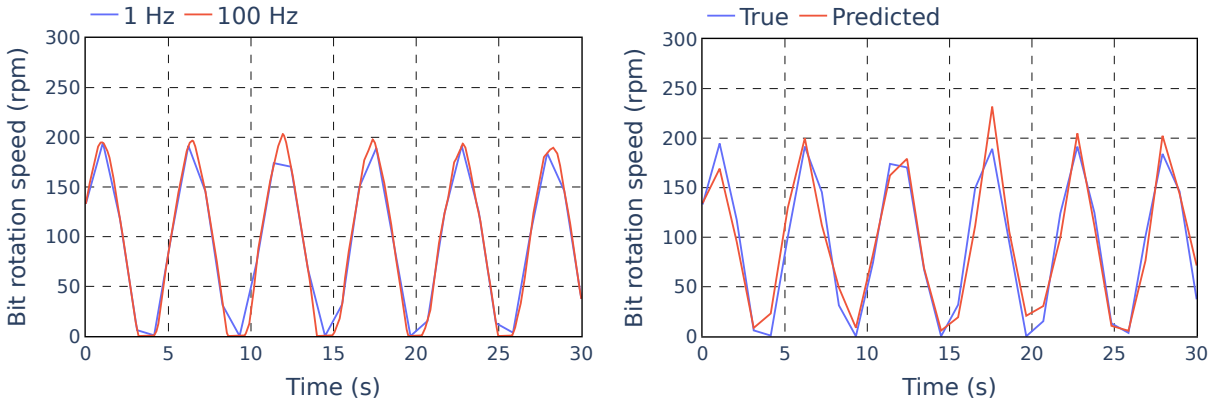
Figure 7.2: True and predicted values of the friction coefficients $(\hat{\mu}_s, \hat{\mu}_d)$ obtained by the PINN applied to 30-second sequences of PDE-based simulated data with a stiction friction model, for two cases: (a) Case 1: PINN using a one-mass nonlinear lumped torsional dynamic model in the loss function; (b) Case 2: PINN using a two-mass nonlinear lumped torsional dynamic model in the loss function.

	Case 1		**Case 2**	
	MAE	**MAPE (%)**	**MAE**	**MAPE (%)**
μ_s	47.25×10^{-3}	15.43	85.31×10^{-3}	26.93
μ_d	21.75×10^{-3}	9.29	71.32×10^{-3}	33.04

Table 7.1: MAE and MAPE of the friction coefficients predicted by the PINN across all test sequences of the two cases: Case 1: PINN using a one-mass nonlinear lumped torsional dynamic model in the loss function; Case 2: PINN using a two-mass nonlinear lumped torsional dynamic model in the loss function.

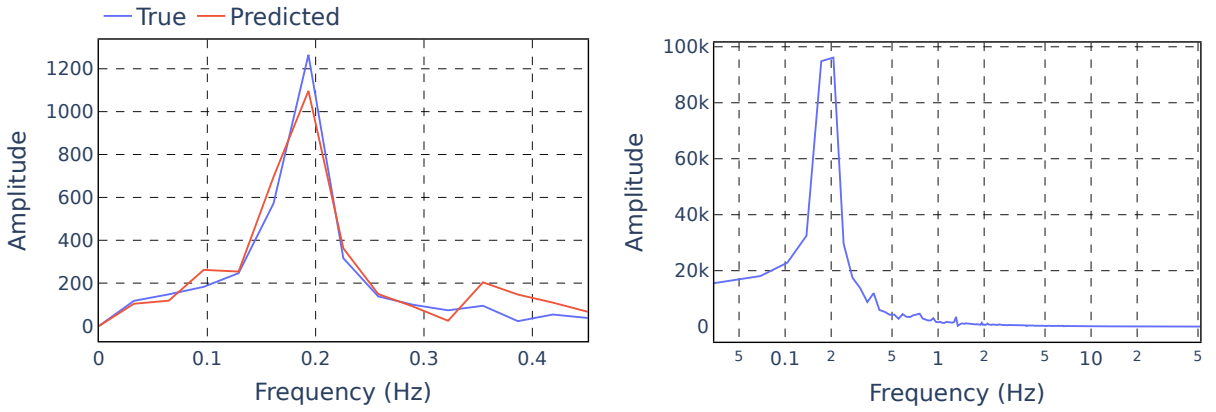
as well as the spectrum of the true bit rotation speed at 100 Hz. We observe that the one-mass PINN accurately predicts the dynamics of the bit rotation speed and captures the intricate nonlinear behavior of the Coulomb friction law implemented in the loss function. For this simulation, the predicted friction coefficients are $\hat{\mu}_s = 0.152$ and $\hat{\mu}_d = 0.114$. It is important to underline that, unlike most classical numerical methods for solving partial or ordinary differential equations, this prediction is obtained without any time-domain discretization. Instead, the time dependency of the solution is approximated by the neural network, which can be seen as a nonlinear projection.

Since the one-mass PINN can predict both the friction coefficients and the bit rotation speed of PDE-based simulated data using the stiction friction model, relying solely on 1-Hz surface measurements and without any supervised training, and since it demonstrates superior performance compared to the two-mass PINN, which is more difficult to train due to the increased number of weighting coefficients and equations integrated into the loss function, the one-mass PINN is used in the remainder of this chapter to predict the bit rotation speed and friction coefficients from field drilling data.



(a) (b)

Figure 7.3: (a) Low-frequency (1 Hz) and high-frequency (100 Hz) downhole bit rotation speed generated by the PDE-based model with static friction coefficient $\mu_s = 0.15$ and dynamic friction coefficient $\mu_d = 0.12$; (b) Comparison of the 1-Hz true bit rotation speed and the predicted bit rotation speed by the one-mass PINN, with predicted friction coefficients $\hat{\mu}_s = 0.152$ and $\hat{\mu}_d = 0.114$.



(a) (b)

Figure 7.4: Frequency spectra of (a) the true and predicted bit rotation speeds shown in Fig. 7.3 at 1 Hz, and (b) the true bit rotation speed at 100 Hz, plotted with a logarithmic x-axis, highlighting the PINN's ability to capture key dynamic features.

7.6.2 Application of transfer learning on simulated data

The effectiveness of TL is evaluated by training a source model on a simulated sequence of surface measurements (ω_t, τ_t) at a fixed depth D. Rather than training a new model for each sequence, the source model is adapted to new simulated sequences via TL to estimate both the friction coefficients and bit rotation speed. In this study, TL is applied exclusively to the PINN using the one-mass lumped model.

Fig. 7.5 illustrates the surface torque of the sequence used to train the source model and that of the new sequence used to test TL. The bit rotation speed predicted by the source model prior to TL is shown in Fig. 7.3, with corresponding friction coefficients of 0.15 (true) and 0.152 (predicted) for the static, and 0.12 (true) and 0.114 (predicted) for the dynamic coefficient. The new

sequence is generated with the same parameters as in Table 8.8, with friction coefficients set to $\mu_s = 0.5$ and $\mu_d = 0.325$.

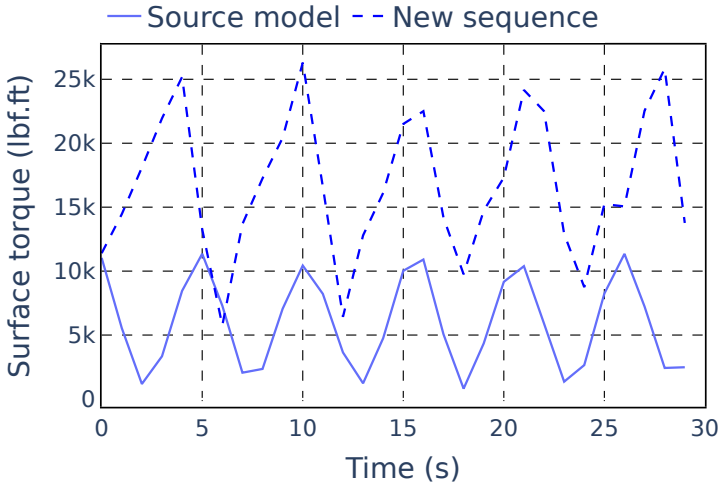


Figure 7.5: Surface torque variations of the original sequence used to train the source model, and the new sequence used for transfer learning application.

The source model is subsequently fine-tuned using the surface torque of the new sequence. After TL, the predicted friction coefficients for the new sequence are $\hat{\mu}_s = 0.477$ and $\hat{\mu}_d = 0.336$, while the predicted bit rotation speed is shown in Fig. 7.6. The training time is reduced substantially, requiring approximately 100 seconds per sequence, compared to nearly 600 seconds when a new PINN is trained from scratch. The true values of (μ_s, μ_d) are successfully approximated, and the downhole rotational speed variations are captured through adjustment of the source model parameters. The pronounced stick-slip oscillations at the bit rendered the static friction coefficient μ_s particularly influential, enabling accurate estimation of both μ_s and μ_d.

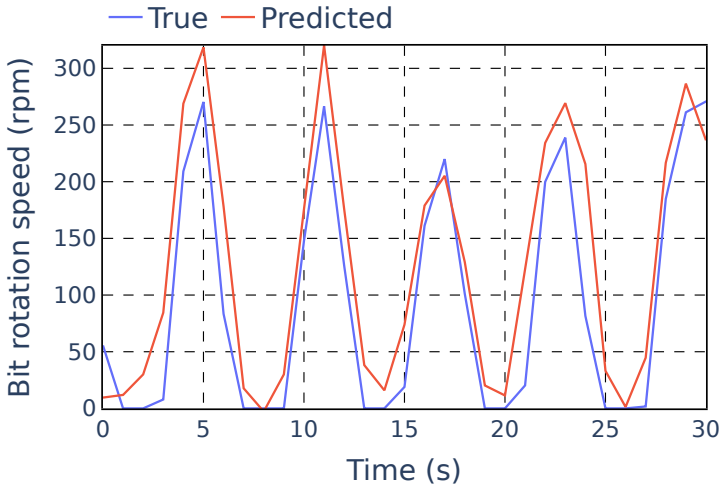


Figure 7.6: Bit rotation speed predicted by the PINN after TL application.

To further evaluate the impact of TL, the test sequences introduced in Section 7.6.1 are re-estimated using TL rather than retraining a new model each time. The predicted coefficients $(\hat{\mu}_s, \hat{\mu}_d)$ are shown in Fig. 7.7, while the corresponding results without TL are given in Fig. 7.2a.

Quantitative performance is summarized in Table 7.2, where MAE and MAPE are reported before and after TL. Slightly more accurate results are obtained when training from scratch, but TL significantly reduces the computational cost while maintaining comparable accuracy across most test sequences.

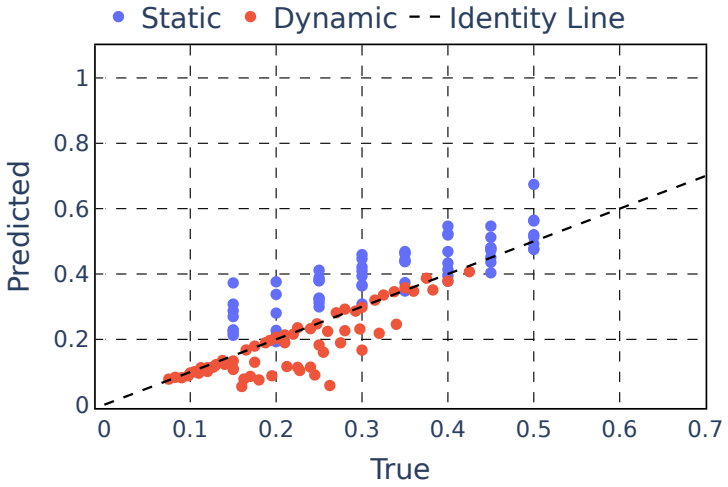


Figure 7.7: True and predicted friction coefficients (μ_s, μ_d) of the PDE-generated test sequences using the one-mass PINN with transfer learning.

	Without TL		With TL	
	MAE	**MAPE**	**MAE**	**MAPE**
μ_s	47.25×10^{-3}	15.43	75.55×10^{-3}	29.53
μ_d	21.75×10^{-3}	9.29	39.24×10^{-3}	17.86

Table 7.2: MAE and MAPE of the predicted friction coefficients using the one-mass PINN across all test sequences, with and without transfer learning.

7.7 Application on field data

To further illustrate the capabilities of the proposed PINN, it is tested on long sequences of field drilling data from multiple wells. While most of the results are presented in Chapter 9, a preliminary analysis is included in this chapter using data from well 1, which is described in Table 3.1 and whose trajectory is shown in Fig. 3.7.

The PINN, using a loss function based on the one-mass lumped torsional dynamic model and without TL application, is tested on six days of drilling data from test well 1. Only surface measurements sampled at 1 Hz are used, which are divided into 30-second sequences. For each sequence, the drill string's equivalent inertia, stiffness, and damping are computed using Eq. (6.1), and the equivalent normal contact force is estimated through Torque and Drag (T&D) modeling [155, 69, 176, 71]. The PINN is then trained for each sequence to minimize the loss function in Eq. 7.5, enabling it to predict the static and dynamic friction coefficients $(\hat{\mu}_s, \hat{\mu}_d)$, as well as the bit rotation speed $\hat{\omega}_b$. While ground truth values for the friction coefficients are not available, the predicted bit rotation speed can be compared against the true downhole

measurements, which are obtained after the well has been drilled, allowing for a quantitative evaluation of the model's performance. The training time for each sequence is approximately 10 minutes.

Fig. 7.8 shows the comparison results for one short sequence, 210 seconds in length, exhibiting severe stick-slip behavior. Surface torque, surface velocity, and downhole measured TOB variations are presented, along with the real bit rotation speed measurements, which are compared against the predicted values $\hat{\omega}_b$ from the PINN. All downhole data are retrieved after the well was drilled. The real bit rotation speed is represented by three curves, maximum, minimum, and mean, extracted from high-frequency downhole sensor data sampled at 80 Hz over 2.56 seconds intervals. As seen in Fig. 7.8, for test well 1, the PINN accurately reproduces the bit rotation speed measurements, including the full range of observed values, using only 1-Hz surface measurements as input.

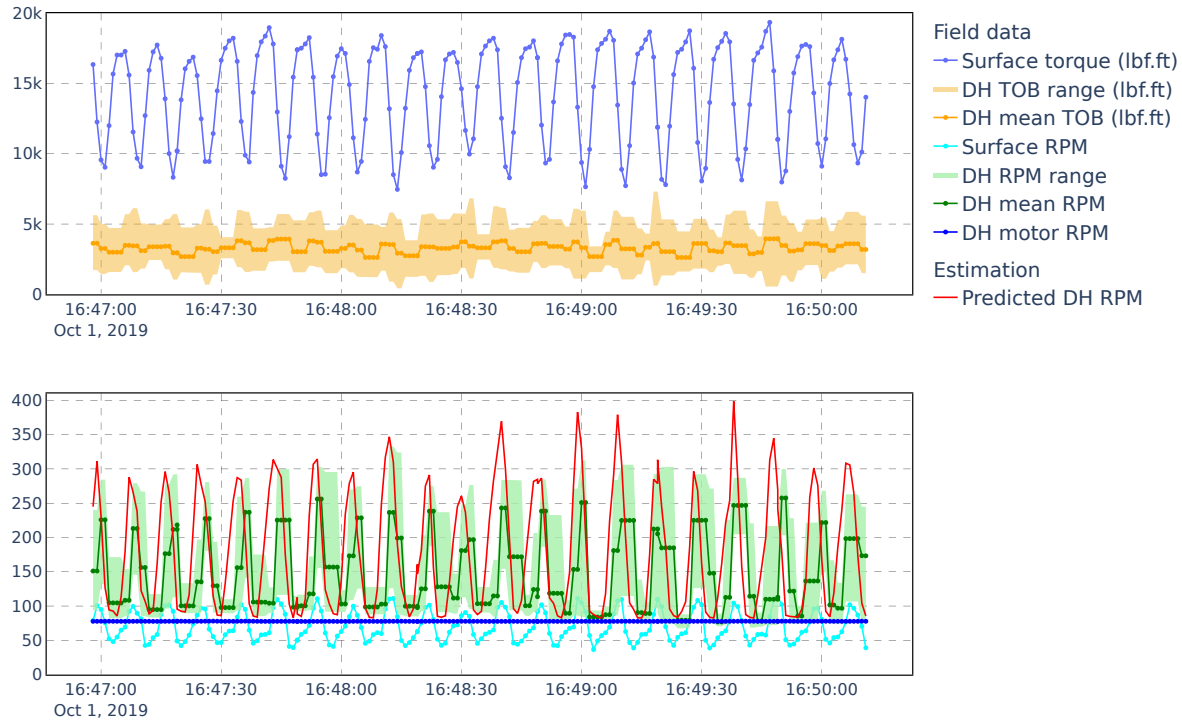


Figure 7.8: Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained by the PINN, using real surface torque and surface rotational speed as inputs for test well 1. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; Surface RPM - surface rotational speed; DH RPM range - maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM - mean of measured downhole bit rotation speed; DH motor RPM - measured downhole motor rotation speed; Predicted DH RPM - predicted downhole bit rotation speed by the PINN.

To evaluate the performance of the PINN over extended sequences, Fig. 7.9 presents the moving maximum, minimum, and average of both measured and predicted bit rotation speeds over a long period of six days. The figure also includes the moving average of the downhole

motor rotation speed. These metrics are computed within a 120-second window. Table 7.3 reports the MAE and Pearson correlation coefficient (r) (see Eq. (3.4)) between the true and predicted bit rotation speeds. As demonstrated, the predicted $\hat{\omega}_b$ successfully captures the severe variations of the downhole bit rotation speed using only 1-Hz surface data as input.

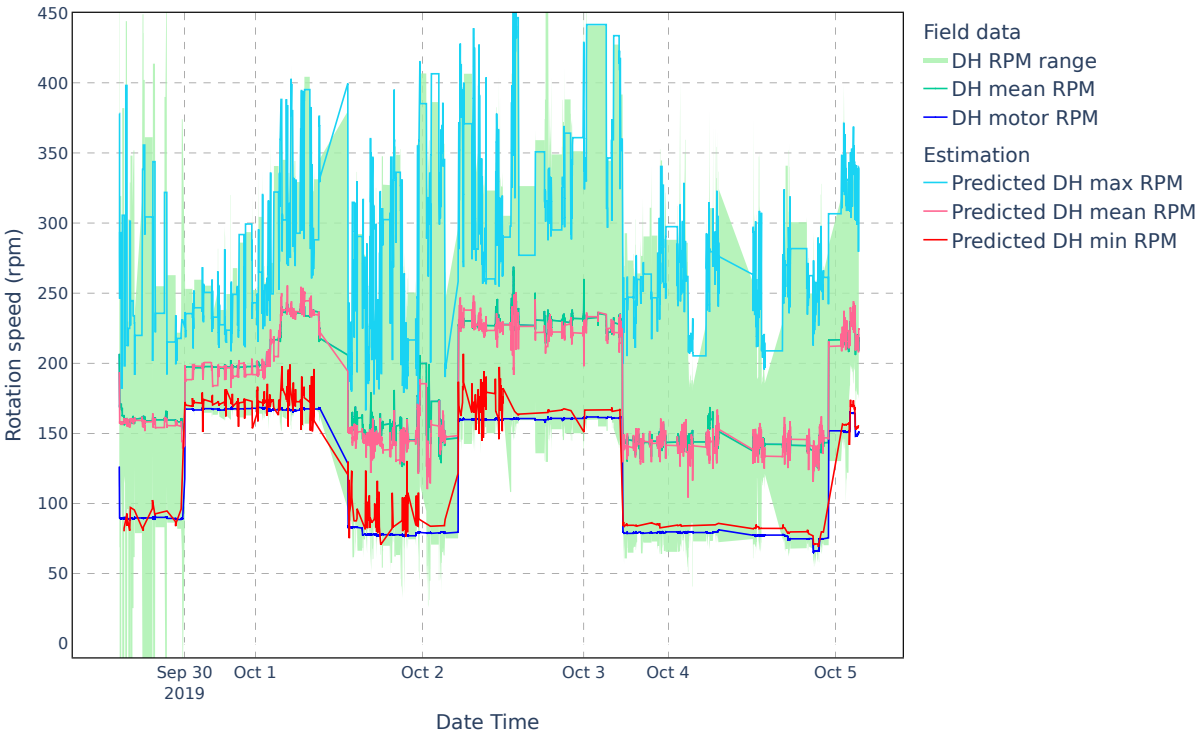


Figure 7.9: Variations of downhole bit rotation speed measurements and predicted $\hat{\omega}_b$ values by the PINN for tets well 1 over six drilling days. The moving minimum, maximum, and average of both the measured and predicted bit rotation speeds, along with the moving average of the downhole motor rotation speed, are calculated over a 120-second window. Definitions of the terms shown: DH RPM range - maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM - mean of measured downhole bit rotation speed; DH motor RPM - measured downhole motor rotation speed; Predicted DH max, mean, and min RPM - moving maximum, mean, and minimum of the predicted downhole bit rotation speed by PINN within the 120-second window.

	Moving max	**Moving mean**	**Moving min**
MAE (rpm)	6.458	1.262	4.402
r	0.994	0.998	0.982

Table 7.3: MAE and Pearson correlation coefficient (r) between the moving maximum, mean, and minimum of true and predicted bit rotation speeds over the long six-day sequences presented in Fig. 7.9 for test well 1.

For the predicted friction coefficients, the uncertainty surrounding the true values makes it

challenging to analyze their variations. Therefore, this part is left for future work.

7.8 Conclusion

In this chapter, a PINN-based method is proposed for estimating unknown parameters and predicting the dynamics of a nonlinear system given partial measurements. Specifically, the PINN is applied to estimate the bit rotation speed as well as the static and dynamic friction coefficients between the drill string and the wellbore in a nonlinear torsional drilling system, using only 1-Hz surface measurements (surface torque and velocity), and without any information about actual or historical downhole measurements.

To assess the robustness of the proposed method, the PINN is tested using two torsional dynamic models of increasing complexity in the loss function: a one-mass and a two-mass lumped model. While the one-mass model demonstrates stable convergence and high accuracy in estimating the friction coefficients, the two-mass model, though physically more representative, proves more challenging to train due to its additional equations and parameters. Nonetheless, it provides valuable insights into the limitations and trade-offs between model fidelity and optimization complexity in PINN-based estimation.

One significant advantage of the PINN approach is its generalizability: it can be applied to any well without limitations, as the training relies solely on surface measurements, whereas traditional physical models require prior knowledge of friction parameters. The method is successfully validated on PDE-based simulated data, where it accurately predicts the bit rotation speed and friction coefficients. Furthermore, the PINN is tested on field drilling data (see Chapter 9 for more results), effectively predicting bit rotation speed variations that are then compared to downhole measurements. Comparative analyses between PINN and other methods is presented in Chapter 9.

Despite these successes, challenges remain. Since true friction coefficients for field data are unknown, analyzing the predicted friction coefficients variations continues to be difficult. Additionally, given the use of 1-Hz surface data, the PINN cannot capture high-frequency variations in bit dynamics. Another limitation is the need to retrain the PINN whenever input or field data change. To address this, a transfer learning technique is implemented, reducing the computational cost substantially and demonstrating promising results by improving the learning of new friction coefficients while decreasing training time by nearly a factor of six.

Chapter 8

Projected Salted Kalman Filter

Résumé du chapitre

Ce chapitre présente le Projected Salted Kalman Filter (PSKF), un observateur conçu pour les systèmes hybrides avec contraintes d'état, telles que celles résultant du frottement de type stiction dans la dynamique du forage. Le PSKF étend le Salted Kalman Filter (SKF) standard en intégrant une étape de projection qui garantit la cohérence avec les contraintes spécifiques au domaine. Ce chapitre évalue d'abord le PSKF sur des données simulées afin de prédire la vitesse de rotation du trépan et les coefficients de frottement à partir de mesures de surface, en utilisant différents modèles torsionnels simplifiés et fréquences de mise à jour. Il analyse ensuite la sensibilité du filtre à la complexité du modèle et à l'initialisation. Enfin, le PSKF est appliqué à des données réelles pour évaluer ses performances pratiques.

Chapter Summary

This chapter introduces the Projected Salted Kalmna Filter (PSKF), an observer designed for hybrid systems with state constraints, such as those arising from stiction-type friction in drilling dynamics. The PSKF extends the standard Salted Kalman Filter (SKF) by integrating a projection step that enforces consistency with domain-specific constraints. This chapter first evaluates the PSKF on simulated data to predict bit rotation speed and friction coefficients using surface measurements, employing various lumped torsional models and update frequencies. It then analyzes the filter's sensitivity to model complexity and initialization. Finally, the PSKF is applied to field data to assess its practical performance.

8.1 Hybrid dynamical systems feature

Hybrid dynamical systems feature both continuous flow and discrete events within a unified framework. As described by Goebel et al. [19], these systems are characterized by the interplay between differential equations that govern continuous evolution and discrete transitions, also called jumps, that modify the system's mode or configuration. This dual nature makes hybrid systems particularly suitable for modeling complex real-world phenomena where continuous processes are interrupted by sudden mode-switching events.

Certain models explicitly separate the state into two components: a continuous state, which evolves over time according to differential equations, and a discrete state, which represent the system's mode. For each mode, the continuous state evolves in a corresponding set. The discrete state changes only during discrete transitions (jumps), while the continuous state x typically evolves continuously but may also experience jumps in certain cases. These types of hybrid systems are often referred to as differential hybrid automata. Typical examples of such systems of varying degrees of complexity include:

- Computer disk drives [226], where reading and writing operations introduce hybrid behavior.
- Electronic circuits, mode switching due to transistors or logic gates leads to discrete transitions superimposed on continuous voltage and current behavior [227, 228].
- Electrical motors, such as stepper and transmission motors [228], where control signals cause abrupt state changes.
- Constrained robotic systems [227], where joint limits, obstacle interactions, contact events, and collisions may produce hybrid dynamics.
- Intelligent transportation systems [229], which involve continuous vehicle dynamics alongside discrete traffic control signals.
- Sampled-data control systems [230], where digital controllers interact with continuous-time plants.
- Switched systems [231], in which system dynamics change according to a switching signal.

In this chapter, focus is placed on a specific class of hybrid automata: mechanical systems affected by stiction-type Coulomb friction (see Section 6.3.2.2). As highlighted in prior works [232, 3], stiction friction model cannot be modeled by differential inclusions, due to the non-uniqueness of Filippov solutions. Hybrid automata, on the other hand, are well-suited to represent them [233, 234]. By explicitly encoding the discrete transitions between sticking and slipping modes, hybrid models can resolve ambiguities and accurately capture the underlying physical phenomena.

Consequently, when designing a state observer for mechanical systems exhibiting stiction friction, it is advantageous to formulate the problem within the hybrid systems framework. This allows the application of parameter estimation and observer design techniques developed specifically for hybrid dynamical systems. Several of these methods are presented and discussed in the following section.

8.2 Review

Several contributions have investigated state estimation for hybrid systems, with some focusing exclusively on switched systems [1] [235, 236, 237], while others [238, 239] concentrate on hybrid systems with state-driven mode transitions [2], which is the main focus of this chapter. The introduction of the hybrid framework [240] has considerably helped the development of generic designs, for both systems with known and unknown jump times. In [241], a hybrid high-gain approach is proposed, while in [242], the authors extend the Kazantzis–Kravaris–Luenberger (KKL) framework to accommodate hybrid systems. Although both methods provide theoretical guarantees of stability, each inherits the limitations associated with its underlying design: the high-gain approach is prone to peaking and sensitivity to measurement noise, whereas the KKL-based design involves significant complexity in constructing the required mappings. Further, the KKL framework is, from a practical perspective, limited to autonomous systems. This is critical in the case of drilling where, often times, measurements of the topside velocity and torque are available, but topside controller parameters are not, which imposes the formulation as a non-autonomous system. For a more complete review of hybrid observers, the reader is referred to [243].

Other contributions focus on the extension of the Kalman filter framework, such as [244] and [245]. There, the reset map is used to update the mean estimate and the Jacobian to adjust the covariance matrices during hybrid events. In more recent developments, the SKF is introduced by Kong et al. [23], which substitutes the Jacobian of the reset map with a saltation matrix in the Kalman filtering process, including an extended Kalman filter variant to handle nonlinear dynamics. This method is further advanced by [246] to include filtering on manifolds, resulting in the creation of the Hybrid Invariant Extended Kalman Filter (HInEKF), which provides enhanced robustness for complex hybrid systems that involve nonlinear dynamics and manifold constraints, and by [247] to account for uncertainties in the structure of the underlying hybrid system, such as variation in guard location and the reset map.

As in a standard Kalman filter, the SKF consists of two parts: a priori update (prediction phase) and a posteriori update (correction phase), with possible hybrid transitions in the two phases between the defined modes. Each mode is associated with a domain that the system can only exit by transitioning to another domain through discrete jumps, governed by so-called *guards*.

A problem may appear when the guards prescribed for the original system do not cover all the boundaries of a domain. This is the case, e.g., for systems with stiction: the guard defining the transition from stick to slip consists of a condition on the external force. Yet, SKF corrections may yield a non-zero velocity, i.e. a corrected state outside of the stick domain, despite no guard being crossed.

This issue is a well-known problem in state estimation for invariant and constrained dynamical systems and is typically addressed using constrained Kalman filtering techniques. Several solutions have been proposed to ensure that known relationships among state variables are satisfied by the updated state estimate. These solutions include system model parameterization reduction [248, 249], optimal Kalman gain restriction [250], and projection of out-of-bound

[1] A switched system is a type of hybrid system composed of multiple subsystems, with a switching logic that determines which subsystem is active at any time.

[2] A state-driven mode transition refers to a change in system dynamics triggered purely by the internal state of the system.

state estimates onto the constraint surface. The projection method has been applied to enforce equality constraints [251, 252, 250] as well as inequality constraints [253, 250], the latter being inherently more complex. Nevertheless, standard quadratic programming techniques can be utilized to address the Kalman filter problem with inequality constraints.

8.3 Objective

The objective of this chapter is to design an SKF observer for mechanical systems exhibiting stiction, with the goal of accurately estimating the bit's rotational speed as well as the dynamic and static friction coefficients. This observer is designed to work with real industrial surface measurements sampled at 1 Hz, a typical rate in drilling operations. To ensure mechanical consistency with the physical stiction model (see Section 6.3.2.2), specifically, to preserve the constraint that the transition from sticking to slipping occurs only when the external force exceeds the static friction threshold, a projection step is proposed to be integrated into the SKF. This step, which is the main contribution of the chapter, guarantees that the updated state vector exits the sticking domain only by satisfying the predefined guard conditions, thereby enforcing a physically valid transition.

The enhanced observer, referred to as the PSKF, is first evaluated against the standard SKF using both one-mass and two-mass lumped torsional dynamic models (see Section 6.3.1). Results show that the PSKF outperforms the original SKF by effectively preventing uncontrolled deviations during the correction step, leading to a more robust and reliable estimation. Building on these findings, the PSKF is tested on simulated data with varying model complexities (two masses or more) to select the optimal physical model to embed within the observer for real data applications. Finally, the chosen PSKF configuration is validated on field data collected from actual drilling operations.

8.4 Salted Kalman Filter

In this section, we recall the design from [23] and illustrate how its application to stiction poses a specific challenge.

8.4.1 Definition

The Salted Kalman Filter (SKF) [23] is an enhanced version of the Kalman filter designed to propagate the estimated state and covariance matrix through hybrid transitions when no intersecting guard conditions are present. A hybrid system is here defined by:

- Set of modes Q: A collection of discrete states, usually integers.
- Domain map D: Associates each mode $q \in Q$ with a set $D(q)$ where the state x evolves (D: $Q \to \mathbb{R}^n$).
- Flow map f: Describes the continuous evolution of x in each mode via differential equations (f_q: $Q \times D_q \to \mathbb{R}^n$).
- Set of discrete transitions Γ: Specifies allowable transitions between modes, identifying pairs (q, q') for possible switches ($\Gamma \subseteq Q \times Q$).

- Guard map G: Defines transition conditions by specifying regions in the state space for mode switches (G: $\Gamma \to \mathbb{R}^n$).
- Reset map R: Determines how the state x updates during mode transitions (R: $\Gamma \times D_q \to D_{q'}$).

Consider a discrete nonlinear hybrid system $\mathscr{H} = (Q, D, f, u, G, R)$ where f defines the nonlinear state dynamics for each mode ($f(q,\cdot) = f_q$), and u represents control inputs. Within each mode $q \in Q$, the state evolution follows:

$$\begin{aligned} \dot{x} &= f_q(x,u) + w_q \quad x \in D_q \\ y &= h_q(x,u) + v_q \end{aligned} \tag{8.1}$$

where x s the state vector, u in the input, y is the measurement, h_q maps the state and input to the output, w_q and v_q are the process and measurement noise, respectively, both assumed to be zero-multivariate gaussian noises with covariance W_q and V_q. The SKF maintains the two-phase structure of the standard Extended Kalman Filter (EKF), consisting of a standard prediction phase followed by a standard correction phase:

- The prediction phase (a priori update) is given by:

$$\begin{aligned} \hat{x}_{k|k-1} &= F_q(\hat{x}_{k-1}, u_k) + w_{k,q} \\ \hat{\Sigma}_{k|k-1} &= A_{k,q}\hat{\Sigma}_{k-1}A_{k,q}^T + W_{k,q} \end{aligned} \tag{8.2}$$

  where k denotes the time index, F_q corresponds to the time-discretized dynamics, e.g. $F_q = I + dt\, f_q$ for a Euler-explicit scheme, $\hat{x}_{k|k-1}$ is the predicted state estimate, $\hat{\Sigma}_{k|k-1}$ is the predicted covariance matrix, $A_{k,q}$ is the Jacobian of the system model evaluated around $\hat{x}_{k|k-1}$, and $W_{k,q}$ is the process noise covariance in mode q. The index k refers to the variable at time k, while $k|k-1$ denotes the variable at time k predicted using the estimate at $k-1$.

- The correction phase (a posteriori update):

$$\begin{aligned} L_k &= \hat{\Sigma}_{k|k-1}C_q^T \left(C_q\hat{\Sigma}_{k|k-1}C_q^T + V_{k,q}\right)^{-1}, \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + L_k\left(y_k - C_q\hat{x}_{k|k-1}\right) \\ \hat{\Sigma}_{k|k} &= \left(I - L_kC_q\right)\hat{\Sigma}_{k|k-1} \end{aligned} \tag{8.3}$$

  where L_k is the filter gain, $C_q = \left.\frac{\partial h(x,u)}{\partial x}\right|_{\hat{x}_{k|k-1}}$ is the Jacobian of the observation function h, $V_{k,q}$ is measurement noise covariance in mode q, $\hat{x}_{k|k}$ is the corrected state estimate, y_k the measurement at time t_k, and $\hat{\Sigma}_{k|k}$ is the corrected estimated covariance. The index $k|k$ refers to the variable at time k estimated using the measurement available at time k.

At each time step, the SKF evaluates whether the current state satisfies the conditions for transitioning from one mode q to another mode $q' \in Q$. This is done by checking the guard conditions associated with the transition. Jumps occur if the estimate belongs to the following set:

$$G_{(q,q')} = \left\{ x \in D_q \mid g_{(q,q')}(t,x) \leq 0, \frac{d}{dt}g_{(q,q')}(t,x(t)) < 0 \right\} \tag{8.4}$$

where $g_{(q,q')}$ is the guard function defining the boundary conditions between the two modes. The condition $g_{(q,q')}(t,x) \leq 0$ ensures the jump condition is active at the boundary, while $\frac{d}{dt}g_{(q,q')}(t,x) < 0$ ensures the guard is transverse, allowing the transition to occur. If the guard conditions are met in the prediction phase, i.e. if $\hat{x}_{k|k-1} \in G_{(q,q')}$ for some q', where $\hat{x}_{k|k-1}$ is the predicted state, the a priori update is modified by applying the reset to the mean and the saltation update to the covariance (Eq. (8.5)) after applying the standard prediction update (Eq. (8.2)).

$$\begin{aligned} \hat{x}_{k|k-1} &= R_{(q,q')}\hat{x}_{k|k-1} \\ \hat{\Sigma}_{k|k-1} &= \Xi_{(q,q')}\hat{\Sigma}_{k|k-1}\Xi^T_{(q,q')} + W_{R_{(q,q')}} \end{aligned} \tag{8.5}$$

where $R_{(q,q')}$ is the reset map, $W_{R_{(q,q')}}$ is the reset process noise covariance, $\hat{x}_{k|k-1}$ and $\hat{\Sigma}_{k|k-1}$ are the updated a priori mean and covariance in the new hybrid domain $D_{q'}$, and $\Xi_{(q,q')}$ is the saltation matrix, the first order approximation of variations at hybrid transitions from mode q to q', evaluated at $\Xi_{(q,q')} = \Xi_{(q,q')}(t_k, \hat{x}_{k|k-1})$ calculated as follow [23, 254]:

$$\Xi_{(q,q')} = D_x R_{(q,q')} + \frac{(f_{q'} - D_x R_{(q,q')}.f_q - D_t R_{(q,q')})D_x g_{(q,q')}}{D_t g_{(q,q')} + D_x g_{(q,q')}.F_q} \tag{8.6}$$

where $\Xi_{(q,q')}$, $R_{(q,q')}$, $g_{(q,q')}$, f_q, and $f_{q'}$ are evaluated at the current time and state:

- $\Xi_{(q,q')} = \Xi_{(q,q')}(t_k, \hat{x}_{k|k-1})$
- $R_{(q,q')} = R_{(q,q')}(t_k, \hat{x}_{k|k-1})$
- $g_{(q,q')} = g_{(q,q')}(t_k, \hat{x}_{k|k-1})$
- $f_q = f_q(t_k, \hat{x}_{k|k-1})$
- $f_{q'} = f_{q'}(t_k, R_{(q,q')}(t_k, \hat{x}_{k|k-1})$

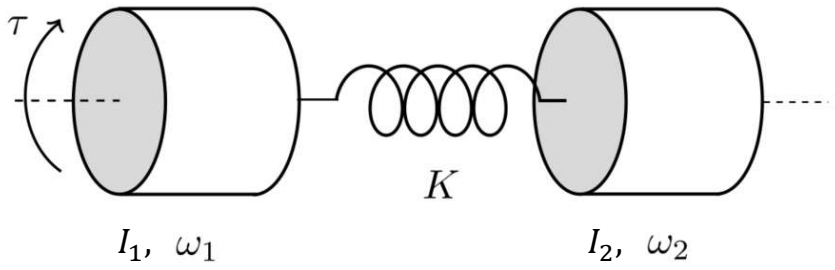
If the measurement update causes the mean estimate to enter a guard set; i.e if $\hat{x}_{k|k} \in G_{(q,q')}$ for some q', where $\hat{x}_{k|k}$ is the corrected state, the a posteriori update is adjusted by first applying the standard correction update in Eq. (8.5), followed by a reset to the mean and a saltation update to the covariance, as follow:

$$\begin{aligned} \hat{x}_{k|k} &= R_{(q,q')}\hat{x}_{k|k} \\ \hat{\Sigma}_{k|k} &= \Xi_{(q,q')}\hat{\Sigma}_{k|k}\Xi^T_{(q,q')} + W_{R_{(q,q')}} \end{aligned} \tag{8.7}$$

where $\hat{x}_{k|k}$ and $\hat{\Sigma}_{k|k}$ are the updated a posteriori mean and covariance in the new hybrid domain $D_{q'}$, and $\Xi_{(q,q')}$ is evaluated at $\Xi_{(q,q')} = \Xi_{(q,q')}(t_k, \hat{x}_{k|k})$

8.4.2 Challenges associated with stiction

We consider a system composed of two rotating masses connected by a spring of stifness K, as depicted in Fig.8.1a. The first mass is subject to a controlled torque τ_t while the second is subject to Coulomb stiction.



(a)

Figure 8.1: Simplified schematic of a two-mass rotational system, with inertia I_1 ans I_2, angular velocities ω_1 and ω_2, external torque τ, connected by a spring with stiffness K.

The system state is represented by the vector:

$$x = (\omega_1, \Delta, \omega_2, \mu_s, \mu_d)$$

where ω_1 and ω_2 represent the angular velocities of the two masses, Δ denotes the angular displacement between them, and μ_s and μ_d correspond to the static and dynamic friction coefficients, respectively, which are included in the state in view of estimating them. The Coulomb stiction friction model presented in Fig. 6.2b is used:

$$F(\omega_2, \mu_d, \mu_s) = \begin{cases} \text{sign}(\omega_2)\, \mu_d\, F^N & \text{if } \omega_2 \neq 0 \\ \text{sign}(K\Delta)\, \mu_s\, F^N & \text{if } \omega_2 = 0 \textbf{ and} \\ & |K\Delta| \geq \mu_s F^N \\ K\Delta & \text{else} \end{cases} \tag{8.8}$$

where F^N is the normal reaction force, assumed constant. This friction law makes the system ill-suited to a representation by differential inclusions [3] due to the existence of multiple Filipov solutions, as explained in Section 6.3.2.2. Similarly to [242], it is represented here using a hybrid automaton with three distinct modes as follows.

- **Slip mode** $q = 1$ **(resp.** $q = -1$**) with** $\omega_2 < 0$ **(resp.** $\omega_2 > 0$**)**: The second mass rotates with a strictly positive (resp. negative) velocity. The corresponding domain D_1 (resp. D_{-1}) is given by

$$D_1 = \{\mathbb{R} \times \mathbb{R} \times]0, +\infty[\times [0,1] \times [0,1]\}$$
$$D_{-1} = \{\mathbb{R} \times \mathbb{R} \times]-\infty, 0[\times [0,1] \times [0,1]\}$$

  with the flow map

$$f_{\pm 1}(x) = \left(\frac{\tau_t - K\Delta}{I_1}, \omega_1 - \omega_2, \frac{K\Delta - sign(\omega_2)\, R\, \mu_d F^N}{I_2}, 0, 0 \right) \tag{8.9}$$

  where I_1 and I_2 are the inertia of the two masses respectively, K is the stiffness of the spring, R the radius of the second mass, and τ_t the torque applied to the first mass.

- **Stick mode** ($q = 0$): The second mass remains stationary ($\omega_2 = 0$). The corresponding domain D_0 is:

$$D_0 = \left\{ \mathbb{R} \times [-\frac{\mu_s}{K}F^N, \frac{\mu_s}{K}F^N] \times \{0\} \times [0,1] \times [0,1] \right\}$$

where the state evolves continuously according to:

$$f_0(x) = \left(\frac{\tau_t - K\Delta}{I_1}, \omega_1, 0, 0, 0 \right) \tag{8.10}$$

Discrete transitions between the three modes are governed by predefined guard functions, which determine the conditions under which jumps occur.

- **Jump from slip phase** ($q = 1$ **or** $q = -1$) **to stick phase** ($q = 0$)**:** This transition occurs when the second mass velocity reaches zero. The corresponding guard function is defined as:

$$g_{(1,0)} = g_{(-1,0)} = \omega_2 \tag{8.11}$$

and the reset map, which updates the system state after the transition, is given by:

$$R_{(1,0)} = R_{(-1,0)} = (\omega_1, \Delta, 0, \mu_s, \mu_d) \tag{8.12}$$

The jump occurs when the guard conditions described in 8.11 are satisfied.

- **Jump from stick phase** $q = 0$ **to slip phase** $q = \pm 1$**:** This transition happens when the external torque applied to the second mass $K\Delta$ exceeds the static friction torque $\pm R\mu_s F^N$. The corresponding guard functions are

$$g_{(0,1)} = R\mu_s F^N - K\Delta, \qquad g_{(0,-1)} = R\mu_s F^N + K\Delta \tag{8.13}$$

with an identity reset map, meaning the state variables remain unchanged.

The objective is to estimate unmeasured states, most notably the second mass rotation speed ω_2, and the friction coefficients, μ_s and μ_d, using only ω_1 as measurement. It is important to note that μ_s influences the system dynamics only during transitions from stick to slip. Consequently, accurately estimating μ_s requires tools capable of handling hybrid dynamics. Moreover, since the system's jump times depend on unmeasured states, classical hybrid observers that assume known jump times are not applicable in this context. The SKF provides a promising alternative for tackling this estimation challenge.

For the SKF, the estimated state vector is given by $\hat{x} = (\hat{\omega}_1, \hat{\Delta}, \hat{\omega}_2, \hat{\mu}_s, \hat{\mu}_d)$ where the estimated friction coefficients must satisfy the following physical constraints:

$$0 \leq \hat{\mu}_d = \hat{r}\hat{\mu}_s \leq \hat{\mu}_s \leq 1 \tag{8.14}$$

with $r = \frac{\mu_d}{\mu_s}$ representing the ratio between the dynamic and static friction coefficients. To enforce these constraints within the SKF, a change of variables is introduced such that the state vector becomes $\hat{x} = (\hat{\omega}_1, \hat{\Delta}, \hat{\omega}_2, \hat{\alpha}, \hat{\beta})$ where $(\hat{\mu}_s, \hat{r}) = \left(\sigma(\hat{\alpha}), \sigma(\hat{\beta})\right)$. The sigmoid function σ is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad \sigma(x) \in [0,1] \quad \forall x \in \mathbb{R}. \tag{8.15}$$

This formulation ensures that $\hat{\mu}_s = \sigma(\hat{\alpha}) \in [0,1]$ and $\hat{r} = \sigma(\hat{\beta}) \in [0,1]$ for all $(\hat{\alpha}, \hat{\beta}) \in \mathbb{R}^2$, which guarantees the conditions (8.14).

Shortcoming of the standard SKF formulation: During the correction phase of the SKF, the predicted state estimate $\hat{x}_{k|k-1}$ is classically updated as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} - L_k(y_k - h(\hat{x}_{k|k-1})) \tag{8.16}$$

where L_k is the filter gain, and h is the observation function linking the predicted state to the measurements. However, if the predicted state $\hat{x}_{k|k-1}$ belongs to the stick phase ($q = 0$), i.e., $\hat{x}_{k|k-1} \in D_0$, the correction term may cause the corrected state $\hat{x}_{k|k}$ to move outside the domain D_0 into D_1 or D_{-1} without crossing the predefined guard conditions $g_{(0,1)}$ or $g_{(0,-1)}$ (see Fig. 8.2), respectively, by violating the equality constraint defining D_0 ($\omega_2 = 0$). While the inequality constraint, $K\Delta \in [-\mu_s F^N, \mu_s F^N]$, is incorporated into the guard function, the equality constraint is not. Consequently, if this equality constraint is violated, the state estimate remains assigned to mode $q = 0$, but its continuous evolution becomes undefined, as the system dynamic f_0 is *a priori* only defined within D_0.

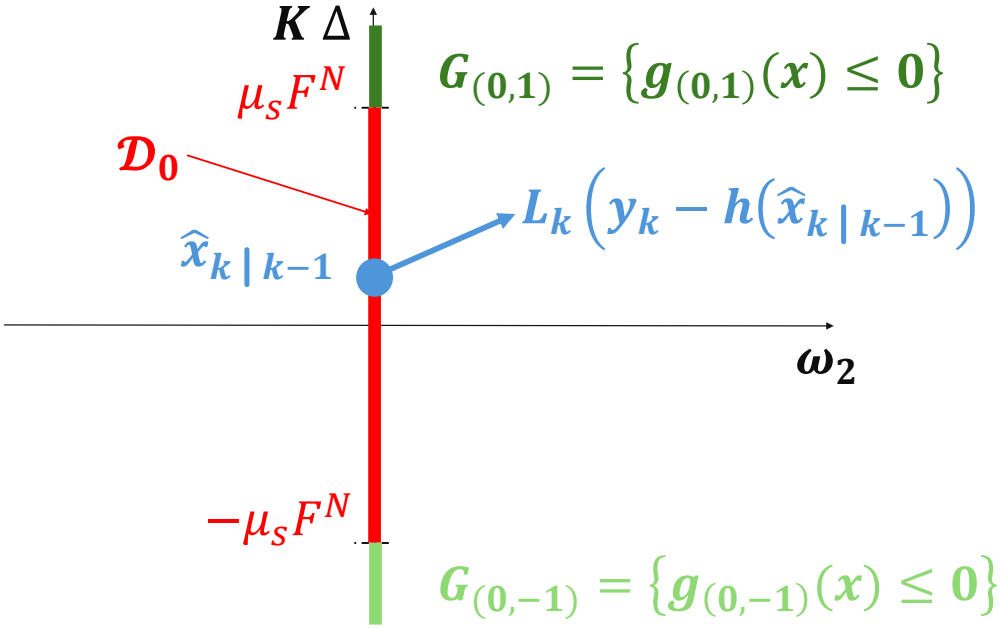


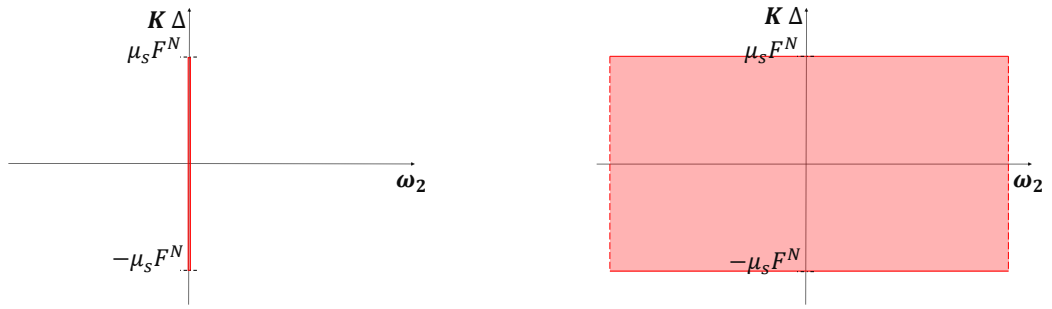
Figure 8.2: Limitation of the SKF: when the predicted state $\hat{x}_{k|k-1}$ lies in the stick phase ($q = 0$), i.e., $\hat{x}_{k|k-1} \in D_0$, the correction term may push the updated state outside D_0 into D_1 or D_{-1} without crossing the corresponding guard conditions $g_{(0,1)}$ or $g_{(0,-1)}$. This occurs through a violation of the equality constraint defining D_0 ($\omega_2 = 0$).

To address this issue, two approaches are possible. The first option involves extending f_0, $g_{(0,1)}$ and $g_{(0,-1)}$ outside of D_0 as presented in Fig. 8.3. In the case of stiction, this could be made allowing ω_2 to evolve in all of $\mathbb{R}$ while still in 'stick' mode, until a sufficient external force is applied (i.e. until $K\Delta > \mu_s F^N$). This poses several issues :

- There is no systematic way to extend the functions outside of their domains of definition. In the case of sticking, f_0 could simply be extended to all of $\mathbb{R}$ but such a trivial extension may not always exist.

- Allowing the estimated bottom velocity to be non-zero while still technically in the 'stick' mode complicates the physical interpretation.

As an alternative, a projection step is proposed to be applied after each correction phase, as detailed in the next section. The merits of this approach are then illustrated by comparing its performance with that of the first solution (extension of f_0) in simulations.



(a) D_0 without extension

(b) D_0 with extension

Figure 8.3: Variability domain of ω_b in the stick phase: (a) Without extension of f_0: $\omega_2 = 0$ until the external force $K\Delta$ surpasses $\mu_s F^N$; (b) With extension of f_0: ω_2 can evolve over all of $\mathbb{R}$ due to the correction term, while still in the 'stick' mode, until $K\Delta > \mu_s F^N$.

8.5 The Projected Salted Kalman Filter

The Projected Salted Kalman Filter can be conceptually summarized as follows:

- Run the standard SKF.
- Check whether the corrected state estimate $\hat{x}_{k|k}$ crosses any of the predefined guard functions $G_{(q,q')}$, i.e.,

$$\hat{x}_{k|k} \in \bigcup_{q' \in Q} G_{(q,q')},$$

  where q is the current mode. If this condition holds, the system transitions to a new mode.
- If the condition is not met, $\hat{x}_{k|k}$ is projected onto the feasible region of the current domain D_q, which means:
  - If $\hat{x}_{k|k} \in D_q$, then the projection has no effect, as the corrected state estimate is already valid, indicating a reliable estimate.
  - If $\hat{x}_{k|k} \notin D_q$ but still lies within one of the defined domains without crossing any guard function, i.e.,

$$\hat{x}_{k|k} \in \bigcup_{q' \in Q} D_{q'} \setminus D_q \quad \text{and} \quad \hat{x}_{k|k} \notin \bigcup_{q' \in Q} G_{(q,q')}, \tag{8.17}$$

    then the projection intervenes to correct the state and bring it back into the feasible region D_q.

The cases where the domain D_q is defined by linear or nonlinear equality and inequality constraints are now treated separately.

8.5.1 Linear constraints

8.5.1.1 Linear equality constraints

Various methods have been proposed to address the problem of Kalman filtering with linear equality constraints in dynamical systems defined as follows:

$$D\hat{x} = d \tag{8.18}$$

where $\hat{x}$ is the state estimate from the unconstrained Kalman filter, D as a known $m \times n$ constant matrix, and d as a known $m \times 1$ vector. Here m represents the number of constraints, n is the number of states, and we assume $m \leq n$. The matrix D must be full rank to ensure a well-defined constraint. To enforce this constraint at each time step, a common approach is to project the unconstrained estimate $\hat{x}$ onto the constraint surface [251]. This is formulated as the optimization problem:

$$\min_{\tilde{x}} (\tilde{x} - \hat{x})^T M (\tilde{x} - \hat{x}) \quad \text{subject to} \quad D\tilde{x} = d \tag{8.19}$$

where M is a symmetric positive definite weighting matrix, and $\tilde{x}$ is the projected state estimate satisfying the equality constraint, and is defined as an unbiased state estimator of the constrained system for any known symmetric positive definite weighting matrix M. The optimal solution is given by:

$$\tilde{x} = \hat{x} - M^{-1} D^T (D M^{-1} D^T)^{-1} (D\hat{x} - d) \tag{8.20}$$

with the smallest estimation error covariance obtained when $M = \Sigma^{-1}$ where Σ is the covariance matrix of the unconstrained estimate.

8.5.1.2 Linear inequality constraints

For a system governed by a linear inequality constraint:

$$D\hat{x} \leq d \tag{8.21}$$

compliance with the constraint is ensured by projecting the unconstrained state estimate $\hat{x}$ onto the feasible set defined by the constraint. This is achieved by solving the following constrained quadratic optimization problem:

$$\min_{\tilde{x}} (\tilde{x} - \hat{x})^T M (\tilde{x} - \hat{x}) \quad \text{subject to} \quad D\tilde{x} \leq d, \tag{8.22}$$

This is a quadratic programming problem commonly solved using active set methods. These methods handle inequality constraints by treating a subset of them, the active set, as equality constraints, assuming the active constraints are known beforehand [253]. Inactive constraints are ignored when solving the optimization problem. After a solution is obtained, it is verified whether the inequality constraints are satisfied. If not, the solution is adjusted iteratively, starting from the result of the previous iteration and moving towards feasibility until a new active set is determined. In each iteration, the active set consists of inequality constraints associated with non-zero Lagrange multipliers. Ultimately, the solution to inequality constrained problem Eq. (8.22) would also be a solution of equality constrained problem Eq. (8.19).

8.5.2 Nonlinear constraints

8.5.2.1 Nonlinear equality constraints

When nonlinear equality constraints exist among the state variables of a discrete nonlinear dynamic system, they can be expressed as:

$$l(\hat{x}) = d \tag{8.23}$$

where l is a nonlinear function representing the equality constraint between the state variables. A commonly used approach to handle such constraints is the two-step method introduced by [255]. This method first projects the unconstrained state estimate $\hat{x}$ onto the constraint surface, then adjusts the distribution to ensure that its mean satisfies the constraint [256]. Assume there exists a projection function p such that:

$$l(p(\hat{x})) = d \tag{8.24}$$

Then, the projection of the state estimate $\hat{x}$ onto the constraint surface is given by:

$$\tilde{x} = p(\hat{x}) \tag{8.25}$$

$$\tilde{\Sigma} = \Delta p \hat{\Sigma} \Delta p^T \tag{8.26}$$

where $\tilde{x}$ denotes the projected state estimate, Δp is the Jacobian matrix of the projection function p evaluated at $\hat{x}$, and $\hat{\Sigma}$ and $\tilde{\Sigma}$ represent the covariance matrices before and after projection, respectively. Although theoretically simple, this approach has been applied in relatively few studies due to the computational challenges associated with complex nonlinear constraints [257, 258].

8.5.2.2 Nonlinear inequality constraints

For nonlinear inequality constraints, a methodology similar to that used for linear inequality constraints can be applied. Without loss of generality, consider the nonlinear inequality constraint expressed as:

$$l(x_k) \leq d \tag{8.27}$$

This constraint can be linearized around the current state prediction $\hat{x}_{k|k-1}$, yielding:

$$l(\hat{x}_{k|k-1}) + \Delta l \, (\hat{x}_{k|k} - \hat{x}_{k|k-1}) \lessapprox d \tag{8.28}$$

where Δl is the Jacobian matrix of l evaluated at $\hat{x}_{k|k-1}$. Consequently, the nonlinear constraint can be approximated by the following linear inequality [250]:

$$\Delta l \, \hat{x}_{k|k} \lessapprox d + \Delta l \, \hat{x}_{k|k-1} - l(\hat{x}_{k|k-1}) \tag{8.29}$$

8.5.3 PSKF Algorithm

The PSKF algorithm, as described above, is formally outlined in Algorithm 1. Its key distinction from the standard SKF is the inclusion of a projection step in the correction phase. Whenever the corrected state $\hat{x}_{k|k}$, initially $\in D_q$, is pushed out of D_q without crossing any of the predefined

guards (as stated in Eq. (8.17)), a projection is applied to obtain the projected state $\tilde{x}_{k|k}$ using the following formula:

$$\tilde{x}_{k|k} = \hat{x}_{k|k} - \Sigma P^T (P\Sigma P^T)^{-1}(P\hat{x} - p) \quad \text{s.t} \quad P\tilde{x}_{k|k} = p \tag{8.30}$$

with Σ represents the covariance matrix. The choice of the projection formula depends on the type of constraint. Eq. (8.20) is suggested, but other equations can also be utilized depending on the specific problem at hand.

Algorithm 1 Projected Salted Kalman Filter

Input $(t_k, x_k, \Sigma_k, m_k, y_{k+1})$
$\hat{t} \leftarrow t_k, \hat{x} \leftarrow x_k, \hat{\Sigma} \leftarrow \Sigma_k, q \leftarrow m_k$
while $(\hat{t} < t_k + \Delta)$ **do**
    $(\hat{t}^+, \hat{x}) \leftarrow$ integrate $F_q(\hat{t}^+, \hat{x})$
        until $(\hat{t}^+ = t_k + \Delta)$ or $(\exists\, q' s.t. \hat{x} \in G_{(q,q')})$
    $\Delta_1 \leftarrow \hat{t}^+ - \hat{t}, \hat{t} \leftarrow \hat{t}^+$
    $\Sigma_k \leftarrow A_{q,\Delta_1} \hat{\Sigma} A_{q,\Delta_1}^\top + W_q$
    if $\exists\, q' s.t. \hat{x} \in G(q, q')$ **then**
        $\hat{x} \leftarrow R_{(q,q')}(\hat{t}, \hat{x})$
        $\hat{\Sigma} \leftarrow \Xi_{(q,q')} \hat{\Sigma} \Xi_{(q,q')}^\top + W_{R_{(q,q')}}$
        $q \leftarrow q'$
    end if
end while
$K \leftarrow \hat{\Sigma} C_q^\top \left[C_q \hat{\Sigma} C_q^\top + V_q \right]^{-1}$
$\varepsilon_{\text{Dyn}} = y_{k+1} - C_q \hat{x}$
$\hat{x} \leftarrow \hat{x} + K \left[y_{k+1} - C_q \hat{x} \right]$
$\varepsilon_{V_q} = y_{k+1} - C_q \hat{x}$
$\hat{\Sigma} \leftarrow \hat{\Sigma} - K C_q \hat{\Sigma}$
if $\exists\, q' s.t. \hat{x} \in G_{(q,q')}$ **then**
    $\hat{x} \leftarrow R_{(q,q')}(\hat{t}, \hat{x})$
    $\hat{\Sigma} \leftarrow \Xi_{(q,q')} \hat{\Sigma} \Xi_{(q,q')}^\top + W_{R_{(q,q')}}$
    $I \leftarrow J$
else
    $\hat{x} = \hat{x} - \hat{\Sigma} P^T (P \hat{\Sigma} P^T)^{-1}(P\hat{x} - p)$
end if
$W_q = \mu W_q + (1-\mu) \varepsilon_{\text{Dyn}} \varepsilon_{\text{Dyn}}^\top$
$V_q = \mu V_q + (1-\mu)(\varepsilon_{V_q} \varepsilon_{V_q}^\top + C_q \hat{\Sigma} C_q^\top)$
$t_{k+1} \leftarrow \hat{t}, x_{k+1} \leftarrow \hat{x}, \Sigma_{k+1} \leftarrow \hat{\Sigma}, m_{k+1} \leftarrow q$
Return $(t_{k+1}, x_{k+1}, \Sigma_{k+1}, m_{k+1})$

8.6 Adaptive estimation of noise covariance matrices

The performance of the Kalman filter (KF) in estimating dynamic states depends heavily on how the process noise W and measurement noise V covariance matrices are set. Choosing inappropriate values for W and V can significantly degrade the filter's accuracy and may even

cause it to diverge [259, 260]. Traditionally, most Dynamic State Estimation (DSE) studies rely on trial-and-error to manually tune these matrices, assuming they remain constant. An approach that can be unreliable, especially when noise levels vary across applications or users have limited experience. To overcome this, Akhlaghi et al. [261] propose an adaptive method that automatically adjusts W and V at each step of the Extended Kalman Filter (EKF). Specifically, an innovation-based method is used to update W, while a residual-based method adjusts V as follows:

- **Innovation-based adaptive estimation of W:**
  The process noise covariance matrix W is adaptively estimated using the innovation-based method:
  $$W_k = \mu W_{k-1} + (1-\mu)\left(L_k \varepsilon_{\text{Dyn}} \varepsilon_{\text{Dyn}}^T L_k^T\right) \tag{8.31}$$
  where the innovation ε_{Dyn} is defined as:
  $$\varepsilon_{\text{Dyn}} = \left[y_{k+1} - h(\hat{x}_{k+1|k}, u_{k+1})\right] \tag{8.32}$$
  This represents the difference between the actual measurement y_{k+1} and the predicted measurement using the **prior** state estimate. L_k is the Kalman gain at time k.

- **Residual-based adaptive estimation of V:**

  The estimation of the measurement noise covariance matrix V follows the residual-based adaptive method proposed by [261, 262], and is computed as:
  $$V_k = \mu V_{k-1} + (1-\mu)\left(\varepsilon_{V_k} \varepsilon_{V_k}^T + H_k \Sigma_k H_k^T\right) \tag{8.33}$$
  where the residual ε_{V_k} is defined as:
  $$\varepsilon_{V_k} = \left[y_{k+1} - h(\hat{x}_{k+1|k+1}, u_{k+1})\right] \tag{8.34}$$
  It represents the difference between the actual measurement y_{k+1} and the estimated output based on the **posterior** state estimate at time $k+1$. Here, $H_k = \left.\frac{\partial h(x_k, u_k)}{\partial x}\right|_{x_k = \hat{x}_{k|k}}$, μ is the forgetting factor that controls how much weight is given to past estimates. A larger μ results in smoother updates but slower adaptation to changes [261].

In our case, the proposed adaptive strategies are employed to update the dynamic and measurement noise covariance matrices. These updates are implemented in lines 27 and 28 of Algorithm 1, for both SKF and PSKF, where the forgetting factor μ is set to 0.5 for the remainder of this chapter. This value provides a moderate compromise between sensitivity to new data and retention of historical information, ensuring smooth and robust updates of the covariance matrices.

Note: Since the static friction coefficient is observable only during the jump from stick phase to slip phase, all the test sequences presented in this chapter exhibit severe stick-slip behavior, unless otherwise specified. This ensures that the observer can effectively estimate the static friction coefficient.

8.7 Comparison of SKF and PSKF

The objective of this section is to compare the performance of the SKF and PSKF on simulated data. Two test cases are presented: the first involves a one-mass lumped torsional dynamic model embedded in both the SKF and PSKF, the second considers a two-mass lumped torsional dynamic model. Both models incorporate a stiction friction model, as described in Section 6.4. For each case, simulated sequences are generated using the same physical model that is embedded in the observer (either SKF or PSKF). The simulations are based on a lateral test well (Well 7 in Table 3.1), whose trajectory is illustrated in Fig. 3.7. The equivalent inertia, stiffness, and normal contact force for both models are detailed in Tables 8.1 and 8.2, respectively. As explained in Section 6.4, for the two-mass model, the drill string is divided into two masses: one representing the vertical section and the other the lateral section. For this test well, the length of the first mass is 15 206 ft, and that of the second mass is 3 800 ft.

Parameter	I_{Top} (kg.m²)	I (kg.m²)	K (N.m)	F^N (N)
Value	2900	890	240	61480

Table 8.1: Parameter values of the one-mass lumped torsional model used in Case 1 Eq. (6.7).

Parameter	I_{Top} (kg.m²)	I_1 (kg.m²)	I_2 (kg.m²)	K_1 (N.m)	K_2 (N.m)	F_1^N (N)	F_2^N (N)
Value	2900	703.89	186.66	300.26	1198.62	18355.10	43125.13

Table 8.2: Parameter values of the two-mass lumped torsional model used in Case 2 Eq. (6.8).

In both models, the torque generated at the bit due to rock cutting is neglected, under the assumption that the bit is off-bottom ($T^b = 0$). Since only surface measurements, namely the measured torque τ_t and the applied top angular speed ω_t, are available during drilling, the objective in both cases is to estimate the downhole angular speed ω_b and the friction coefficients μ_s and μ_d from these observations.

8.7.1 Case 1: One-mass lumped torsional dynamic model

The state vector for the one-mass model is defined as $x = (\omega_t, \Delta, \omega_b, \mu_s, \mu_d)$ with dynamics governed by Eq. (6.7). The domains and guard functions governing the one-mass drilling stick-slip dynamics are consistent with those described in Section 8.4.2, where ω_1 corresponds to ω_t and ω_2 to ω_b. In this setting, transitions between stick ($q = 0$) and slip ($q = \pm 1$) modes, are controlled by the defined guard functions, with a projection step applied to ensure mechanical consistency with the physical stiction model as outlined in Algorithm 1 for PSKF.

8.7.2 Case 2: Two-mass lumped torsional dynamic model

For the two-mass model, the state vector is defined as $x = (\omega_t, \Delta_1, \omega_1, \Delta_2, \omega_b, \mu_s, \mu_d)$, with dynamics governed by Eq. (6.8). As discussed in Section 6.4, only the bit is allowed to transition between the three modes (stick and slip with $q = 0, \pm 1$ respectively). The first mass, representing the top part of the drill string, is assumed to remain continuously in the slip phase. The domains and guard functions governing the two-mass drilling stick-slip dynamics are described as follows:

- **Slip mode** $q = 1$ **(resp.** $q = -1$**) with** $\omega_b > 0$ **(resp.** $\omega_b > 0$**)**: The lateral section rotates with a strictly positive (resp. negative) velocity. The corresponding domain D_1 (resp. D_{-1}) is given by

$$D_1 = \{\mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times]0, +\infty[\times [0,1] \times [0,1]\}$$
$$D_{-1} = \{\mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times]-\infty, 0[\times [0,1] \times [0,1]\}$$

with the flow map

$$f_{\pm 1}(x) = \left(\frac{\tau_t - K_1\Delta_1}{I_1}, \omega_t - \omega_1, \frac{-C_1\omega_1 + K_1\Delta_1 - K_2\Delta_2 - R_1\mu_d F_1^N}{I_1}, \right.$$
$$\left. \omega_1 - \omega_b, \frac{K_2\Delta_2 - \text{sign}(\omega_b)R_2\mu_d F_2^N}{I_2}, 0, 0 \right)$$

where I_1 and I_2 are the inertias, K_1 and K_2 are the stiffness coefficients, R_1 and R_2 denote the radius of the two masses, respectively. F_1^N and F_2^N represent the equivalent normal contact forces along the vertical and lateral sections, respectively, and τ_t is the torque measured at the first surface.

- **Stick mode** $(q = 0)$: The lateral section remains stationary $(\omega_b = 0)$. The corresponding domain is $D_0 = \left\{ \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [-\frac{\mu_s}{K_2}F^N, \frac{\mu_s}{K_2}F^N] \times \{0\} \times [0,1] \times [0,1] \right\}$ where the state evolves continuously according to:

$$f_0(x) = \left(\frac{\tau_t - K_1\Delta_1}{I_1}, \omega_t - \omega_1, \frac{-C_1\omega_1 + K_1\Delta_1 - K_2\Delta_2 - R_1\mu_d F_1^N}{I_1}, \right.$$
$$\left. \omega_1, 0, 0, 0 \right)$$

Discrete transitions between the three operational modes of the lateral section are governed by the predefined guard functions.

- **Jump from slip phase** $(q = 1$ **or** $q = -1)$ **to stick phase** $(q = 0)$**:** This transition occurs when ω_b reaches zero. The corresponding guard function is defined as:

$$g_{(1,0)} = g_{(-1,0)} = \omega_b \tag{8.35}$$

and the reset map is given by:

$$R_{(1,0)} = R_{(-1,0)} = (\omega_t, \Delta_1, \omega_1, \Delta_2, 0, \mu_s, \mu_d) \tag{8.36}$$

- **Jump from stick phase** $q = 0$ **to slip phase** $q = \pm 1$**:** This transition happens when the external torque applied to the lateral section $K_2\Delta_2$ exceeds the static friction torque $\pm R_2\mu_s F_2^N$. The corresponding guard functions are

$$g_{(0,1)} = R_2\mu_s F_2^N - K_2\Delta_2, \qquad g_{(0,-1)} = R_2\mu_s F_2^N + K_2\Delta_2 \tag{8.37}$$

with an identity reset map.

8.7.3 Results and discussions

As discussed in section 8.4.2, two approaches are tested to address the issue of violating the equality constraint $\hat{\omega}_b = 0$ during the correction phase without crossing the predefined guard functions. The first approach, using the standard SKF, involves extending D_0 to $\tilde{D}_0$ in order to maintain the feasibility of f_0 when the equality constraint is violated. The extended domains are defined as:

$$\tilde{D}_0 = \left\{ \mathbb{R} \times [-\frac{\mu_s}{K}F^N, \frac{\mu_s}{K}F^N] \times \mathbb{R} \times [0,1] \times [0,1] \right\}$$

for the one-mass model and

$$\tilde{D}_0 = \left\{ \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [-\frac{\mu_s}{K_2}F^N, \frac{\mu_s}{K_2}F^N] \times \mathbb{R} \times [0,1] \times [0,1] \right\}$$

for the two-mass model.

The second approach, using the PSKF, incorporates a projection step during the correction phase whenever the updated state estimate $\hat{x}_{k|k}$ violates the equality constraint without crossing the predefined guard functions. This projection is formulated as in Eq. (8.20), where $D = [0,0,1,0,0]$ for Case 1 and $D = [0,0,0,0,1,0,0]$ for Case 2, with $d = 0$ and $M = \hat{\Sigma}^{-1}_{(k|k)}$, where $\hat{\Sigma}_{(k|k)}$ denotes the updated covariance estimate.

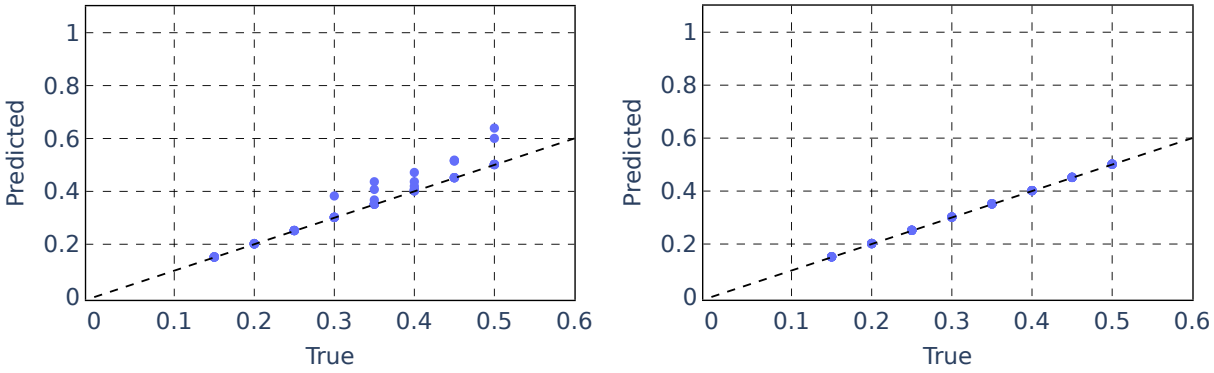
Both the one-mass and two-mass lumped torsional dynamic models are used consistently for simulation and estimation. Simulation data are generated using each model respectively, and then the SKF and PSKF algorithms estimate the downhole rotation speed and friction coefficients based on the same embedded model. A total of 64 simulation sequences are produced by sampling friction coefficients within the ranges $\mu_s \in [0.15, 0.5]$ and $\mu_d \in [0.075, 0.425]$. In both SKF and PSKF implementations, the prediction step is executed at a frequency of 1000 Hz, while the update step occurs at 1 Hz, consistent with typical industrial sampling rates for surface measurements.

Friction coefficients estimation results for the two cases, are presented in Table. 8.3 and Figs. 8.4 to 8.7. As observed, PSKF significantly improves accuracy, reducing errors by over 80% compared to extension approach with SKF in the two cases. This improvement highlights the effectiveness of the projection step in enforcing the equality constraint and reducing estimation errors.

	Case 1				**Case 2**			
	MAE		**MAPE (%)**		**MAE**		**MAPE (%)**	
	SKF	**PSKF**	**SKF**	**PSKF**	**SKF**	**PSKF**	**SKF**	**PSKF**
μ_s	14.1×10^{-3}	1.72×10^{-3}	3.66	0.62	56.1×10^{-3}	11.1×10^{-3}	17.24	3.77
μ_d	7.2×10^{-3}	0.06×10^{-3}	2.17	0.03	23.8×10^{-3}	6.3×10^{-3}	9.98	2.93

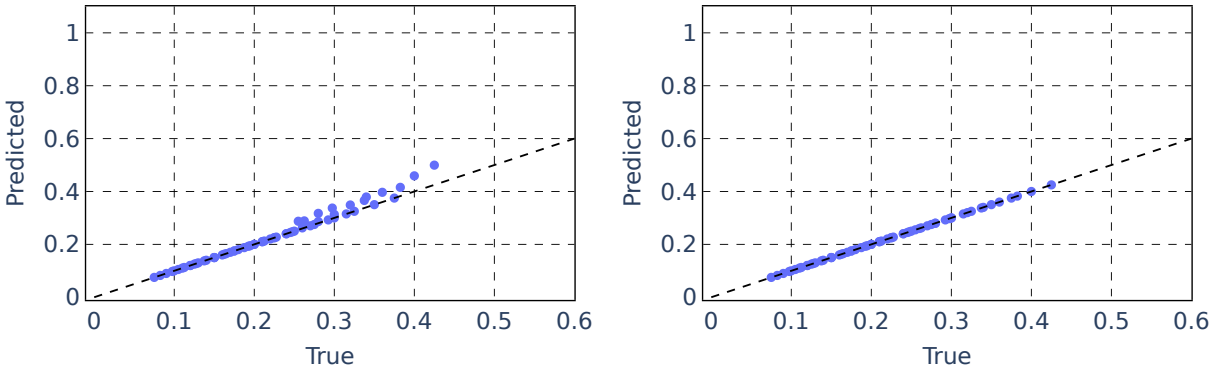
Table 8.3: MAE and MAPE of the friction coefficients estimated using SKF and PSKF across all test sequences of the two cases: Case 1: using the one-mass lumped torsional dynamic model; Case 2: using the two-mass lumped torsional dynamic model.

Figs. 8.8 to 8.10 show the estimated and true values of ω_b and the friction coefficients for both cases for a test sequence. In these particular sequences, the correction term causes ω_b to move outside the domain D_0 to D_1 without crossing the guard function $g_{(0,1)}$. When f_0 is extended to $\tilde{D}_0$, the system continues to evolve according to the same dynamics as f_0



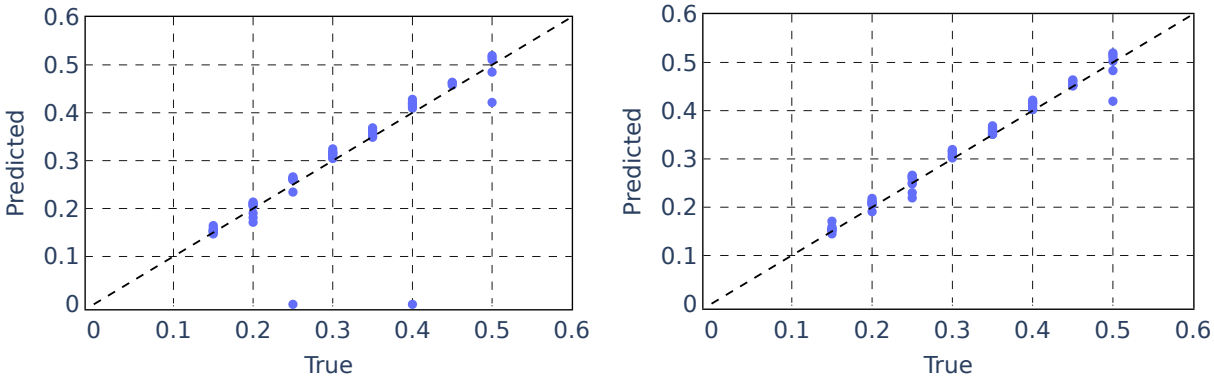
(a) SKF (b) PSKF

Figure 8.4: Estimated and true static friction coefficient using SKF and PSKF in Case 1,with the one-mass lumped torsional dynamic model.



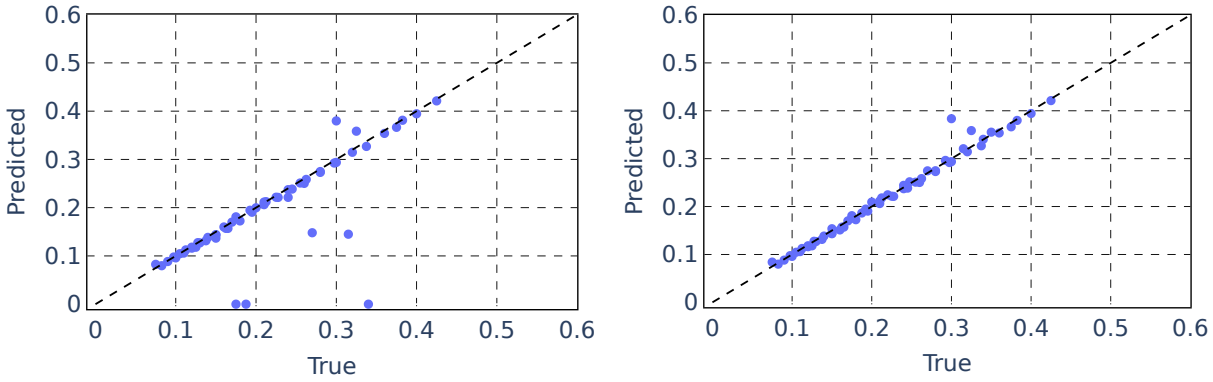
(a) SKF (b) PSKF

Figure 8.5: Estimated and true dynamic friction coefficient using SKF and PSKF in Case 1, with the one-mass lumped torsional dynamic model.



(a) SKF (b) PSKF

Figure 8.6: Estimated and true static friction coefficient using SKF and PSKF in Case 2, with the two-mass lumped torsional dynamic model. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences.

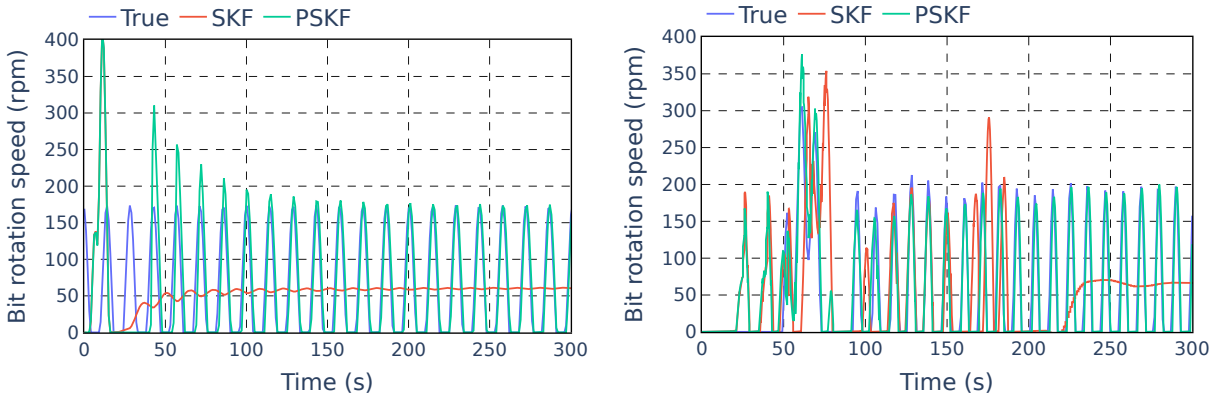


(a) SKF (b) PSKF

Figure 8.7: Estimated and true dynamic friction coefficient using SKF and PSKF in Case 2, with the two-mass lumped torsional dynamic model. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences.

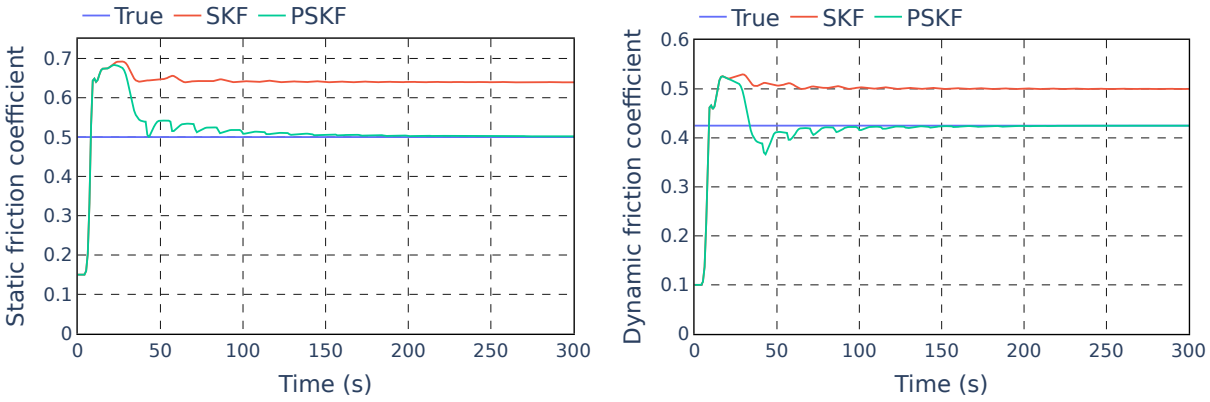
throughout the entire time interval [20 s,300 s] for Case 1 and [225 s,300 s] for Case 2, with the mode value remaining zero, despite the bit velocity being nonzero. During this period, the SKF fails to trigger the predefined guard functions $g_{(0,1)}$ or, $g_{(0,-1)}$ preventing the transition to the slip phase. Consequently, it does not converge to the correct friction coefficients values. On the other hand, the PSKF projects the corrected state each time the correction term pushes it outside D_0, ensuring transitions only occur upon crossing guard functions. This highlights SKF's limitations in handling domain constraints, reinforcing the importance of the projection step in PSKF for maintaining physical consistency.

Note: *Throughout the remainder of this chapter, the PSKF described in Algorithm 1 is used to predict the downhole rotation speed and friction coefficients, with adaptive updates of the noise covariance matrices.*



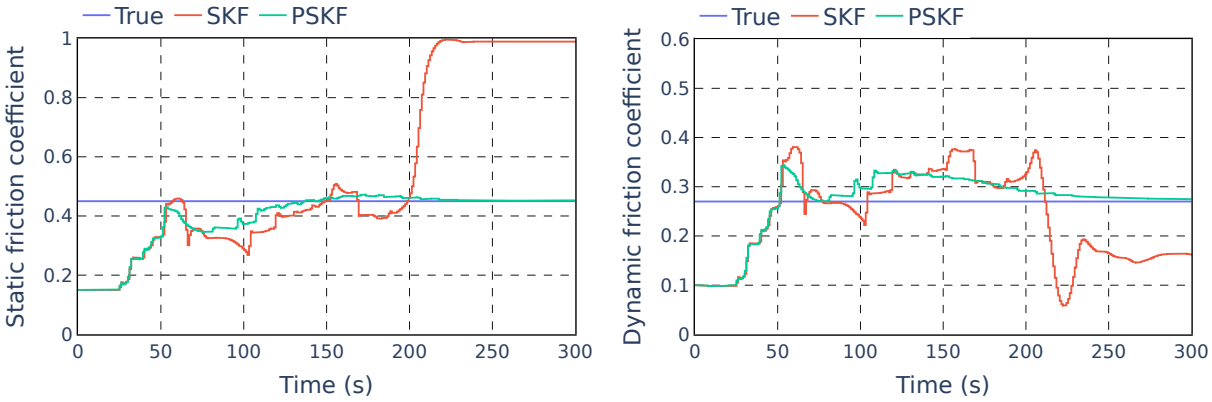
(a) Case 1 (b) Case 2

Figure 8.8: Comparison of estimated and true bit rotation speed using SKF and PSKF for a test sequence in both cases: (a) Case 1: using the one-mass lumped torsional dynamic model; (b) Case 2: using the two-mass lumped torsional dynamic model.



(a) μ_s (b) μ_d

Figure 8.9: Comparison of estimated and true friction coefficients using SKF and PSKF for a test sequence in Case 1.



(a) μ_s (b) μ_d

Figure 8.10: Comparison of estimated and true friction coefficients using SKF and PSKF for a test sequence in Case 2.

8.8 PSKF validation on complex torsional physical models: Multi-mass and PDE Cases

The primary objective of this chapter is to develop an observer capable of predicting the bit rotation speed and friction coefficients using 1-Hz surface measurements as input, and to evaluate its performance on real data. To achieve this, it is crucial to carefully select the physical model to be embedded within the PSKF observer. Two cases are first tested using simulated data:

- **Case 1:** Sequences are generated using a multi-mass lumped torsional dynamic model. The PSKF is then used to estimate the bit rotation speed and friction coefficients from these sequences, using the same physical model as the one used for data generation.

- **Case 2:** Sequences are generated using a PDE-based torsional physical model, while a simplified lumped torsional model is embedded within the PSKF to estimate the bit

rotation speed and friction coefficients. This setup allows us to evaluate the effect of model mismatch on estimation performance.

8.8.1 Case 1: PSKF with identical multi-mass model for generation and estimation

In this case, the objective is to analyze the influence of the complexity of the physical model on the performance of the PSKF. Specifically, the effect of increasing the number of masses in the lumped torsional model on the observer's ability to estimate ω_b, μ_s, and μ_d is investigated. To this end, four different lumped torsional models with increasing levels of complexity are tested: one-mass, two-mass, three-mass, and four-mass systems. For each case, the same model is used for data generation and as the embedded model within the PSKF.

In addition to model complexity, the relationship between the measurement update frequency and the performance of the observer is also studied. By varying the update rate, the aim is to understand how the complexity of the embedded physical model interacts with the measurement frequency, and to what extent estimation accuracy is impacted.

8.8.1.1 Multi-mass lumped torsional models with hybrid stick-slip dynamics

In Section 6.4, the one-mass and two-mass lumped torsional dynamic models are presented in Eq. (6.7) and Eq. (6.8) respectively. The three-mass and four-mass lumped torsional dynamic models are constructed by dividing the drill string into three and four segments, respectively. In both cases, the BHA is modeled as a single mass, while the remaining portion of the drill string is divided into two and three equal segments, respectively. The corresponding physical model is then defined as:

$$
\begin{aligned}
I_{\text{Top}}\ \dot{\omega}_{\text{t}} &= \tau_t - K_1\ \Delta_1 \\
\dot{\Delta}_1 &= \omega_t - \omega_1 \\
I_1\ \dot{\omega}_1 &= -C_1\omega_1 + K_1\Delta_1 - K_2\Delta_2 - R_1\mu_d F_1^N \\
&\vdots \\
\dot{\Delta}_i &= \omega_{i-1} - \omega_i \\
I_i\ \dot{\omega}_i &= -C_i\omega_i + K_i\Delta_i - K_{i+1}\Delta_{i+1} - R_i\mu_d F_i^N \\
&\vdots \\
\dot{\Delta}_N &= \omega_{N-1} - \omega_b \\
I_N\ \dot{\omega}_{\text{Bottom}} &= -C_N\omega_b + K_N\Delta_N - T^f(\omega_b, \mu_s, \mu_d) - T^b(\omega_b + \omega_{\text{Motor}}, \tau_s, \tau_d)
\end{aligned}
\tag{8.38}
$$

where N denotes the number of masses, equal to three for the three-mass model and four for the four-mass lumped torsional dynamic model. I_i, K_i, and C_i represent the equivalent inertia, torsional stiffness, and torsional damping of mass i, respectively, and are calculated according to Eq. (6.1). Δ_i denotes the relative angular displacement between mass $i-1$ and mass i, while ω_i is the angular velocity of mass i. R_i is the outer radius of mass i, and F_i^N is the equivalent normal contact force along mass i. T^f denotes the friction torque applied along the i-th mass, and T^b is the TOB, as defined in Eq. (6.5) and Eq. (6.6) respectively. ω_{Motor} is the downhole

motor rotational speed, ω_t is the applied surface angular velocity, I_{Top} is the inertia of the top drive system, and τ_t is the surface-measured torque.

As we can see, the friction torque for all masses, except the last one, is always equal to $R_i \mu_d F_i^N$, meaning that a purely dynamic (slip-phase) friction model is used. The stiction friction model is integrated only for the last mass, which corresponds to the bit. This modeling choice is made to manage the complexity introduced by increasing the number of masses. Specifically, it is assumed that only the last mass is allowed to transition between the three stick-slip modes, while all other masses remain in the slip phase ($q = 1$). This simplification is motivated by the combinatorial nature of the problem: since each mass can be in one of three modes, the total number of possible mode combinations is 3^N, where N is the number of masses. Allowing all masses to switch modes would dramatically increase the computational burden, making the estimation problem intractable. For this reason, only the last mass is permitted to transition between modes. This approach is consistent with the semi-lumped approximation in [155].

The state vector is then defined as $x = (\omega_t, \Delta_1, \omega_1, \cdots, \Delta_i, \omega_i, \cdots, \Delta_N, \omega_b, \mu_s, \mu_d)$. The domains and guard functions follow the same strategy as those presented in Section 8.7.2, where only the bit is allowed to transition between the three modes ($q = 0$, $q = \pm 1$). The guard functions are defined as:

$$g_{(1,0)} = g_{(-1,0)} = \omega_b$$

with the corresponding reset map:

$$R_{(1,0)} = R_{(-1,0)} = (\omega_t, \Delta_1, \omega_1, \cdots, \Delta_i, \omega_i, \cdots, \Delta_N, 0, \mu_s, \mu_d)$$

and the transition from stick to slip is defined by the following guard functions:

$$g_{(0,1)} = R_N \mu_s F_N^N - K_N \Delta_N \qquad g_{(0,-1)} = R_N \mu_s F_N^N + K_N \Delta_N$$

with an identity reset map.

8.8.1.2 Ablation studies

The PSKF, described in Algorithm 1, is used to predict the bit rotation speed and friction coefficients on simulated data for the four proposed lumped torsional dynamic models. As previously mentioned, for each of the four tests, the same physical model used to generate the data is embedded within the PSKF.

A total of 64 sequences are generated by sampling the friction coefficients within the ranges $\mu_s \in [0.15, 0.5]$ and $\mu_d \in [0.075, 0.425]$, using the four proposed physical models. The parameters for the one-mass and two-mass models are the same as those used in Section 8.7 (see Table 6.7 and Table 8.2, respectively), while the parameters for the three-mass and four-mass models are provided in Table 8.4 and Table 8.5, respectively.

The prediction step of the PSKF is executed at a high internal frequency of 1000 Hz to ensure accurate state propagation. However, to evaluate the observer's robustness under practical constraints, different update frequencies, 1 Hz, 10 Hz, 100 Hz, and 1000 Hz, are tested. This allows us to analyze the impact of measurement frequency on the PSKF's performance across varying levels of physical model complexity.

Friction coefficients estimation results for the PSKF with the four tested lumped torsional dynamic models are presented in Table 8.6 for μ_s and Table 8.7 for μ_d. The initial values of the static and dynamic friction coefficients, μ_s and μ_d, are kept fixed across all test sequences and

Inertia (kg.m²)	I_1 388.64	I_2 426.41	I_3 75.50
Stiffness (N.m)	K_1 498.69	K_2 526.90	K_3 3823.15
Normal contact force (N)	F_1^N 10310.25	F_2^N 24422.78	F_3^N 26747.20
Length (ft)	L_1 8869.97	L_2 8869.97	L_3 1267.13

Table 8.4: Parameter values of the three-mass lumped torsional model Eq. (8.38) with $N = 3$.

Inertia (kg.m²)	I_1 221.98	I_2 315.42	I_3 277.64	I_4 75.50
Stiffness (N.m)	K_1 872.59	K_2 754.78	K_3 698.21	K_4 3823.15
Normal contact force (N)	F_1^N 8369.03	F_2^N 3678.81	F_3^N 22685.19	F_4^N 26747.20
Length (ft)	L_1 5068.55	L_2 6335.69	L_3 6335.69	L_4 1267.13

Table 8.5: Parameter values of the four-mass lumped torsional model Eq. (8.38) with $N = 4$.

physical models. Specifically, the initialization is set to $\mu_s^{\text{Init}} = 0.15$ and $\mu_d^{\text{Init}} = 0.1$, regardless of the true values used to generate the simulated data. For the PSKF with the one-mass and two-mass models, the estimation results are illustrated in Figs. 8.4b, 8.5b, 8.6b and 8.7b. In these cases, an update frequency of 1 Hz is sufficient to accurately estimate the friction coefficients. The low MAE and MAPE values confirm that the PSKF achieves precise predictions, as visually confirmed in the corresponding figures [3]. For the PSKF using the three-mass and four-mass lumped models, as shown in Table 8.6 and Table 8.7, the observer diverges for all test sequences when the update frequency is 1 Hz or 10 Hz. However, it manages to converge at 100-Hz and 1000-Hz update frequencies, although not for all test sequences. Moreover, even when convergence is achieved, the observer sometimes estimates incorrect values, as illustrated in Figs. 8.11 and 8.12. An example of the PSKF's performance using the three-mass lumped torsional dynamic model on a sequence generated with the same physical model is presented in Figs. 8.13 and 8.14.

As shown, for this sequence, the PSKF accurately predicts both the bit rotation speed and the friction coefficients using surface measurements sampled at 100 Hz. Starting from initial values of $\mu_s = 0.15$ and $\mu_d = 0.1$, the observer successfully converges to the true values $\mu_s = 0.35$ and $\mu_d = 0.2$ for the static and dynamic friction coefficients, respectively.

These simulations suggest that the more complex the physical model embedded in the PSKF, the higher the update frequency is required for successful convergence. This may be explained by the increased dimensionality of the state vector: more complex models, such as those with a larger number of masses, involve more states to estimate. Consequently, more measurement updates are needed to accurately infer all state components and friction coefficients. This trend

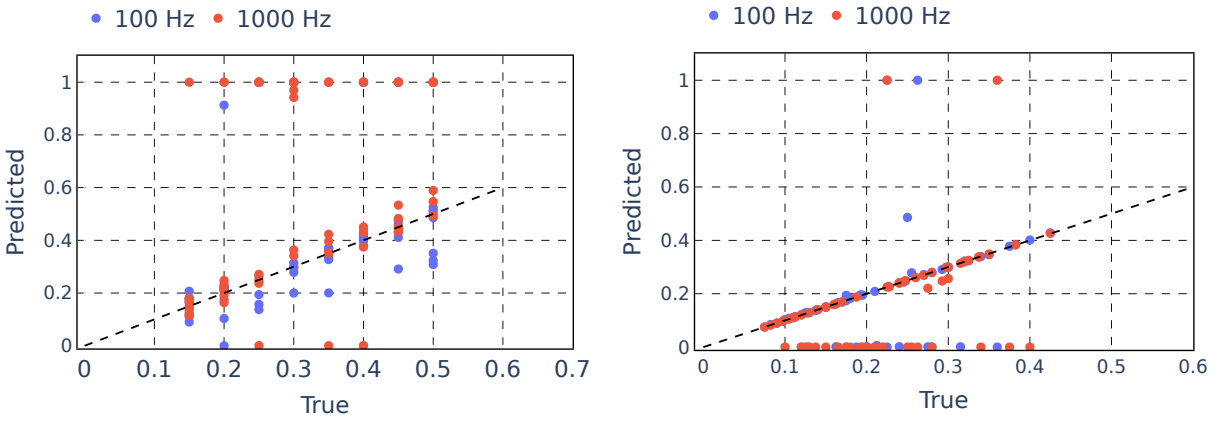
[3] For the one-mass and two-mass lumped models, only a 1-Hz update frequency is tested, as it is sufficient to accurately predict the friction coefficients using the PSKF observer.

Physical model	1 Hz		10 Hz		100 Hz		1000 Hz	
	MAE	MAPE (%)	MAE	MAPE (%)	MAE	MAPE (%)	MAE	MAPE (%)
One-mass	1.72×10^{-3}	0.62	-	-	-	-	-	-
Two-mass	11.1×10^{-3}	3.77	-	-	-	-	-	-
Three-mass	Diverges	Diverges	Diverges	Diverges	197.4×10^{-3}	64.86	300.83×10^{-3}	100.21
Four-mass	Diverges	Diverges	Diverges	Diverges	214.89×10^{-3}	60.46	214.2×10^{-3}	60.28

Table 8.6: PSKF performance on static friction coefficient estimation for different physical models and update frequencies, For the one-mass and two-mass lumped models, only a 1-Hz update frequency is tested, as it is sufficient to accurately predict the friction coefficients using the PSKF observer.

Physical model	1 Hz		10 Hz		100 Hz		1000 Hz	
	MAE	MAPE (%)	MAE	MAPE (%)	MAE	MAPE (%)	MAE	MAPE (%)
One-mass	0.06×10^{-3}	0.03	-	-	-	-	-	-
Two-mass	6.3×10^{-3}	2.93	-	-	-	-	-	-
Three-mass	Diverges	Diverges	Diverges	Diverges	80.8×10^{-3}	35.39	106.81×10^{-3}	48.14
Four-mass	Diverges	Diverges	Diverges	Diverges	144.76×10^{-3}	59.33	111.7×10^{-3}	46.02

Table 8.7: PSKF performance on dynamic friction coefficient estimation for different physical models and update frequencies.

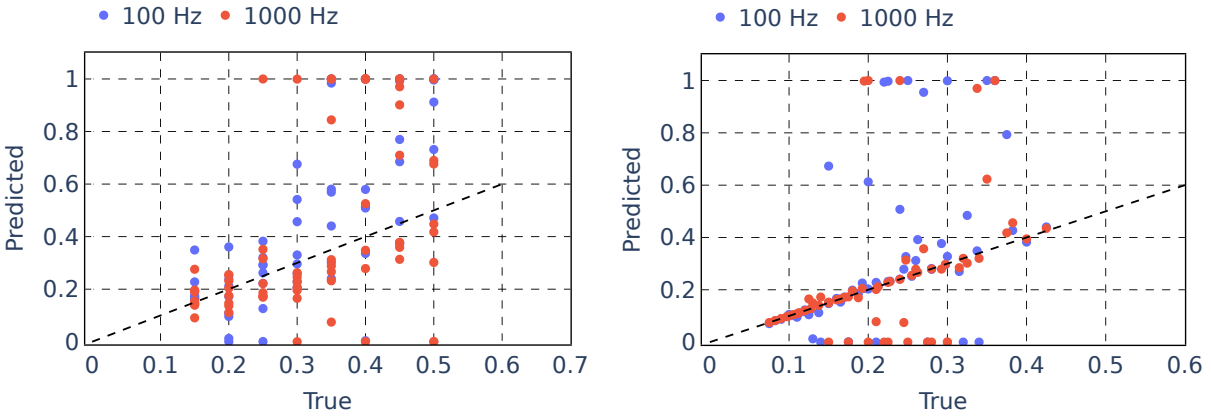


(a) μ_s (b) μ_d

Figure 8.11: Comparison of estimated and true friction coefficients using PSKF with the three-mass lumped torsional dynamic model at two update frequencies: (a) 100 Hz and (b) 1000 Hz. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences.

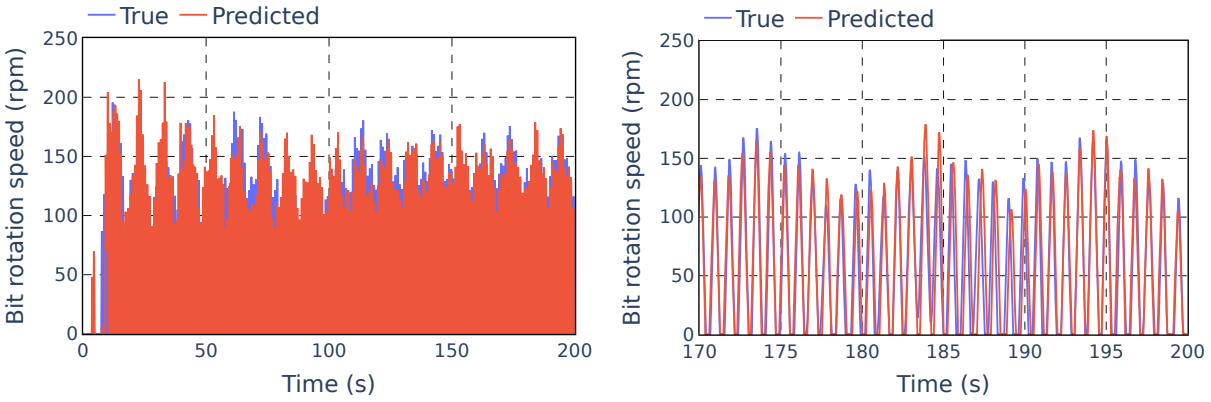
is illustrated by the results: with an update frequency of 1 Hz, the PSKF can accurately estimate the friction coefficients when using the one-mass or two-mass embedded physical models. However, for the three-mass and four-mass models, 1 Hz is insufficient, and a much higher measurement frequency is required to achieve reliable convergence and accurate estimation.

Effect of model complexity on the basin of attraction of the PSKF In addition to evaluating estimation accuracy, the impact of physical model complexity on the PSKF basin of attraction is analyzed. By basin of attraction, we refer to the tendency of the observer to converge only when the true parameter values are sufficiently close to the initial estimates, where poor initialization may lead to divergence or convergence to incorrect values. As shown in Figs. 8.4b, 8.5b, 8.6b and 8.7b, the one-mass model consistently predicts the friction coefficients accurately, with



(a) μ_s (b) μ_d

Figure 8.12: Comparison of estimated and true friction coefficients using PSKF with the four-mass lumped torsional dynamic model at two update frequencies: (a) 100 Hz and (b) 1000 Hz. The initial friction coefficients values are $\mu_s^{\text{Init}} = 0.15$, $\mu_d^{\text{Init}} = 0.1$ for all the sequences.



(a) Full sequence (b) Zoomed-in view

Figure 8.13: Comparison of estimated and true bit angular velocity using PSKF with a three-mass model and 100 Hz update frequency for a test sequence.

fixed initial values ($\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$), for all test sequences. This suggests that the basin of attraction of the PSKF is relatively wide for the one-mass model. For the two-mass model, the basin of attraction appears narrower, as reflected by slightly higher MAE and MAPE values (see Tables 8.6 and 8.7). The situation worsens for the three-mass and four-mass models: when the true friction coefficients deviate significantly from the initial values, the observer often diverges or converges to incorrect estimates (see Figs. 8.11 and 8.12). An example is shown in Figs. 8.15 and 8.16, where the PSKF is applied using a four-mass lumped torsional dynamic model with an update frequency of 1000 Hz. The true and estimated bit rotation speed and friction coefficients are plotted for a test sequence. In particular, Figs. 8.15a and 8.16a show a case where the PSKF is initialized with $\mu_s^{Init} = 0.15$ and $\mu_d^{Init} = 0.1$. In this case, the observer converges correctly to the true value of μ_d, but fails to estimate μ_s, instead converging to an incorrect value $\hat{\mu}_s = \hat{\mu}_d = 0.2$. However, when the same PSKF is re-run on the same sequence with different initial values ($\mu_s^{Init} = 0.3$ and $\mu_d^{Init} = 0.15$), closer to the true values,

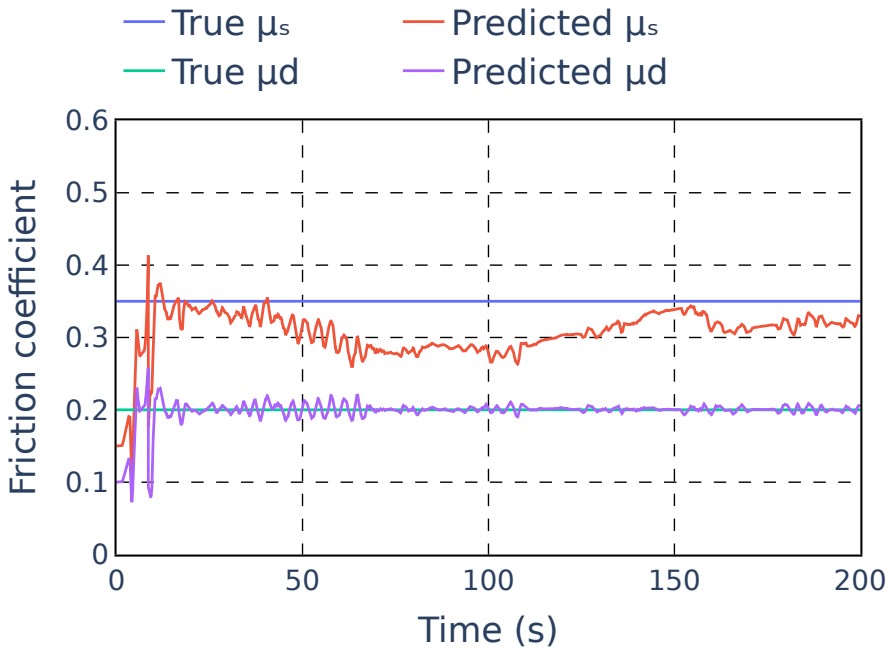


Figure 8.14: Comparison of estimated and true friction coefficients over time using PSKF with three-mass lumped model at 100-Hz update frequency for a test sequence.

and keeping everything else unchanged, the observer successfully converges to the true values of both friction coefficients, as shown in Figs. 8.15c and 8.16b.

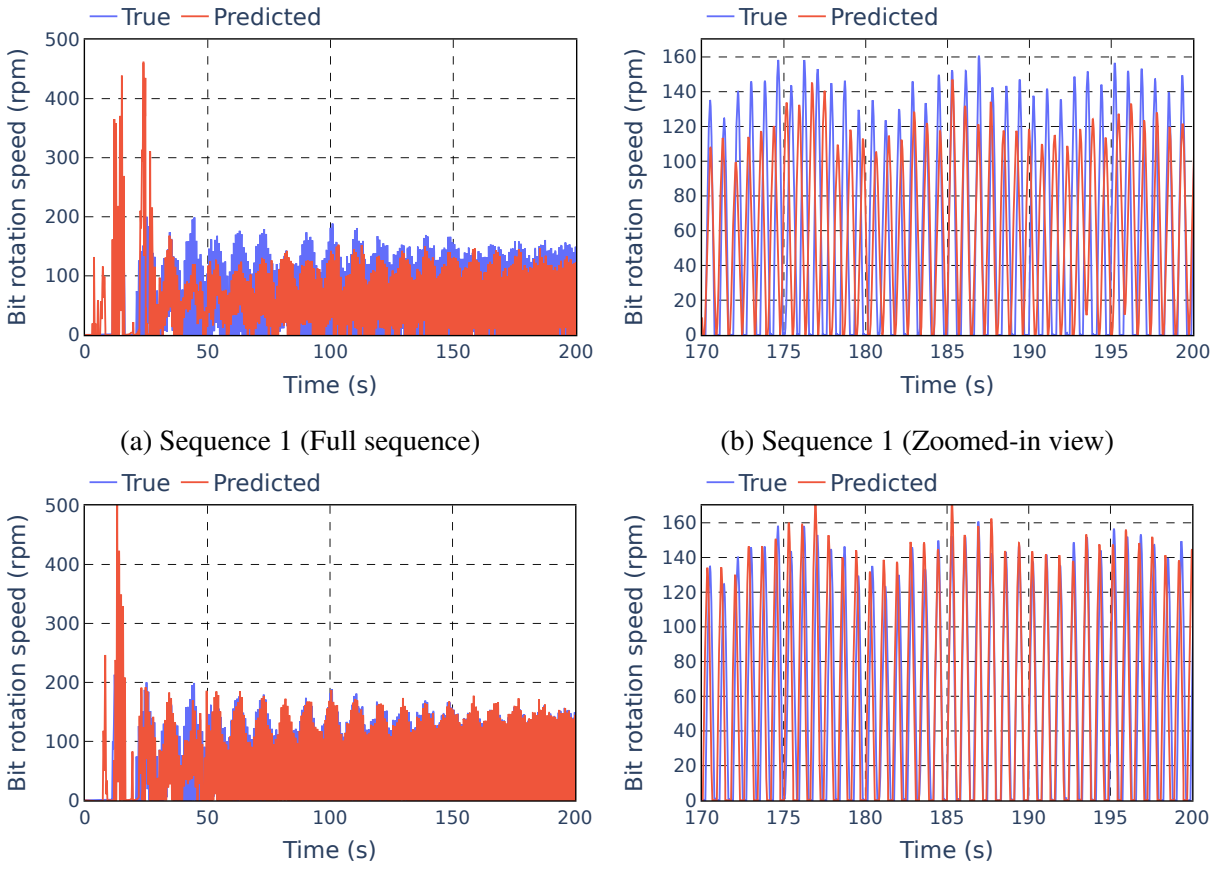
To further investigate the radius of convergence of the PSKF when embedded with each of the four proposed physical models, an experiment is designed involving three distinct test sequences. Each sequence is characterized by different values of static and dynamic friction coefficients: $\{\mu_{s_1} = 0.15,\ \mu_{d_1} = 0.075\}$, $\{\mu_{s_2} = 0.3,\ \mu_{d_2} = 0.165\}$, and $\{\mu_{s_3} = 0.4,\ \mu_{d_3} = 0.24\}$. Each of these three friction configurations is simulated using the four proposed physical models, resulting in a total of 12 test sequences (3 friction settings $\times$ 4 physical models). For the evaluation, a PSKF that is embedded with the same physical model as the one used to generate the corresponding test sequence is used. Specifically, for each combination, the PSKF is used to jointly estimate the friction coefficients (μ_s, μ_d) and the bit rotation speed throughout the test sequence.

For each test sequence, multiple runs are performed using randomly selected initial values for the friction coefficients. These initializations are categorized into three regions as showed in Fig. 8.17:

- Region I: $\mu_s^{Init} \in [0,\ 0.5\,\mu_{s_i}] \cup [1.5\,\mu_{s_i}, 1]$ and $\mu_d^{Init} \in \left[0,\ \mu_s^{Init}\right] \cup \left[1.5\,\mu_{d_i}, \mu_s^{Init}\right]$ where μ_s^{Init} and μ_d^{Init} are the initial values of the friction coefficients. This region corresponds to initial values that are far from the true friction coefficients.

- Region II: $\mu_s^{Init} \in [\,0.5\,\mu_{s_i},\ 0.7\,\mu_{s_i}] \cup [1.3\,\mu_{s_i}, 1.5\,\mu_{s_i}]$ and $\mu_d^{Init} \in \left[0.5\,\mu_{d_i},\ \mu_s^{Init}\right] \cup \left[1.3\,\mu_{d_i}, \mu_s^{Init}\right]$. This region corresponds to initial values that are at a moderate distance from the true friction coefficients.

- Region III: $\mu_s^{Init} \in [\,0.7\,\mu_{s_i},\ 0.9\,\mu_{s_i}] \cup [1.1\,\mu_{s_i}, 1.3\,\mu_{s_i}]$



(a) Sequence 1 (Full sequence)

(b) Sequence 1 (Zoomed-in view)

(c) Sequence 2 (Full sequence)

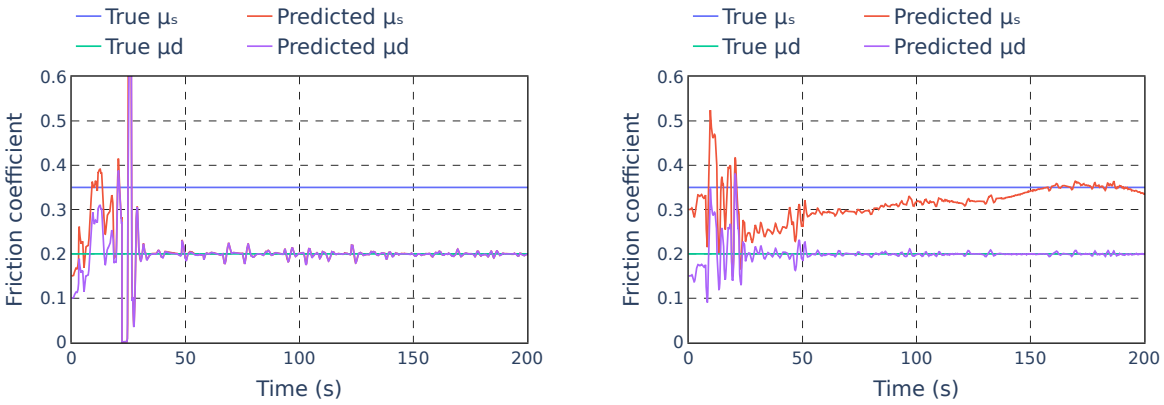
(d) Sequence 2 (Zoomed-in view)

Figure 8.15: Comparison of estimated and true bit angular velocity using PSKF with a four-mass model and a 1000-Hz update frequency. Two sets of initial friction coefficient values are tested for the same sequence: Sequence 1: $\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$, and sequence 2 : $\mu_s^{Init} = 0.3$, $\mu_d^{Init} = 0.15$.

and $\mu_d^{Init} \in [0.7\,\mu_{d_i},\ \mu_{s_{Init}}] \cup [1.1\,\mu_{d_i}, \mu_{s_{\text{Init}}}]$. This region corresponds to initial values that are close to the true friction coefficients.

For each region, 100 runs are conducted per physical model. In each run, the initial values of the friction coefficients are chosen randomly within the region-defined bounds. This controlled setup allows us to evaluate whether the basin of attraction of the PSKF is influenced by the complexity of the embedded physical model. To ensure fair comparison across models, the update frequency is fixed at 1000 Hz for all four physical models. As discussed previously, an update rate of 1 Hz is sufficient for the simpler one-mass and two-mass models to achieve convergence. However, it proves inadequate for the more complex three-mass and four-mass models. Since the objective here is to analyze the effect of the initial friction coefficient values on the convergence behavior of the PSKF, the update frequency and all other parameters are kept consistent across models. This isolates the influence of model complexity and initialization from other confounding factors.

Fig. 8.17 presents the results obtained for the static friction coefficient estimation using the PSKF with each of the four physical models.The analysis is performed on three test sequences with ground-truth friction coefficients $(\mu_s, \mu_d) = (0.15, 0.075)$, $(0.3, 0.165)$, and $(0.4, 0.24)$,



(a) Initial values: $\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$

(b) Initial values: $\mu_s^{Init} = 0.3$, $\mu_d^{Init} = 0.15$

Figure 8.16: Comparison of estimated and true friction coefficients using PSKF with a four-mass lumped torsional dynamic model and 1000-Hz update frequency for a test sequence. The two subfigures show results with different initializations.

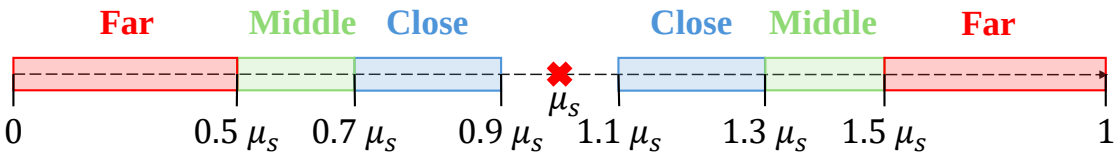


Figure 8.17: The three regions used for generating random initializations of the static friction coefficient values: Region I (Far): $\mu_s^{Init} \in [0, 0.5\,\mu_{s_i}] \cup [1.5\,\mu_{s_i}, 1]$; Region II (Middle): $\mu_s^{Init} \in [0.5\,\mu_{s_i}, 0.7\,\mu_{s_i}] \cup [1.3\,\mu_{s_i}, 1.5\,\mu_{s_i}]$; Region III (Close): $\mu_s^{Init} \in [0.7\,\mu_{s_i}, 0.9\,\mu_{s_i}] \cup [1.1\,\mu_{s_i}, 1.3\,\mu_{s_i}]$. μ_s^{Init} is the initial value. The red cross in the figure represents the true static friction coefficient.

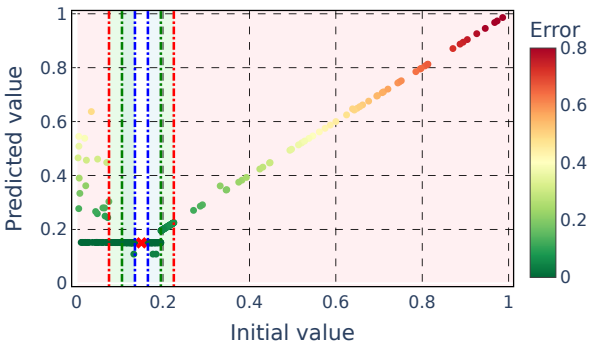
respectively. In each subplot, the red cross denotes the true value of the friction coefficient, while each circle represents one of the 100 independent runs per region. The x-axis shows the initial value of the friction coefficient used to initialize the PSKF, and the y-axis indicates the corresponding predicted value. The background color illustrates the initialization region:

- Red region (Region I): Far from the true value
- Green region (Region II): Moderately distant from the true value
- Blue region (Region III): Close to the true value

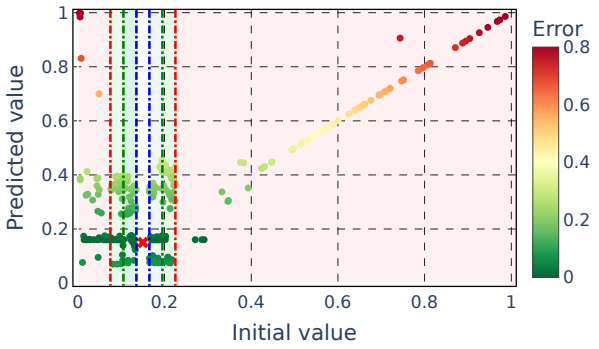
Each circle is also color-coded according to the prediction error, computed as:

$$\text{Error} = |\mu_s - \hat{\mu}_s|$$

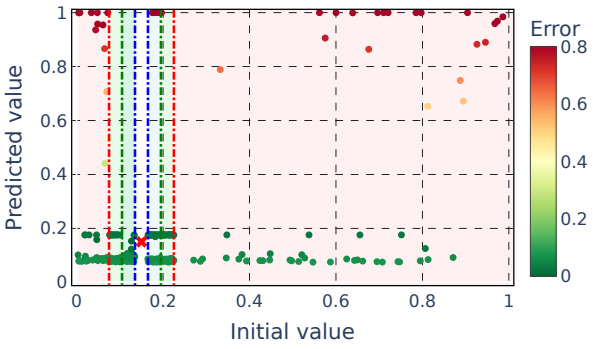
where μ_s is the true value and $\hat{\mu}_s$ is the estimated value of the static or dynamic friction coefficient.



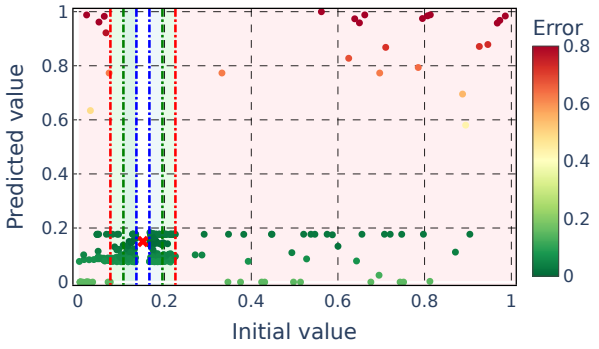
(a) Sequence 1: One-mass



(b) Sequence 1: Two-mass



(c) Sequence 1: Three-mass

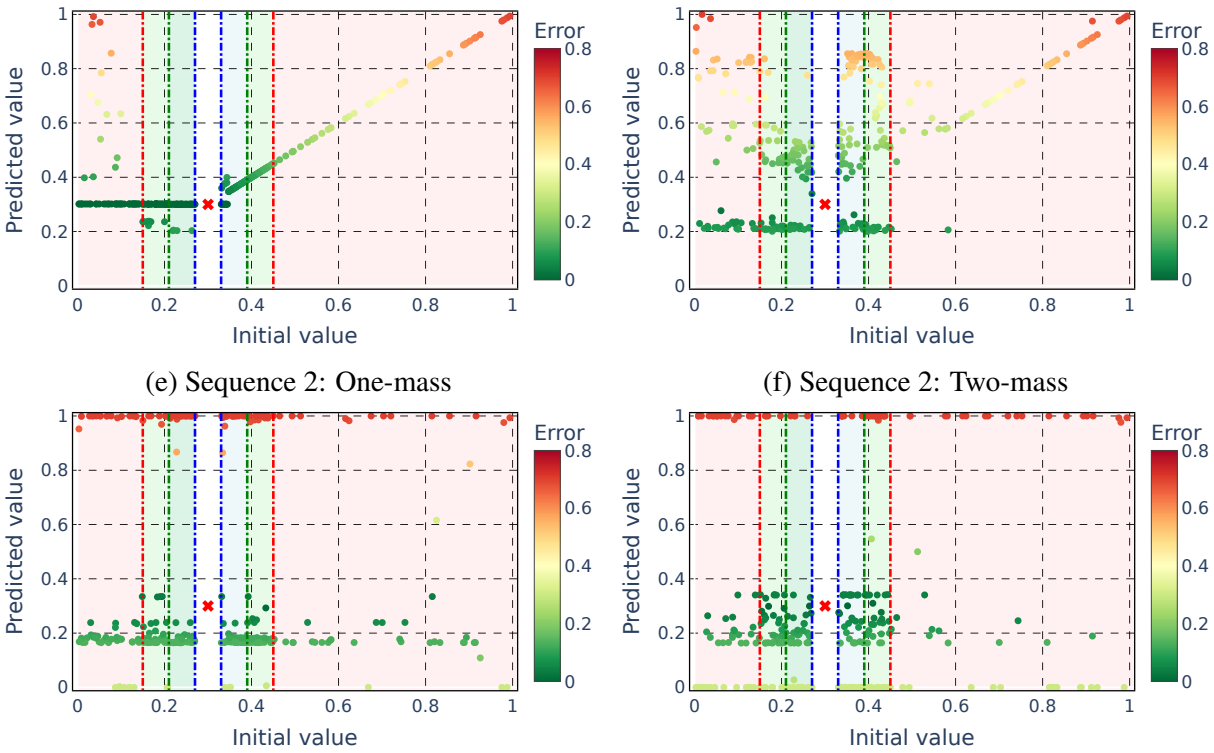


(d) Sequence 1: Four-mass

Figure 8.17

As shown in Fig. 8.17, increasing the complexity of the physical model leads to a greater number of red dots, which represent a high error between the predicted static friction coefficient and its true value, specifically for the three- and four-mass physical models compared to the one-mass model, with the far region exhibiting more red points than the middle and close regions. For the one- and two-mass models, when the initial static friction coefficient is significantly higher than the true value (i.e., in the right red zone), the predicted value remains nearly unchanged. This suggests that the observer fails to correct the static friction estimate and instead retains the initial value. This behavior can be explained by the nature of the test sequences: as previously mentioned, all PDE-generated sequences exhibit severe stick-slip behavior. In the case of the PSKF using a one or two-mass physical models, when the system enters the stick phase, it must satisfy the guard function defined in Eq. (8.37), which depends on the predicted value of the static friction coefficient. If the initial value is too high, the guard threshold becomes more difficult to satisfy, making it harder for the observer to transition to the slip phase. As a result, the observer remains trapped in the stick phase, unable to update its estimate. **This analysis highlights the necessity to initialize the static friction coefficient with a low value to allow the observer to escape the stick phase and improve estimation accuracy.**

Fig. 8.18 summarizes the performance of the PSKF over the three test sequences by presenting the MAE of the estimated static and dynamic friction coefficients. For each sequence, two subplots are shown: one for the static coefficient and another for the dynamic coefficient. In each subplot, the MAE is computed over 100 runs per region (far, middle, close) and plotted for the four physical models (one-mass to four-mass).



(e) Sequence 2: One-mass

(f) Sequence 2: Two-mass

(g) Sequence 2: Three-mass

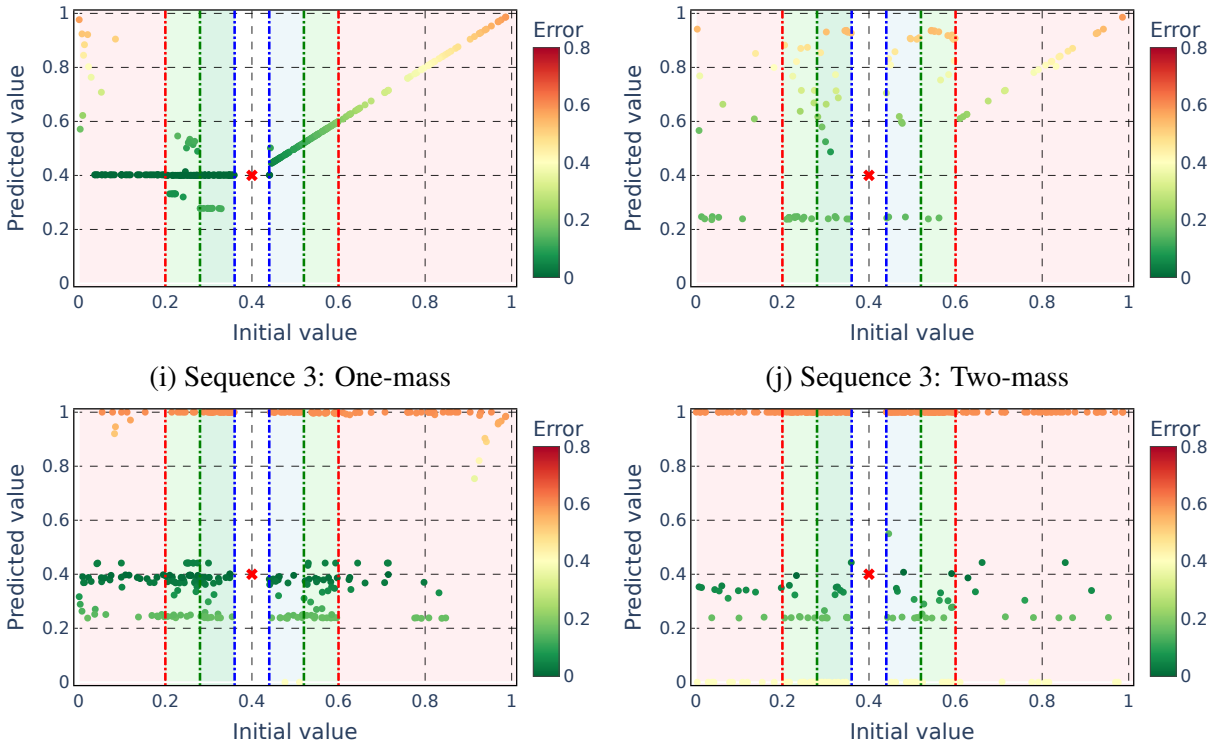
(h) Sequence 2: Four-mass

Figure 8.17

It is important to note that the shape of the PSKF's basin of attraction depends on several factors: the complexity of the physical model, the true friction coefficients of the test sequence, and the chosen process and measurement noise covariance matrices. In other words, modifying the update rule or the initialization of these matrices will alter the basin of attraction. For this reason, the results in the figure should be viewed as an overall illustration of the PSKF's behavior under varying initial values of the friction coefficients, rather than as fixed outcomes.

As shown in Fig. 8.18, for the one-mass model across the three test sequences, the estimation error decreases as the initialization approaches the true value, with the lowest errors observed in the left close region. For the two-, three-, and four-mass physical models, the behavior is more complex: in some cases, the observer performs better when the initial values lie in the left far region (i.e., much smaller than the true values) than when they are closer, in other cases, the estimates never converge to the true value or even diverge completely. Furthermore, the complexity of the physical model has a strong impact on estimation accuracy. As the model complexity increases, the MAE also increases, reflecting a narrower basin of attraction for more complex systems. This effect is particularly pronounced for the static friction coefficient, which is inherently more challenging to estimate than the dynamic one, since it is only observable during the jump from stick to slip phases. As a result, the MAE for the static friction coefficient is consistently higher than that for the dynamic coefficient.

In conclusion, the results demonstrate that the PSKF's basin of attraction becomes increasingly sensitive to the choice of initial values as the complexity of the embedded physical model increases. Moreover, higher model complexity introduces a larger number of parameters that must be carefully tuned, making parameter selection more challenging and further impacting



(i) Sequence 3: One-mass

(j) Sequence 3: Two-mass

(k) Sequence 3: Three-mass

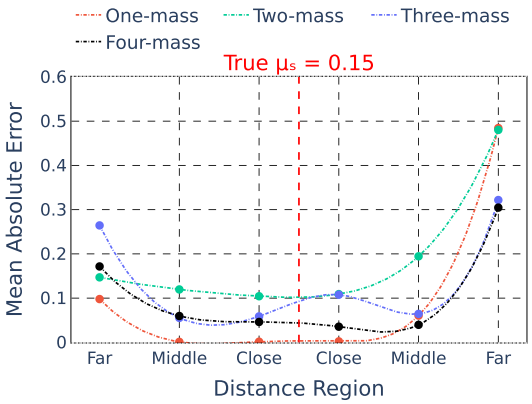
(l) Sequence 3: Four-mass

Figure 8.17: Effect of physical model complexity on the basin of attraction of the PSKF for estimating the static friction coefficient μ_s. The experiments consider three test sequences with true values of $\mu_s = 0.15$, $\mu_s = 0.3$, and $\mu_s = 0.4$, respectively. Each circle represents a distinct initialization run, while the red cross indicates the true value. The background shading denotes the initialization regions: blue for values close to the true μ_s, green for moderate deviations, and red for distant initializations. The estimation error is defined as Error $= |\mu_s - \hat{\mu}_s|$.

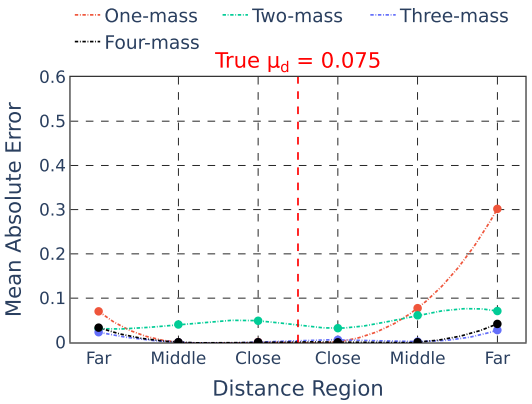
the observer's robustness and convergence. Since the primary objective is to design a PSKF observer suitable for real-world applications, where only low-frequency surface measurements (1 Hz) are available, this limitation must be taken into account when selecting the embedded physical model. Based on the presented results and the modeling rationale discussed in Section 6.4, only the one-mass and two-mass lumped torsional models allow the PSKF to both converge reliably and accurately estimate the friction coefficients and downhole rotational speed at a 1-Hz update rate.

8.8.2 Case 2: Performance of PSKF with lumped models on PDE-generated sequences

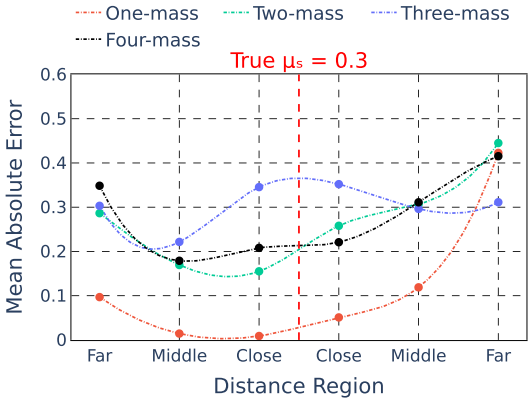
As discussed in the previous section, only PSKF with embedded one-mass (Eq. (6.7)) and two-mass (Eq. (6.8)) lumped torsional dynamic models are suitable for predicting friction coefficients and bit rotational speed when using surface measurements sampled at 1 Hz. However, in this previous analysis, the PSKF is tested on simulated data generated using the same physical model embedded in the observer. In reality, the true downhole dynamics are likely more complex than those captured by simplified one- or two-mass models.



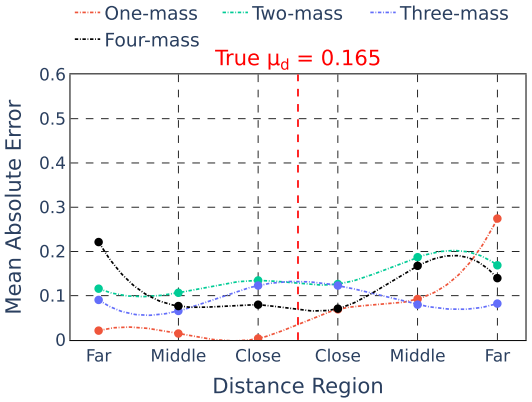
(a) Sequence 1: Static friction coefficient



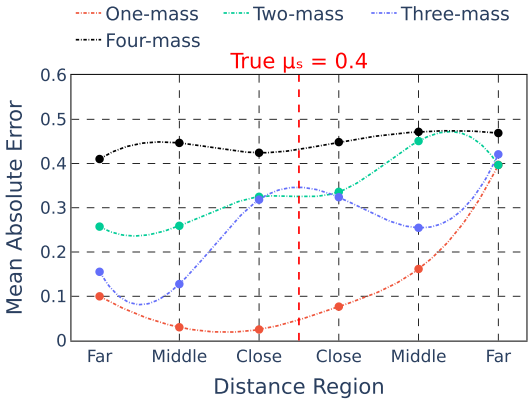
(b) Sequence 1: Dynamic friction coefficient



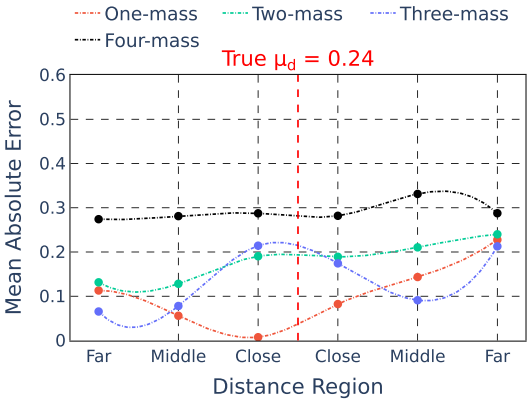
(c) Sequence 2: Static friction coefficient



(d) Sequence 2: Dynamic friction coefficient



(e) Sequence 3: Static friction coefficient



(f) Sequence 3: Dynamic friction coefficient

Figure 8.18: MAE of the predicted static (μ_s) and dynamic (μ_d) friction coefficients using the PSKF applied to three test sequences with ground truth values: $\{\mu_{s_1} = 0.15,\ \mu_{d_1} = 0.075\}$, $\{\mu_{s_2} = 0.3,\ \mu_{d_2} = 0.165\}$, and $\{\mu_{s_3} = 0.4,\ \mu_{d_3} = 0.24\}$. Each sequence is processed using four physical models: one-mass, two-mass, three-mass, and four-mass systems. For each sequence, 100 runs are conducted with randomly selected initializations drawn from three predefined regions: Far, middle, and close. The MAE values represent the average error over the 100 runs for each region.

Therefore, the goal of this section is to evaluate the performance of the PSKF when applied to simulated data generated from a more realistic and complex physical model, specifically, a distributed PDE-based model, while still using a one-mass or two-mass lumped model within the PSKF. This evaluation is conducted under the constraint of a low measurement update frequency of 1 Hz, to reflect real-world data availability.

The PDE-based model used to generate the data is described in [155]. The trajectory of the test well used to generate the data is shown in Fig. 8.19. The parameters of the one-mass and two-mass physical models used in within PSKF are presented in Table 8.8. For the one-mass model, the parameters represent the equivalent properties of the entire well. In contrast, the two-mass model divides the drill string into two parts: one mass representing the vertical section and the other representing the lateral section as explained in Section 6.4. A total of 64 sequences are simulated by randomly sampling static and dynamic friction coefficients from the ranges $\mu_s \in [0.15, 0.5]$ and $\mu_d \in [0.075, 0.425]$, respectively. The simulated data are then downsampled to 1 Hz to match the limited update frequency available in real drilling operations.

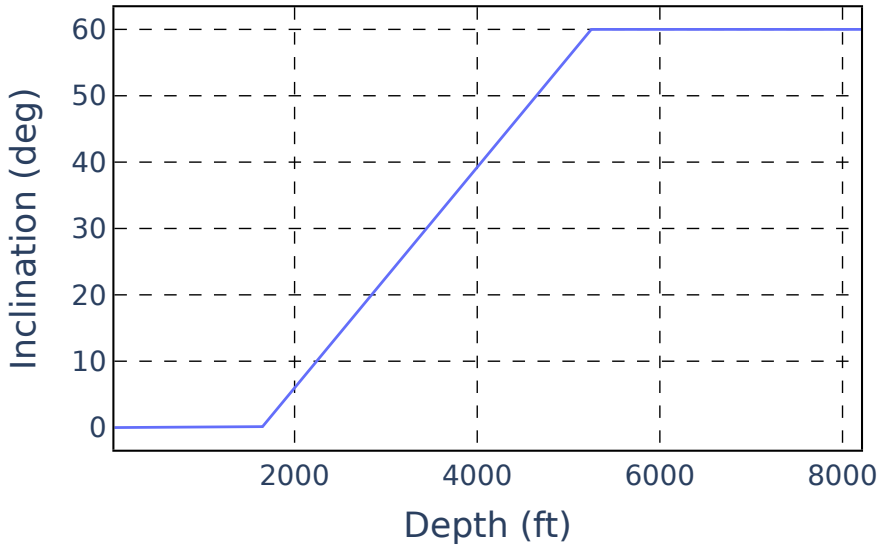


Figure 8.19: Trajectory of the test well used for generating sequences in the PDE-based model simulations.

Two observer configurations are tested: one using the PSKF with a one-mass lumped torsional dynamic model, and the other using a two-mass model. For each configuration, two initialization strategies for the friction coefficients are evaluated: (i) fixed initial values across all sequences, where $\mu_s^{Init} = 0.15$ and $\mu_d^{Init} = 0.1$; and (ii) scaled initialization, where the initial values vary depending on the true coefficients: $\mu_s^{Init} = 0.5\,\mu_s$ and $\mu_d^{Init} = 0.5\,\mu_d$, where μ_s and μ_d denote the ground truth values used to generate the data.

8.8.2.1 Evaluation of PSKF with one-mass model on PDE-based simulations

The performance of the PSKF with a one-mass embedded physical model and 1-Hz update frequency under the two conditions of initialization is summarized in Table 8.9, where both

	One-mass	**Two-mass**	
Inertia (kg.m²)	$I = 637.56$	$I_1 = 34.13$	$I_2 = 603.42$
Stiffness (N.m)	$K = 611.22$	$K_1 = 868.14$	$K_2 = 2065.31$
Normal contact force (N)	$F^N = 65205.34$	$F_1^N = 14844.79$	$F_2^N = 50360.54$
Length (ft)	$L = 8202$	$L_1 = 5249$	$L_2 = 2953$

Table 8.8: Physical parameters of the one-mass and two-mass lumped torsional dynamic models used for the test well. For the one-mass model, parameters represent the entire drill string. In the two-mass model, the drill string is divided into vertical (I_1, K_1, F_1^N, L_1) and lateral (I_2, K_2, F_2^N, L_2) sections.

the MAE and the MAPE for the estimated friction coefficients are reported. The results clearly indicate that the scaled initialization yields significantly lower estimation errors, particularly for μ_s, where MAE drops from 53.51×10^{-3} to 20.68×10^{-3} and MAPE from 15.95% to 5.85%. Similar improvements are observed for μ_d.

	Fixed intial values		**Scaled initial values**	
	MAE	**MAPE**	**MAE**	**MAPE**
μ_s	53.51×10^{-3}	15.95	20.68×10^{-3}	5.85
μ_d	26.94×10^{-3}	9.56	13.9×10^{-3}	6.99

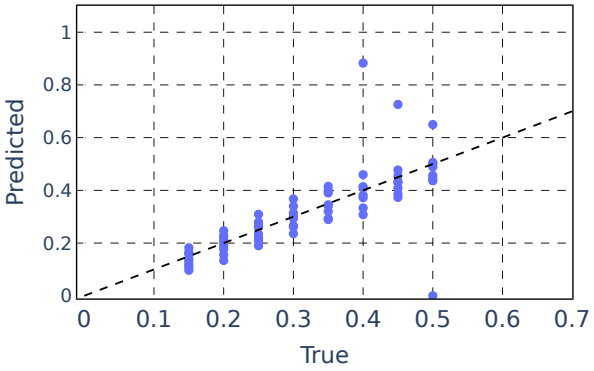
Table 8.9: Comparison of one-mass PSKF performance for friction coefficient estimation at 1-Hz update frequency on simulated data generated from a PDE-based model. Results are presented for two initialization strategies: fixed initial values ($\mu_s^{Init} = 0.15$, $\mu_d^{Init} = 0.1$), and scaled initial values ($\mu_s^{Init} = 0.5\mu_s$, $\mu_d^{Init} = 0.5\mu_d$), where μ_s and μ_d are the true values.

Fig. 8.20 further illustrate the estimation performance by showing the estimated and true values of μ_s and μ_d across all sequences under both initialization strategies. With fixed initialization, the observer sometimes fails to converge, especially when the true values are far from the initial ones, confirming the sensitivity of the PSKF to initial conditions. The same problem appears for some sequences with scaled initial values. However, the number of sequences with this kind of issue is much lower than that where the observer successfully converges to the true values under both initialization strategies.

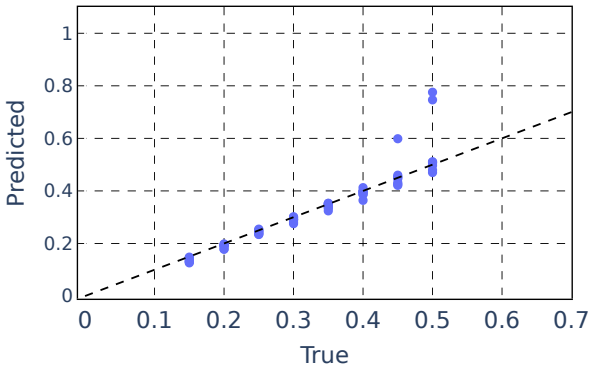
An example of the PSKF output on a single test sequence is presented in Fig. 8.21 with fixed friction coefficients initial values. Figs. 8.21a and 8.22 show the predicted and true bit rotation speed over time and their corresponding frequency spectra, respectively. Fig. 8.21b displays the estimated versus true static and dynamic friction coefficients. For this sequence the PSKF successfully converges to the true friction coefficients.

The bit rotation speed prediction closely follows the simulated downhole signal generated by the PDE-based model, effectively capturing the stick-slip behavior, with a stick-slip period of approximately 6 seconds.

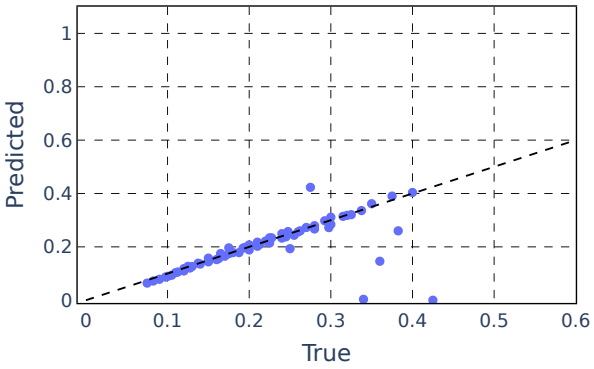
These results confirm that, even when using a simplified one-mass model, the PSKF is capable of capturing complex dynamics and accurately estimating friction coefficients with only a 1-Hz update frequency. This suggests that a PSKF based on a one-mass lumped torsional dynamic model can be effectively used to predict downhole rotational speed and friction coefficients from real data using only 1-Hz surface measurements.



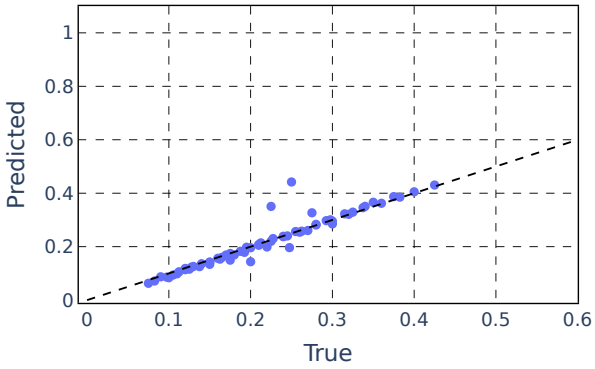
(a) μ_s with fixed initial value $\mu_s^{Init} = 0.15$



(b) μ_s with scaled initial value $\mu_s^{Init} = 0.5\,\mu_s$

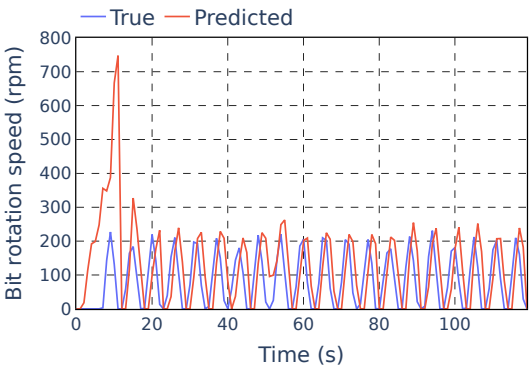


(c) μ_d with fixed initial value $\mu_d^{Init} = 0.1$

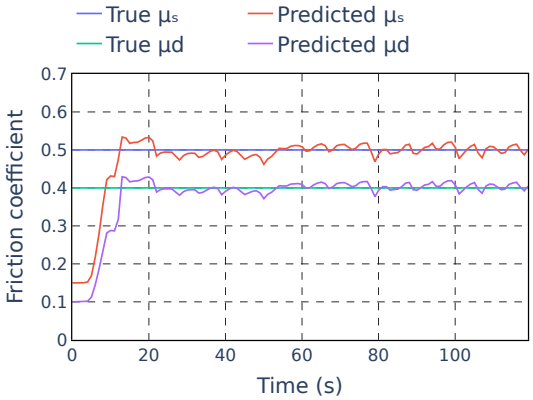


(d) μ_d with scaled initial value: $\mu_d^{Init} = 0.5\,\mu_d$

Figure 8.20: Comparison of estimated and true friction coefficients using PSKF with a one-mass lumped torsional dynamic embedded model and 1-Hz update frequency. The simulated data are generated using a PDE-based model. Results are shown for two different initialization strategies: fixed values vs. scaled true values.

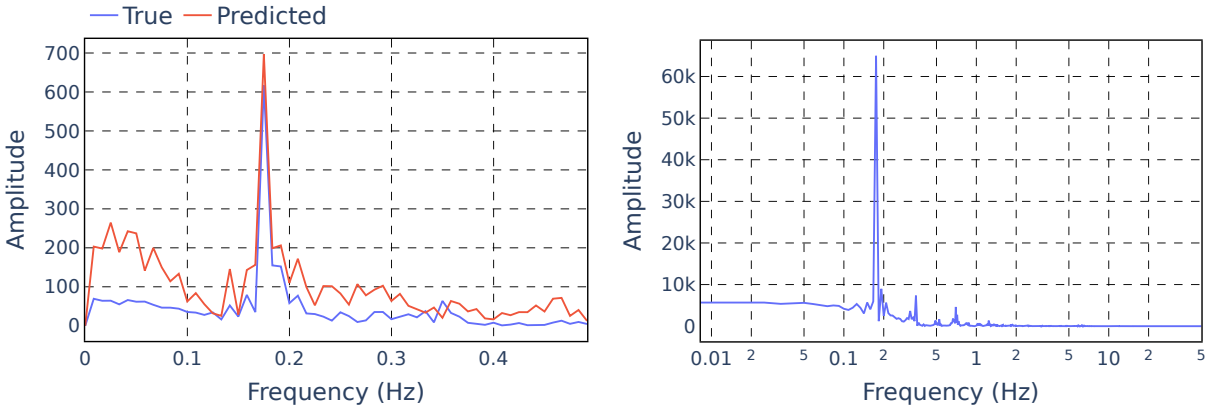


(a) Estimated and true bit rotation speed variations over time.



(b) Estimated and true static and dynamic friction coefficients.

Figure 8.21: Example of one-mass PSKF performance on a test sequence generated using a PDE-based model at 1-Hz update frequency: (a) shows the estimated and true bit rotation speed over time; and (b) presents the estimated and true static and dynamic friction coefficients.



(a) Frequency spectra of estimated and true bit rotation speed et 1 Hz.

(b) Frequency spectra of true bit rotation speed at 100 Hz.

Figure 8.22: Frequency spectra of (a) the true and predicted bit rotation speeds shown in Fig. 8.21 at 1 Hz, and (b) the true bit rotation speed at 100 Hz, plotted with a logarithmic x-axis.

8.8.2.2 Evaluation of PSKF with two-mass model on PDE-based simulations

This section evaluates the performance of the PSKF when a two-mass lumped torsional dynamic model is used as the embedded physical model, while the measurement data are generated using a PDE-based model that represents a more realistic and complex dynamical system.

Tests with a 1-Hz update frequency demonstrate that the PSKF with the two-mass model consistently fails to converge for all test sequences. Specifically, the estimated friction coefficients tend to saturate at the parameter boundaries (e.g., $\mu \to 0$ or $\mu \to 1$). This behavior highlights that the update frequency of 1 Hz does not provide sufficiently frequent measurement information to effectively track the states and parameters of the more complex system represented by the PDE-based model.

There are a few reasons for this. First, the two-mass model is more complex than the one-mass version, with more states to estimate. This means the observer needs more frequent measurements to keep uncertainty low and to correct its predictions before errors build up.

This highlights how crucial it is to align the observer's update frequency with the system's complexity and the speed of its dynamics. Just using a more detailed physical model inside the PSKF might not help if the data feeding it is not frequent enough. Without a high enough update rate, the observer can not always take advantage of the richer model to produce accurate estimates. As demonstrated using PDE-based simulated data, with a **1-Hz update frequency**, the PSKF with a **one-mass lumped torsional dynamic model** can reproduce the downhole velocity and predict friction coefficients of such complex dynamics. Therefore, this model is used in the next section on real field data.

8.9 PSKF application on field data

To further assess the performance of the one-mass PSKF, real drilling data from test well 7 are used (additional and longer sequences are presented in Chapter 9). The main characteristics of this well are summarized in Table 3.1, and its trajectory is illustrated in Fig. 3.7. The PSKF

takes as input surface measurements recorded at 1 Hz, specifically torque and rotational velocity. Similar to the PINN, it also incorporates the time evolution of the drill string's equivalent inertia, stiffness, and damping, computed using Eq. (6.1), as well as the time evolution of the equivalent normal contact force, estimated through Torque and Drag (T&D) modeling [155, 69, 176, 71]. Based on these inputs, it estimates the downhole rotational speed and friction coefficients in real-time.

Unlike the previous cases where the bit is assumed to be off-bottom, this test sequence corresponds to on-bottom drilling. Consequently, the TOB term in Eq. (6.7), denoted as $T^b(\omega_b + \omega_{\text{Motor}}, \tau_s, \tau_d)$, is no longer zero. TOB can either be estimated directly at the surface from the differential pressure, computed as the difference between off-bottom and on-bottom standpipe pressure readings, or indirectly, based on the estimated surface WOB (WOB_{Surface}). In the latter case, WOB_{Surface} is derived from the difference between off-bottom and on-bottom hookload measurements. The static torque is then modeled as $\tau_s = Agg \cdot WOB_{\text{Surface}}$, where Agg is the bit aggressiveness. The dynamic torque is defined as $\tau_d = \rho\tau_s$, with ρ representing the ratio between dynamic and static torque. In our case, ρ is fixed at 0.8. The length of the selected test sequence is approximately 190 seconds, recorded at a depth of 18 732 ft. This sequence is particularly interesting because it exhibits severe stick-slip oscillations with a period of approximately 10 s.

The results are shown in Fig. 8.23, which compares the measured bit rotation speed with the PSKF estimates for this test sequence. It displays the surface torque, surface rotational speed, downhole measured TOB, and downhole motor speed. In addition, the PSKF-predicted bit speed $\hat{\omega}_b$ is compared with actual bit speed measurements, which are represented by three curves: the maximum, minimum, and mean values, derived from high-frequency downhole sensor data sampled at 80 Hz over a 2.56-second interval. These measurements are obtained from downhole sensors after drilling is completed.

As seen in Fig. 8.23, the PSKF accurately reproduces the bit rotation speed measurements, maximum, minimum, and mean, using only surface data. A small time shift is observed between the predicted $\hat{\omega}_b$ and the actual bit speed. This discrepancy can be attributed to the propagation time required for downhole vibrations to reach the surface, and to environmental factors [136], which may cause clock drift at the bottom. Since this delay is time-varying, aligning long sequences of surface and downhole data is inherently difficult. Fortunately, this test sequence is short and is manually synchronized. Under these conditions, the PSKF achieved a MAE of 20.44 rpm and a MAPE of 13.5% between the predicted and actual bit velocities.

For the predicted friction coefficients, as mentioned in Chapter 7, the uncertainty associated with the true friction coefficients makes it challenging to analyze their absolute values. Therefore, this analysis is deferred to future work.

8.10 Conclusion

This chapter presents the PSKF, an extension of the standard SKF, ensuring state estimates remain consistent with predefined invariant sets. By adding a projection step during the correction phase, the PSKF ensures that state transitions between domains occur only by crossing specified guard conditions. Simulations using one-mass and two-mass lumped torsional drill string models with varying friction coefficients show that the PSKF significantly outperforms the SKF,

[3] Bit aggressiveness = $\frac{TOB}{WOB}$

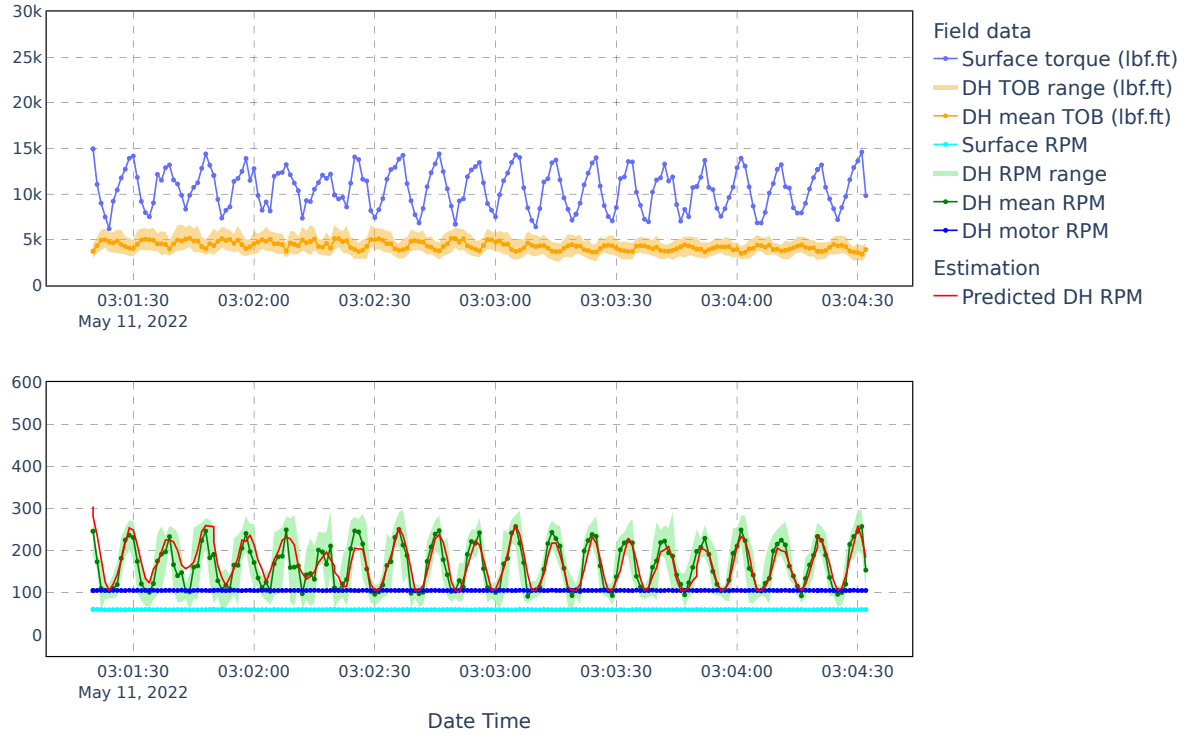


Figure 8.23: Comparison between actual bit rotation speed measurements and PSKF predictions using 1-Hz surface torque and surface rotational speed as inputs, for a test sequence from Well 7 (see Table 3.1). 'DH TOB range' refers to the max–min range of downhole measured TOB, and 'DH mean TOB' to its mean value. Similarly, 'DH RPM range' and 'DH mean RPM' represent the range and mean of the measured downhole bit rotation speed. 'DH motor RPM' corresponds to the downhole measured motor rotational speed.

reducing estimation errors by over 80 %. This makes the PSKF a more reliable filtering method for hybrid systems, preserving consistency with domain constraints.

Next, the PSKF is tested on multiple lumped torsional dynamic models to explore how its performance relates to update frequency and the complexity of the embedded physical model. These tests include simulations generated either from the same physical model embedded in the filter or from more detailed PDE-based torsional models, where the PSKF uses a simplified lumped model.

The objective is to identify the most suitable model for predicting friction coefficients and bit rotation speed from field data using 1-Hz surface measurements. The results demonstrate that the PSKF with a one-mass model accurately reconstructs the downhole rotation speed from PDE-based simulated data at a 1 Hz update frequency. Consequently, this model is selected for application to real field data. Results from a short real-data sequence are presented here, while additional sequences and a more detailed analysis of field data tests are provided in Chapter 9.

Chapter 9

PINN and PSKF comparison

Résumé du chapitre

Ce chapitre présente une étude comparative des deux méthodes introduites dans les chapitres précédents, à savoir le PINN et le PSKF, toutes deux développées pour estimer la vitesse de rotation du trépan et les coefficients de frottement à partir de mesures de surface à basse fréquence. La comparaison est effectuée à l'aide de données simulées générées par des EDP ainsi que de données réelles provenant de différents puits horizontaux. Pour les données simulées, l'évaluation inclut une troisième méthode de référence : `nlgreyest`, un outil de modélisation boîte grise non linéaire issu du Matlab System Identification Toolbox [263]. Pour les données réelles, la vitesse de rotation du trépan prédite par les deux méthodes, PINN et PSKF, est comparée aux mesures réelles du fond du puits.

Chapter Summary

This chapter presents a comparative study of the two methods introduced in the previous chapters, PINN and PSKF, both developed to estimate bit rotation speed and friction coefficients using low-frequency surface measurements. The comparison is carried out using both PDE-generated simulated data and field data from various lateral wells. For the simulated data, the evaluation includes a third benchmark method: `nlgreyest`, a nonlinear grey-box modeling tool from the Matlab System Identification Toolbox [263]. For field data, the predicted bit rotation speed, for both methods, PINN and PSKF, is compared to the true downhole measurements.

9.1 Objective

This chapter aims to compare two estimators of downhole conditions, the PSKF and the PINN. Both methods rely solely on low-frequency surface measurements sampled at 1 Hz, without

access to real-time or historical downhole data.

As detailed in Chapters 8 and 9, PSKF and PINN are built upon the same underlying physical model: a one-mass nonlinear lumped torsional dynamic system, presented in Eq. (6.7). The rationale for this modeling choice is discussed in Sections 6.4, 8.8, and 7.4.1. Despite the simplicity of the embedded model, both PSKF and PINN demonstrate strong performance in recovering downhole dynamics and estimating friction coefficients from synthetic data generated by a high-fidelity PDE-based simulator that emulates real-world drilling complexity.

This chapter is structured in two parts:

- **Part 1: Comparison using PDE-simulated data**
  Both PSKF and PINN are evaluated using synthetic data generated by a PDE-based simulator. Their performance is compared to a third method: *nonlinear grey-box modeling* using `nlgreyest` from MATLAB's System Identification Toolbox. Recognized as a state-of-the-art technique [263], `nlgreyest` estimates model parameters by fitting a nonlinear dynamic model to observed data using physical priors and optimization. The evaluation sequences are identical to those in Sections 8.8 and 7.6.1. Since the ground-truth friction coefficients are known, this comparison enables direct and quantitative performance assessment.

- **Part 2: Application on field data**
  Both methods are tested on real drilling data acquired from four different wells, each corresponding to distinct rigs and geographical regions. Even within the same well, the data span different time intervals, capturing variations in lithology and drilling conditions, thus ensuring a robust and realistic performance comparison. In this setting, the predicted bit rotational speed is validated against actual measurements. The ground-truth friction coefficients, however, remain unknown and are not considered in this dissertation, their analysis is deferred to future work, as it requires additional data and a more detailed study of their variations.

The three methods, PINN, PSKF, and `nlgreyest`, are all designed to solve the optimization problem described in Eq. (7.1), but each adopts a different strategy. The PINN, an offline method, employs a neural network to approximate the solution while incorporating a smooth friction model, which avoids discontinuities, as described in Section 7.4.1. In contrast, the PSKF, an online method, is an iterative method that explicitly accounts for stick-slip behavior using a stiction friction model, thus handling discontinuities in the system dynamics. Finally, `nlgreyest`, another offline method, discretizes the governing Ordinary Differential Equations (ODEs) and performs classical optimization using a smooth friction model similar to that of the PINN. All these methods require a prior characterizing the mass, damping, and stiffness of the system in addition to an estimation of the contact forces along the drill string.

9.2 Comparison on simulated data

As discussed in Section 7.6.1 for PINN and Section 8.8.2 for PSKF, both models are evaluated on simulated data generated by a more realistic and complex physical model based on a PDE. Despite the increased complexity of the simulation environment, both PINN and PSKF, while internally relying on a simplified one-mass nonlinear lumped torsional dynamic model, succeed in predicting the simulated bit rotation speed and friction coefficients. The goal of this section is

to compare the performance of both methods on the same PDE-simulated data and to benchmark them against `nlgreyest`, a state-of-the-art nonlinear grey-box estimation method.

9.2.1 Parameters identification with `nlgreyest`

The `nlgreyest` function estimates the parameters of nonlinear grey-box models by combining known physical laws with measured data, in a way that is conceptually similar to PINN and PSKF. This method is widely recognized as a state-of-the-art approach in system identification, as demonstrated in prior works such as [264, 263, 265].

Parameter estimation with `nlgreyest` involves fitting a predefined physical model to observed data by minimizing the error between the model's simulated outputs and the actual measurements. The optimization is typically carried out using iterative algorithms like Levenberg-Marquardt or Gauss-Newton, which require the computation of output gradients with respect to the model parameters.

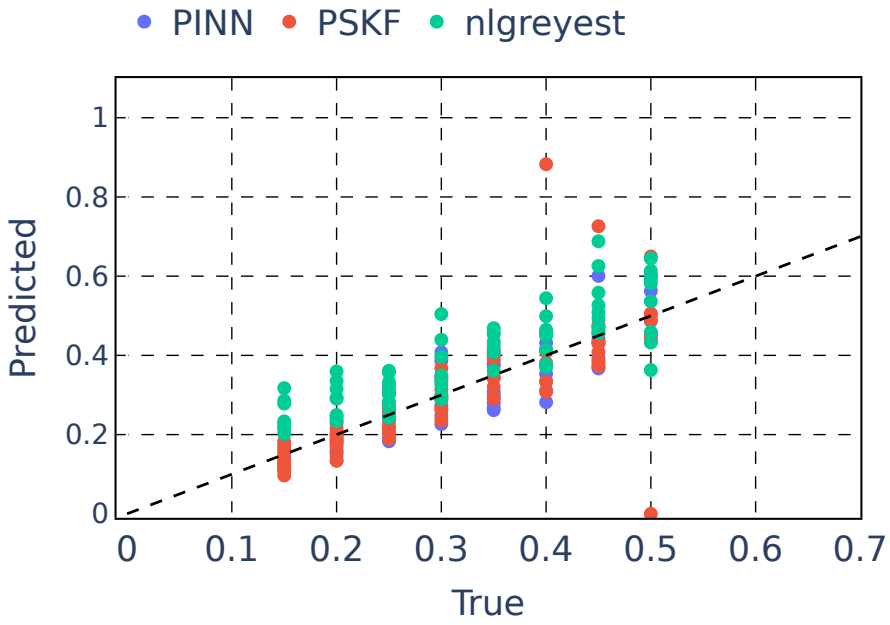
As previously mentioned, training the PINN model for each new sequence requires approximately 10 minutes. In contrast, the `nlgreyest` function converges much faster, typically in less than one minute. To demonstrate its performance, we perform ten identification runs using `nlgreyest`, each initialized with random friction coefficient values drawn from a uniform distribution $\mathscr{U}(0,1)$. The final predicted friction coefficients are obtained by averaging the results across the ten runs.

9.2.2 Comparison results

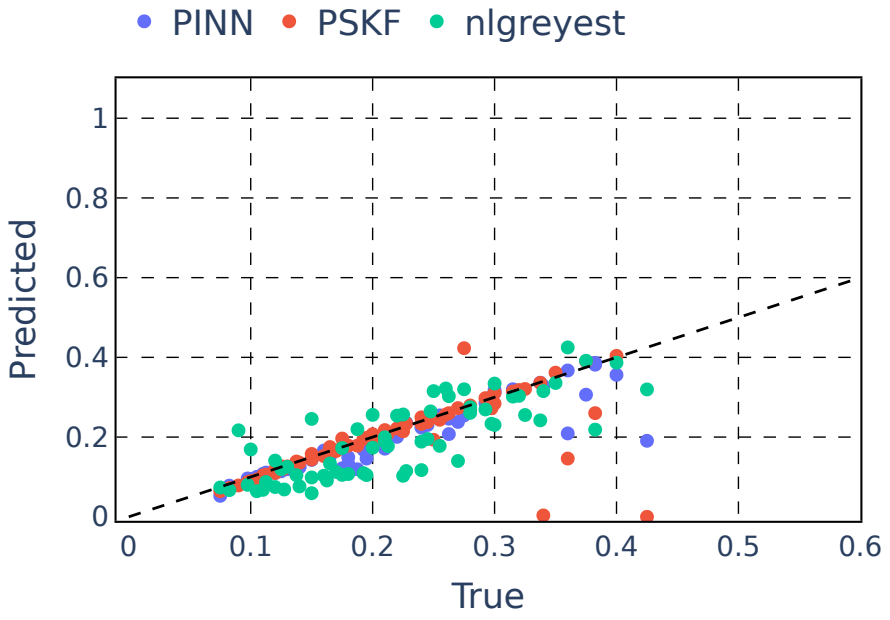
The PDE-based simulator used to generate the data is described in [155]. A total of 64 sequences are employed in this evaluation, with static friction coefficients sampled in the range $[0.15, 0.5]$ and dynamic friction coefficients in the range $[0.075, 0.425]$. The well trajectory used for these simulations is shown in Fig. 8.19, with the corresponding equivalent parameters listed in Table 8.8. The results are shown in Fig. 9.1, which compares the predicted friction coefficients from all three tested methods. Fig. 9.2 presents the calculated MAE and MAPE of the three tested methods over all test sequences, for the static and dynamic friction coefficients.

To compare the three methods, several aspects are considered, as illustrated in Table 9.1, starting with the prediction error. As shown in the results, all three methods exhibit higher errors when predicting the static friction coefficient μ_s compared to the dynamic coefficient μ_d. This is expected, as the static coefficient is only active during the jump from stick phase to slip phase, which is more difficult to observe and estimate. It is worth noting that all 64 test sequences used are characterized by severe stick-slip behavior. Among the three methods, the PINN achieves the lowest prediction error for both μ_s and μ_d. The PSKF ranks second, with errors higher than those of the PINN by approximately 11.58% for μ_s and 18.95% for μ_d. The `nlgreyest` method performs worst, with errors exceeding those of the PINN by about 40.35% for μ_s and 57.75% for μ_d.

In terms of speed, both the PINN and `nlgreyest` are offline methods, while the PSKF operates online. The convergence time of the PSKF depends on the sequence but typically takes around 20 seconds, as shown in Fig. 8.21b. The PINN requires about 10 minutes to train on a 30-second sequence sampled at 1 Hz. The `nlgreyest` is faster, typically converging in under one minute. However, to ensure a fair comparison, ten identification runs are performed with different random initializations.



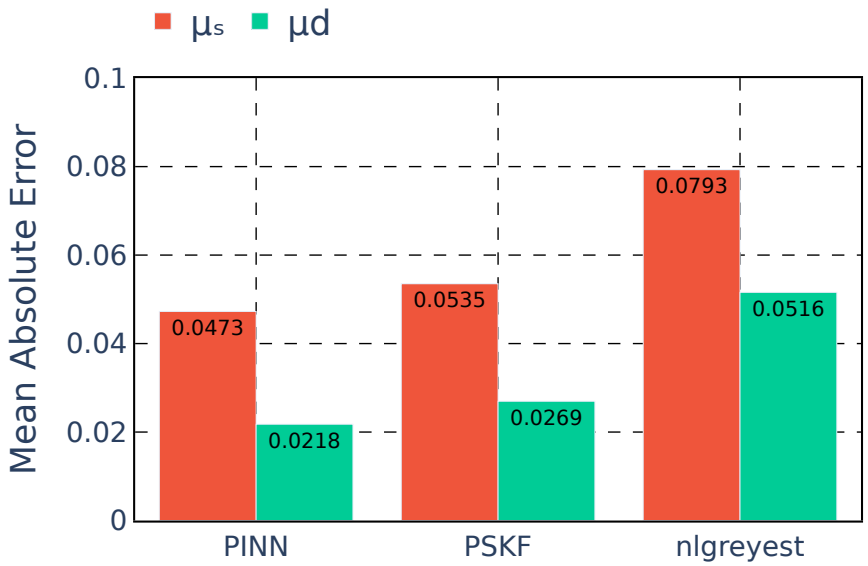
(a) Static friction coefficient



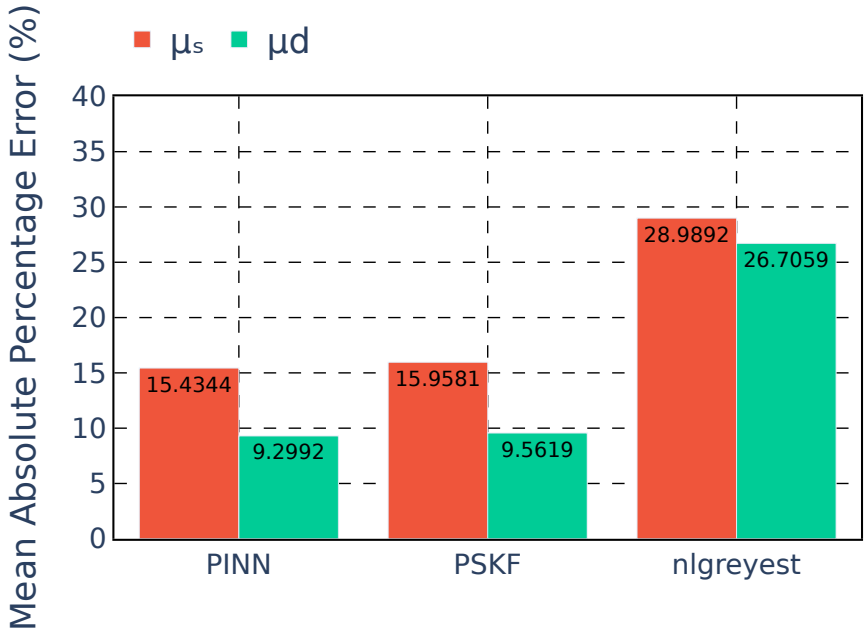
(b) Dynamic friction coefficient

Figure 9.1: Predicted static and dynamic friction coefficients for the test PDE-generated sequences, obtained using the three methods: PINN, PSKF, and *nlgreyest*.

To evaluate robustness to initialization, in both the PINN and the `nlgreyest` methods, the initial values of the friction coefficients are chosen randomly. In contrast, for the PSKF, the initial friction coefficients are fixed at $\mu_s = 0.15$ and $\mu_d = 0.1$. As discussed in Paragraph 8.8.1.2,



(a) Mean Absolute Error



(b) Mean Absolute Percentage Error

Figure 9.2: MAE and MAPE of the three methods, PINN, PSKF, and *nlgreyest*, for the predicted static (μ_s) and dynamic (μ_d) friction coefficients.

keeping the initial values low allows the observer to escape the stick phase, thereby improving estimation accuracy. However, the PINN is independent of the initial values, that is, when the PINN is run multiple times on the same sequence with different initial values, it consistently

converges to the same predicted values. This is not the case for the PSKF (as discussed in Paragraph 8.8.1.2) and `nlgreyest` methods, which require careful tuning of initial parameters.

From an online/offline perspective, the PSKF is the only online method, making it suitable for real-time applications. The PINN and `nlgreyest` are offline methods, better suited for batch processing. In terms of deployment ease, the PSKF stands out due to its low computational cost and online capability, making it ideal for embedded or real-time systems. The PINN, while being more robust and slightly more accurate, requires significant computational resources and time for training, which can be prohibitive unless executed offline. The `nlgreyest` method can be deployed offline but is less reliable without expert parameter tuning. Finally, regarding the need for parameter tuning: the PINN requires tuning of its architecture (e.g., number of layers and neurons) and particularly the loss weighting coefficients used to balance the data and physics-based terms. For the PSKF, several parameters need to be configured, including the initial state vector, process noise covariance, and measurement noise covariance matrices. The `nlgreyest` method depends also on the quality of initial values.

Criterion	**PINN**	**PSKF**	**`nlgreyest`**
Accuracy (MAE/MAPE)	★★★(Best)	★★☆	★☆☆(Worst)
Prediction/Convergence time	★☆☆(Slow, ∼10 min)	★★★(∼20 s)	★★☆(<1 min)
Sensitivity to Initialization	Not Sensitive	Sensitive to initial values	Sensitive to initial values
Need for Parameter Tuning	Architecture tuning	Model and initial values tuning	Initial values
Initial Values	Randomized	Fixed ($\mu_s = 0.15$, $\mu_d = 0.1$)	Randomized
Execution type	Offline	Online	Offline

Table 9.1: Benchmark of PINN, PSKF, and nlgreyest approaches based on accuracy, computational efficiency, and model sensitivity.

9.3 Comparison on field data

In this section, the capabilities of both methods, PINN and PSKF, are further evaluated using field drilling data from multiple wells, followed by a direct comparison between the two methods. For both methods, the same one-mass lumped nonlinear torsional dynamic model described in Eq. (6.7), is employed. In the PSKF, a stiction friction model is incorporated (see Section 6.3.2.2), while a smooth friction model is adopted in the PINN (see Section 6.3.2.1). The choice of the smooth friction model for the PINN is justified in Section 7.4.1.

For the comparison with field data, the variation of downhole bit rotation speed, measured directly by downhole sensors, is used to assess the prediction accuracy of both methods. These measurements are collected after drilling operations are completed. For the friction coefficients, a major challenge is encountered due to the absence of ground-truth values, in contrast to the simulated data case, where the true values are known. As a result, the analysis of the estimated static and dynamic friction coefficients is more challenging and has been deferred to future work. Data from several wells are used for this comparison, and the corresponding well characteristics are described in the following section.

9.3.1 Field data description

Four different lateral wells are used for this comparison (wells 1, 5, 6, and 7), whose main characteristics are summarized in Table 3.1 and trajectories illustrated in Fig. 3.7. For each well,

surface and downhole measurements are collected at 1 Hz. The depth ranges of the available data used for this comparison for each test well are shown in Fig. 9.3. The surface data serve as input to both methods (PINN and PSKF) for predicting the downhole bit rotation speed and friction coefficients, whereas the downhole measurements are used only to validate the predicted bit rotation speed against the actual measurements. The preprocessing of the 1-Hz signals follows the same procedure described in Section 3.4.1. However, no information is available regarding the preprocessing steps applied to the original high-frequency raw data. Then, the surface and downhole data are synchronized, and an operation detection step is performed to isolate the rotary drilling phases[1]. For the PINN, the synchronized data are divided into 30-second segments, with each segment processed independently. In contrast, the PSKF is applied directly to the full sequences without segmentation. The choice of a 30-second window is user-defined and can be adjusted as needed.

Once cleaned, surface torque and surface rotational speed are used as inputs for both methods. Additionally, the surface-estimated downhole motor rotation speed is included, calculated as

$$\omega_{\text{Motor}} = Cy \cdot FlowRate$$

where Cy is a motor-specific constant that defines the number of revolutions per unit of flow rate, and the flow rate is measured at the surface. Because all sequences correspond to on-bottom drilling, the TOB term in Eq. (6.7) is nonzero and is either directly estimated at the surface or derived from the surface-estimated WOB, as described in Section 8.9. In addition to these surface inputs, drill string characteristics, including equivalent inertia, stiffness, damping are computed at each depth according to Eq. (6.1), and the normal contact force is determined via the Torque and Drag (T&D) model.

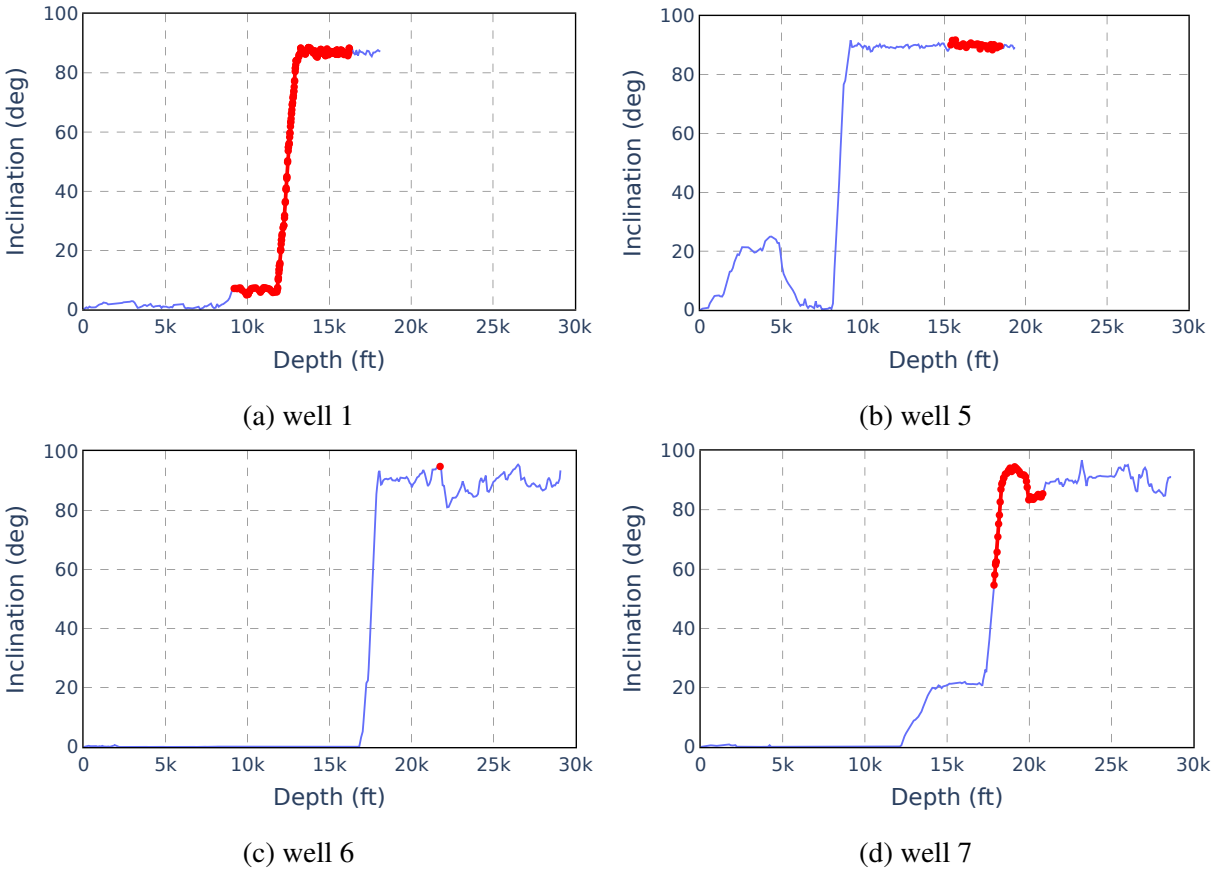
9.3.2 Results and discussions

9.3.2.1 Application on a selection of short sequences of field data

As discussed earlier, field downhole measurements are used to validate the bit rotation speed predicted by both methods. For the PINN, Figs. 9.4 and 9.5 show the comparison results for two short sequences from wells 1 and 7, respectively, both exhibiting severe stick-slip. Another short sequence from well 1 is presented in Fig. 7.8. The surface torque, surface rotational speed, and downhole measurements of TOB variations (when available) are shown alongside the actual bit rotation speed measurements, which are compared with the predicted values $\hat{\omega}_b$ by PINN. The measured bit rotation speed is represented by three curves: maximum, minimum, and mean, derived from high-frequency downhole sensor data sampled at 80 Hz over a 2.56-second interval, as explained earlier.

As illustrated in Fig. 9.4, for test well 1, the PINN accurately reproduces the bit rotation speed measurements, including maximum, minimum, and mean values, using only surface measurements. For the test well 7 (Fig. 9.5), unlike test well 1, the bit velocity predicted by the PINN aligns with the maximum and mean values of real bit rotation speed, but deviates from the minimum, which is approximately 100 rpm lower than the downhole motor's speed. This indicates that high-frequency bit rotation speed sensor detects instances where the bit velocity drops to zero even though the downhole motor continues rotating. This phenomenon,

[1] Although both methods can also be applied to off-bottom data, only on-bottom sequences are considered here due to the limited availability of off-bottom records.



(a) well 1

(b) well 5

(c) well 6

(d) well 7

Figure 9.3: Inclination of test wells 1, 5, 6, and 7 with depth. Red dots indicate the available data for each well, which are used to test both methods, PINN and PSKF.

known as bit-sticking, can lead to either motor back-drive events, which occurs when the motor is powerful enough to drive the drill string backward, or motor microstalls, which are high frequency events compared to motor stall, and are not captured by surface data sampled at less than 10 Hz [40]. Consequently, anomalies caused by motor microstalls are not observable from surface flow rate or differential pressure data sampled at 1 Hz (Fig. 9.5). Since the PINN relies on 1-Hz surface data, it is unable to detect motor microstalls. Another reason for the PINN's inability to reproduce bit-sticking lies in the assumptions of our physical model. In our model, the downhole motor rotation speed is assumed constant, whereas in reality, microstalls cause fluctuations in motor speed. Furthermore, bit-sticking itself is a high-frequency vibration phenomenon [40], while our simplified physical model is capable of reproducing only low-frequency dynamics. Taken together, these limitations explain why the PINN fails to capture the bit-sticking phenomenon.

The same sequences from test wells 1 and 7 are used to evaluate the PSKF. Figs. 9.6a and 9.6b show the predicted bit rotation speed by the PSKF for these sequences, corresponding to the same cases previously presented for PINN. For the sequence of well 7, the same remark applies as for the PINN: the PSKF reproduces the mean and maximum downhole rotation speeds, but not the actual min due to the bit-sticking phenomenon.

To provide a clearer comparison between the two methods, Tables 9.2 and 9.3 summarize the performance of PINN and PSKF on the short test sequences from well 1 and well 7, respectively. Three metrics are reported: the Mean Absolute Error (MAE, Eq. 4.9), the Pear-



Figure 9.4: Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with the PINN, using real surface torque and surface rotational speed as inputs for test well 1. The measured bit rotation speed is recorded by downhole sensors and received after drilling. Definitions of the terms shown: Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by PINN.

son correlation coefficient (r, Eq. 3.4), and the normalized DTW distance (scaled by sequence length, see Section 4.8.2). These metrics are computed between the predicted bit rotation speed and the downhole measured mean, maximum, and minimum rotation speeds. The use of DTW is particularly important in this context. Although surface and downhole measurements are synchronized in principle, small time shifts remain and vary over time. This explains the slight misalignment observed between predicted and measured signals (see Fig. 9.6b). Dynamic Time Warping (DTW) (see section 4.8.2) provides a way to account for this variability by comparing the overall similarity of the curves rather than point-by-point alignment.

For well 1 (Table 9.2), the PINN approach achieves slightly lower MAE on the downhole mean rotation speed (65.06 rpm vs. 66.11 rpm for PSKF), along with a higher correlation coefficient ($r = 0.52$ vs. 0.35). However, the DTW value is higher for PINN (3.37 vs. 2.75). For the maximum rotation speed, PINN shows a clear reduction in MAE (73.30 rpm vs. 90.80 rpm) and DTW (2.67 vs. 3.78), with slightly higher correlation ($r = 0.53$ vs. 0.33). This indicates that PINN tracks the peaks more effectively. On the other hand, for the minimum rotation speed, PSKF performs better: it achieves a lower MAE (68.49 rpm vs. 82.81 rpm) and DTW (3.91 vs. 5.50), while the correlation values remain relatively low for both methods. Overall, for

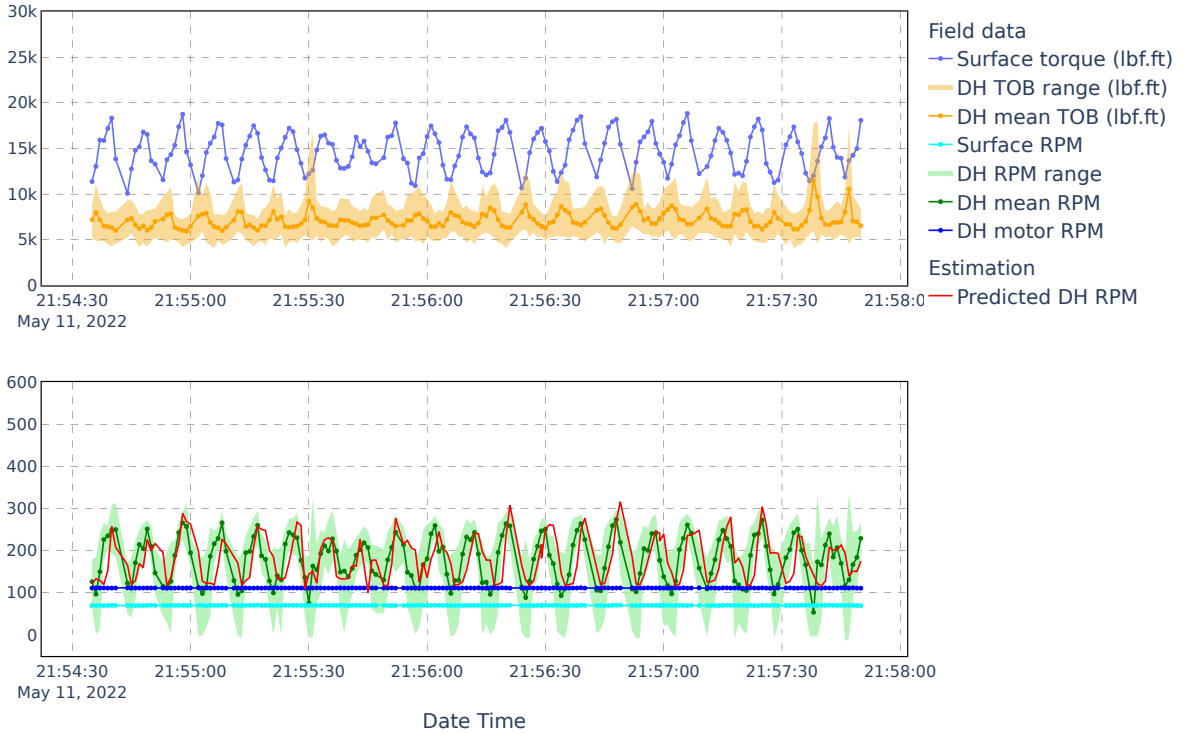
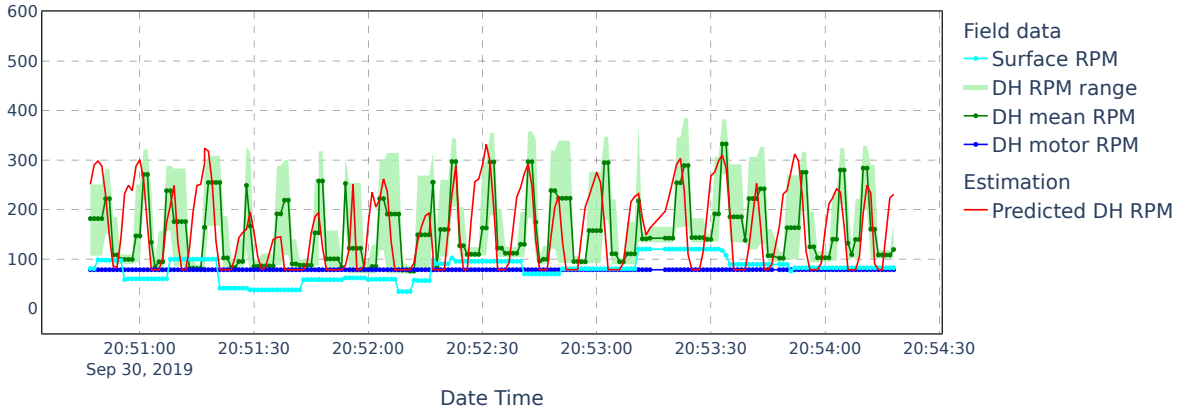


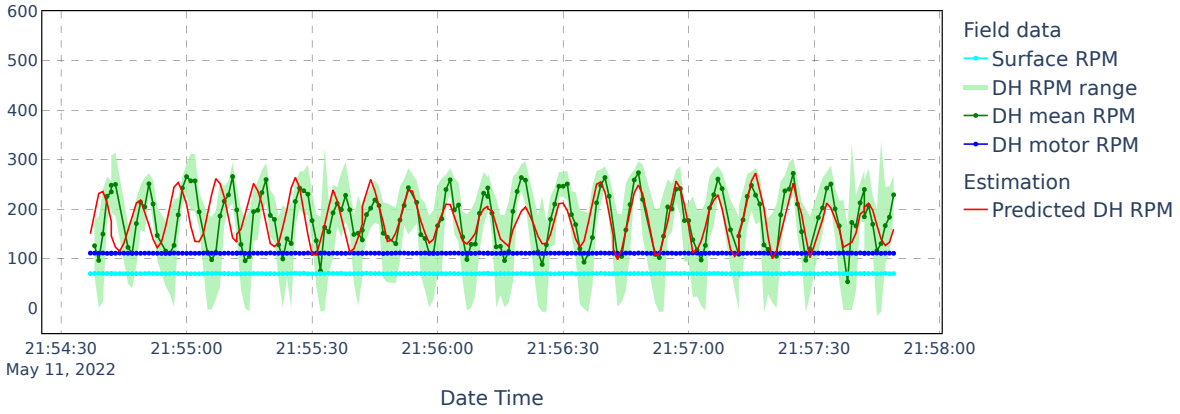
Figure 9.5: Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with the PINN, using real surface torque and surface rotational speed as inputs for test well 7. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by the PINN.

well 1, PINN is better suited for predicting the mean and maximum downhole speeds, whereas PSKF retains an advantage at the minimum.

For well 7 (Table 9.3), the differences between PINN and PSKF are marginal across all metrics. For the mean rotation speed, PINN slightly outperforms in DTW (1.78 vs. 1.83) but underperforms in MAE (43.88 rpm vs. 37.49 rpm). Both methods yield nearly identical correlation coefficients ($r \approx 0.49$–0.50). For the maximum rotation speed, the two approaches are almost indistinguishable. At the minimum rotation speed, PINN yields a slightly lower MAE (115.78 rpm vs. 120.35 rpm) and DTW (62.48 vs. 64.35), with a marginally higher correlation ($r = 0.51$ vs. 0.48). An additional observation for well 7 is that, for both methods, the largest errors (MAE and DTW) occur at the minimum rotation speed. This behavior is consistent with the bit-sticking phenomenon, which makes the reproduction of minimum speeds particularly challenging. In summary, PINN provides a modest advantage in well 1, especially for mean and maximum speeds, while in well 7 the two methods perform almost identically, with both struggling in the prediction of the minimum variations.



(a) well 1



(b) well 7

Figure 9.6: Comparison between the measured bit rotation speed and the predicted values $\hat{\omega}_b$ obtained with PSKF, using real surface torque and surface rotational speed as inputs for: (a) test well 1 and (b) test well 7. Definitions of the terms shown: Surface RPM – surface rotational speed; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH RPM – predicted downhole bit rotation speed by PINN.

Method	**DH mean RPM**			**DH max RPM**			**DH min RPM**		
	MAE (rpm)	**DTW (rpm)**	**r**	**MAE (rpm)**	**DTW (rpm)**	**r**	**MAE (rpm)**	**DTW (rpm)**	**r**
PINN	65.06	3.37	0.52	73.30	2.67	0.53	82.81	5.50	0.39
PSKF	66.11	2.75	0.35	90.80	3.78	0.33	68.49	3.91	0.32
Improvement (%)	1.6	-22.5	-48.6	19.3	29.3	-60.6	-20.9	-29.4	-21.9

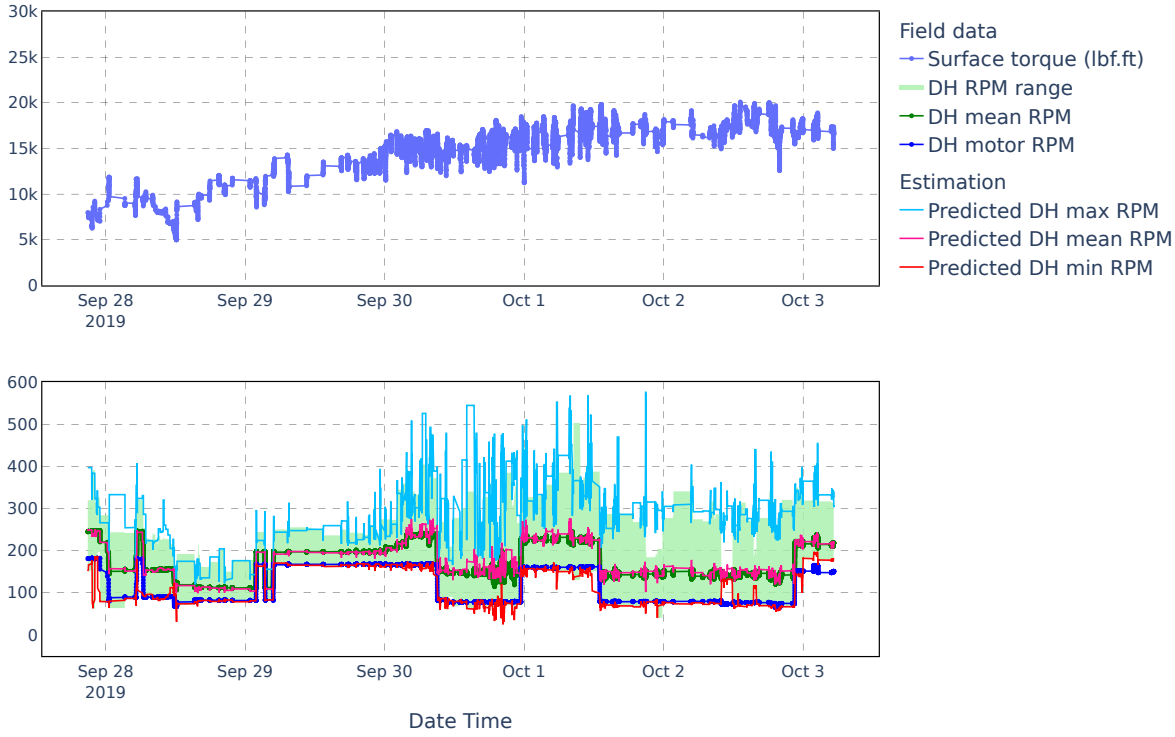
Table 9.2: MAE, normalized DTW, and Pearson correlation coefficient (r) between the mean, maximum, and minimum field downhole measured rotation speeds and the predicted bit rotation speed by both methods, PINN and PSKF, for the short test sequence of well 1. Improvement is calculated using Eq. (9.1).

Method	DH mean RPM			DH max RPM			DH min RPM		
	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r
PINN	43.88	1.78	0.49	64.55	3.50	0.40	115.78	62.48	0.51
PSKF	37.49	1.83	0.50	64.45	3.48	0.40	120.35	64.35	0.48
Improvement (%)	-17.0	2.7	2.0	-0.1	-0.6	0.0	3.79	2.9	-6.2

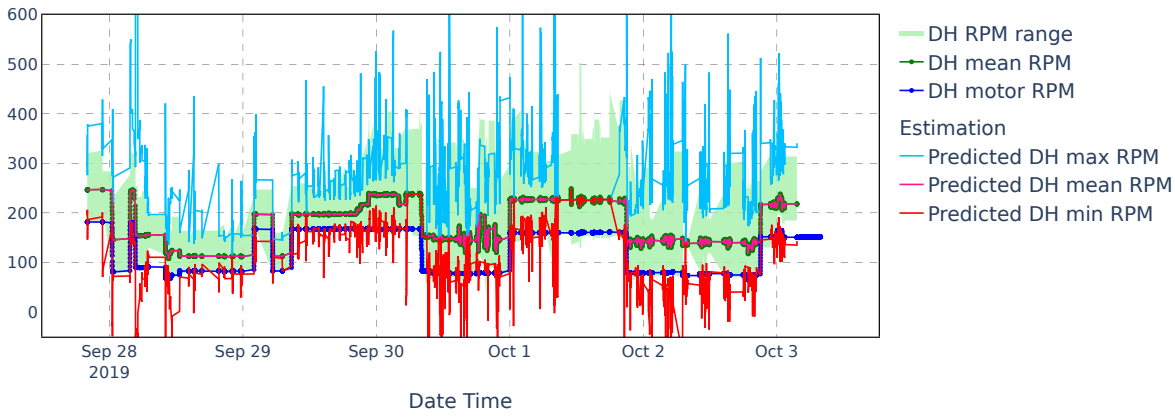
Table 9.3: MAE, normalized DTW, and Pearson correlation coefficient (r) between the mean, maximum, and minimum field downhole measured rotation speeds and the predicted bit rotation speed by both methods, PINN and PSKF, for the short test sequence of well 7.

9.3.2.2 Application on long sequences of field data

To demonstrate the performance of the PINN and PSKF on extended field sequences, Fig. 9.7a, Fig. 9.7c, Fig. 9.7e, and Fig. 9.7g illustrate the results for the PINN, while Fig. 9.7b, Fig. 9.7d, Fig. 9.7f, and Fig. 9.7h show those for the PSKF. Each figure displays the moving maximum, minimum, and average of both the measured and predicted downhole bit rotation speeds, along with the moving average of the downhole motor rotation speed, and surface torque for the four test wells 1, 7, 6, and 5, respectively.

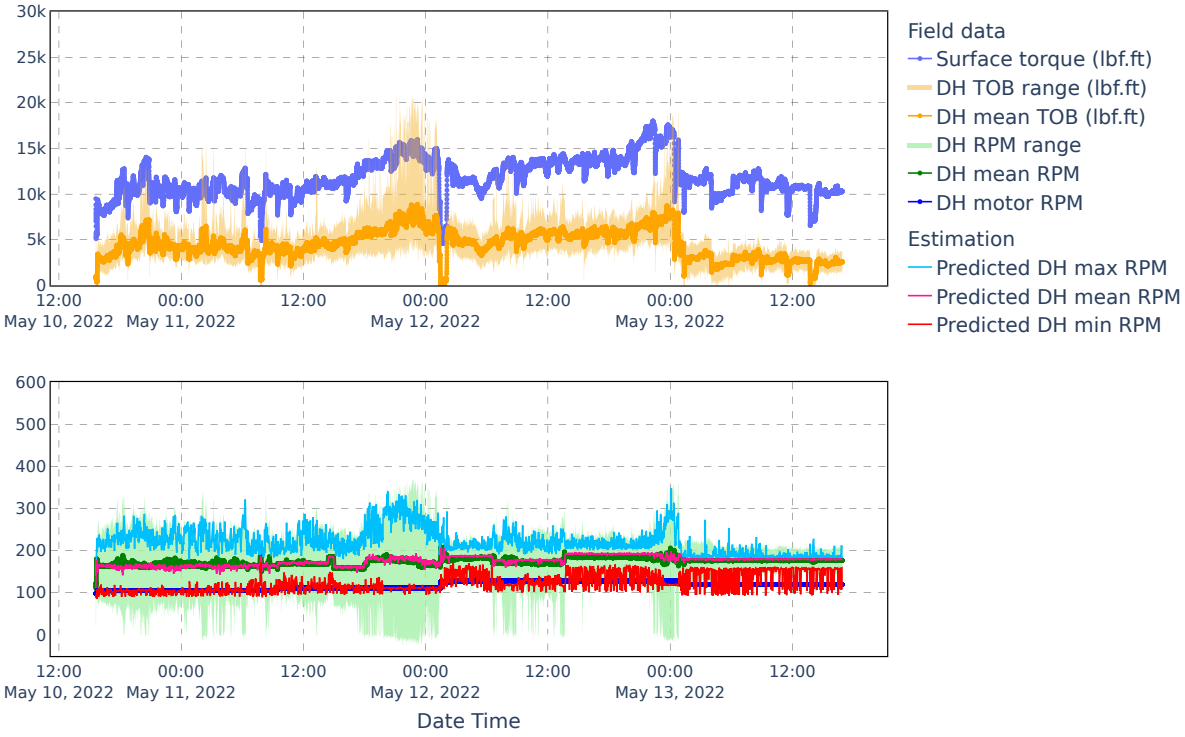


(a) Well 1: PINN

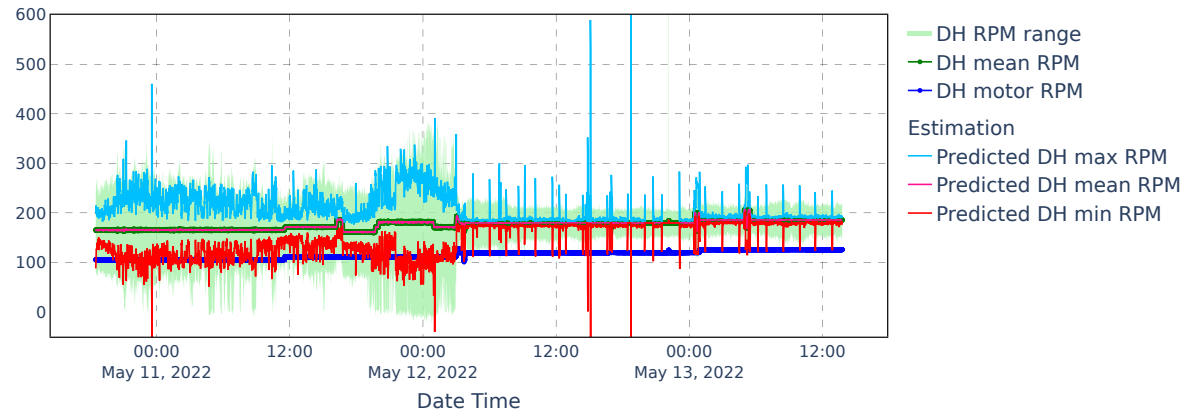


(b) Well 1: PSKF

Figure 9.7

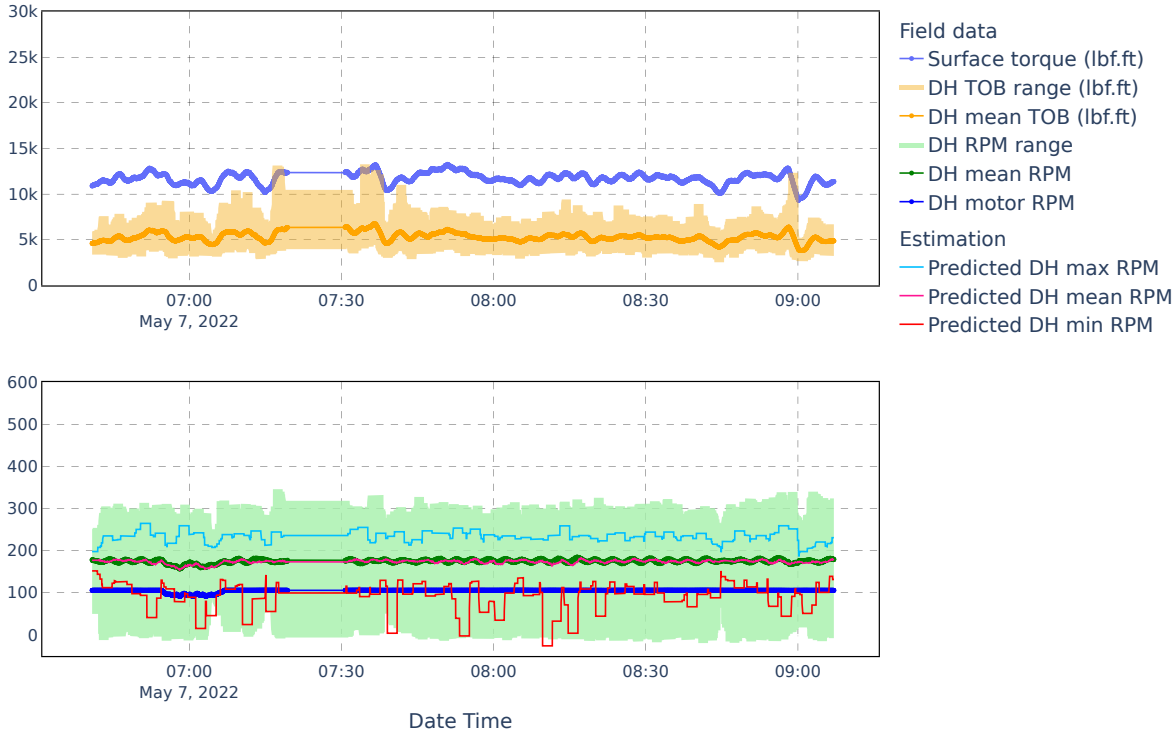


(c) Well 7: PINN

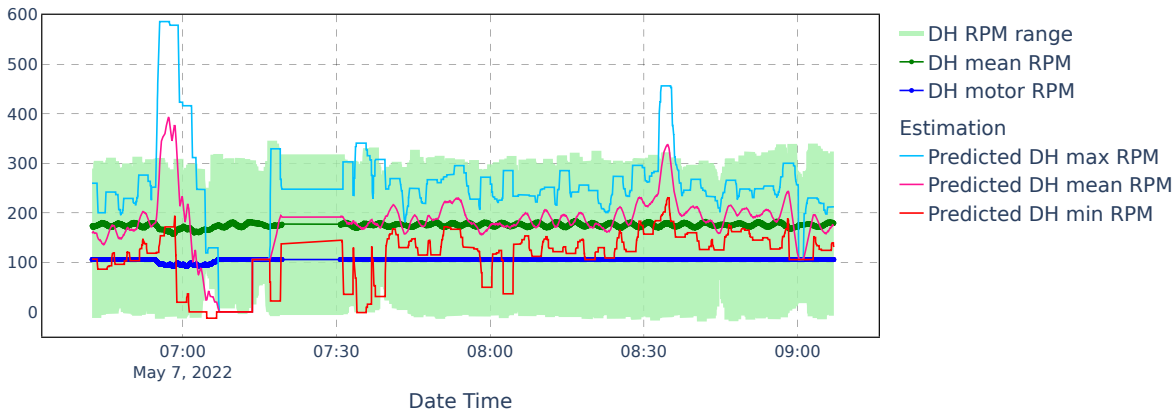


(d) Well 7: PSKF

Figure 9.7

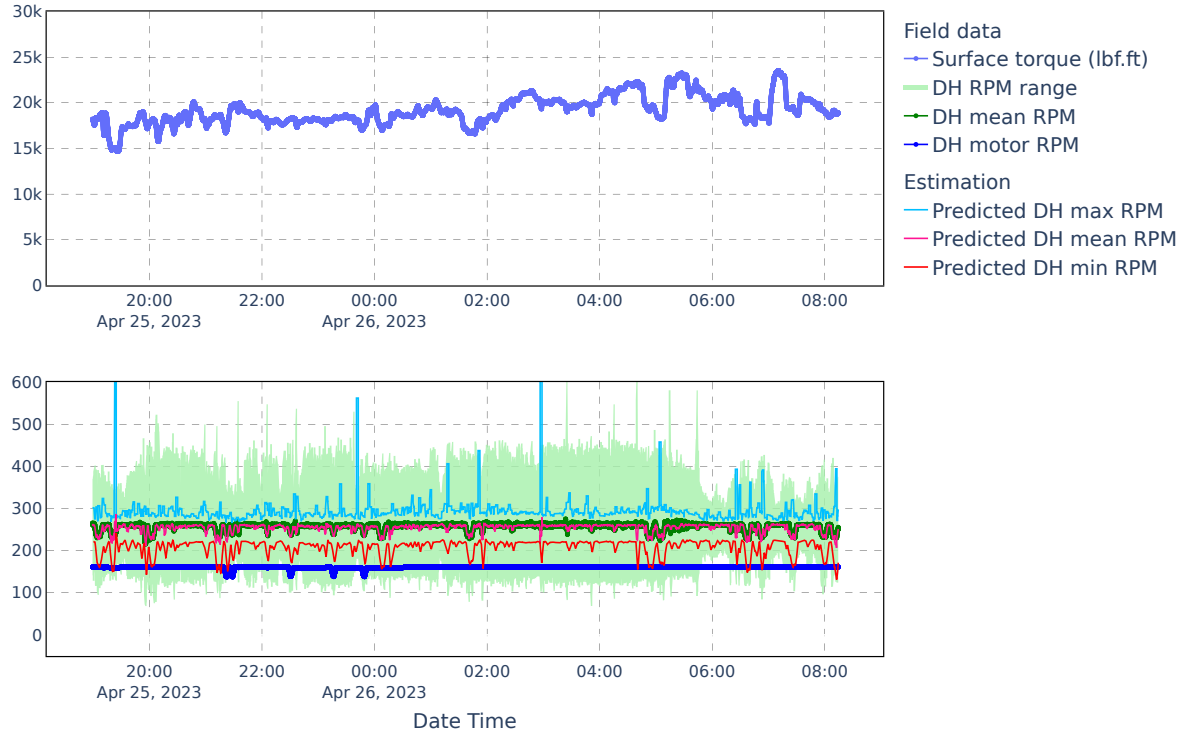


(e) Well 6: PINN

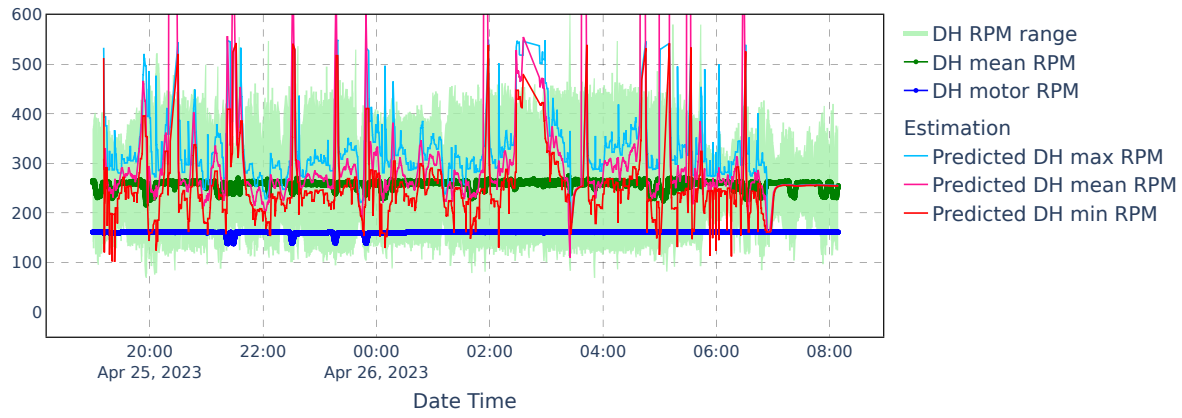


(f) Well 6: PSKF

Figure 9.7



(g) Well 5: PINN



(h) Well 5: PSKF

Figure 9.7: Variations of downhole bit rotation speed measurements and predicted values $\hat{\omega}_b$ obtained with (a) PINN and (b) PSKF, using the real surface torque and surface rotational speed as inputs for test wells 1,7,6,5, respectively. The moving minimum, maximum, and average of both the measured and predicted bit rotation speeds, along with the moving average of the downhole motor rotation speed, are calculated over a 120-second window. Definitions of the terms shown: DH TOB range - maximum-to-minimum range of measured downhole TOB; DH mean TOB - mean of measured downhole TOB; DH RPM range – maximum-to-minimum range of measured downhole bit rotation speed; DH mean RPM – mean of measured downhole bit rotation speed; DH motor RPM – measured downhole motor rotation speed; Predicted DH max, mean, and min RPM – moving maximum, mean, and minimum of the predicted downhole bit rotation speed by PINN within the 120-second window.

Note: As the PINN predicts the friction coefficients and bit rotation speed using 30-second sequences of surface data as inputs (unlike the PSKF, which uses the data directly), missing data can affect the results. For instance, if there is a gap in the measurements and we do not have 30 successive seconds of data, that segment cannot be used by the PINN. Consequently, some differences may appear between the figures of the PINN and the PSKF, as they are not based on exactly the same data.

Tables 9.4a, 9.4b, 9.4c, and 9.4d summarize the performance of the PINN and PSKF on the drilling data of wells 1, 7, 6, and 5, respectively, using the metrics MAE, r, and DTW. The metrics are computed between the moving maximum, mean, and minimum of the field downhole measurements and the corresponding moving statistics of the predicted bit rotation speeds. The improvement row is calculated as the relative error reduction with respect to PSKF, according to the following formula:

$$\text{Improvement } (\%) = \frac{\text{Metric}_{\text{PSKF}} - \text{Metric}_{\text{PINN}}}{\text{Metric}_{\text{PSKF}}} \times 100 \tag{9.1}$$

A positive improvement value for MAE and DTW indicates that PINN outperforms PSKF, whereas for the correlation metric (r), a positive value indicates that PSKF performs better.

Method	Moving mean			Moving maximum			Moving minimum		
	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r
PINN	7.06	4.67	0.97	36.13	30.80	0.80	3.77	1.49	0.98
PSKF	1.18×10^{-17}	1.2×10^{-17}	0.99	57.06	42.44	0.46	32.03	25.50	0.70
Improvement (%)	100	100	-1.98	36.7	27.4	42.1	88.2	94.1	40.0

(a) Test well 1

Method	Moving mean			Moving maximum			Moving minimum		
	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r
PINN	4.94	3.11	0.83	25.13	18.96	0.85	44.39	33.04	0.62
PSKF	10.90	10.30	0.22	66.46	42.07	-0.16	89.30	88.13	-0.26
Improvement (%)	54.7	69.8	62.0	62.2	54.9	-431.3	50.3	62.5	138.5

(b) Test well 7

Method	Moving mean			Moving maximum			Moving minimum		
	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r
PINN	2.90	1.29	0.68	74.96	73.47	0.31	93.21	71.84	-0.01
PSKF	39.65	34.64	-0.13	84.92	70.74	0.07	116.78	114.09	-0.39
Improvement (%)	92.7	96.3	-623.1	11.7	3.8	-342.9	20.2	37.0	97.4

(c) Test well 6

Method	Moving mean			Moving maximum			Moving minimum		
	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r	MAE (rpm)	DTW (rpm)	r
PINN	4.43	4.01	0.83	125.49	119.45	-0.01	78.13	66.76	0.34
PSKF	5250.38	5245.73	0.01	116.26	69.75	0.20	119.21	109.75	-0.01
Improvement (%)	99.91	99.91	-8200	-7.93	-71.25	95	34.46	39.17	-3300

(d) Test well 5

Table 9.4: MAE, Pearson correlation coefficient (r), and normalized DTW between the mean, maximum, and minimum of the field downhole measured rotation speeds and those of the predicted bit rotation speeds obtained with both methods, PINN and PSKF, for: (a) test well 1, (b) test well 7, (c) test well 6, and (d) test well 5. The improvement row is computed as relative error reduction with respect to PSKF using Eq. (9.1).

As mentioned above, all four wells considered are lateral wells, where both the PINN and PSKF rely on the one-mass lumped torsional dynamic model, and the results are derived from 1-Hz surface measurements. For well 1 (see Fig. 9.7 and Table 9.4a), both methods successfully reproduce the downhole speed under severe stick-slip conditions. However, the PINN achieves higher accuracy, particularly in capturing the variations of the maximum and minimum values. For instance, the PINN reduces the MAE by 36.7 % for the maximum and by 88.2 % for the minimum compared to the PSKF. For well 7, as previously noted, bit-sticking phenomena occur, making it impossible to accurately reproduce the downhole minimum speed from 1-Hz surface data [40]. Consequently, the error for the minimum is significantly larger than in well 1 for both methods. Nevertheless, both methods accurately reproduce the maximum and mean bit rotation speed, with the PINN again achieving superior accuracy.

An additional limitation of the PSKF can be observed in Fig. 9.7b. The filter produces unrealistically large peaks for both maximum and minimum values, and in some sequences the moving maximum, minimum, and mean of the predicted downhole velocity even overlap. This indicates divergence, as the PSKF fails to track the downhole vibrations, for instance, between October 1 and October 2. This behavior is strongly influenced by the calibration of the covariance matrices used in the PSKF. High sensitivity to sensor failure could be reduced by increasing the covariance matrix associated to noise, at the expense of a slower convergence. A comprehensive study of the PSKF tuning is, in this regard, necessary, but left for future work.

For wells 5 and 6, both methods show large prediction errors, most notably in well 5, struggling to capture the downhole maximum and minimum variations of bit velocity. The difference lies in how surface measurements reflect what happens downhole. In wells 1 and 7, strong variations in downhole velocity are mirrored by fluctuations in surface torque, signaling that vibrations travel up the string. This link provides the models with valuable information to reconstruct downhole behavior more accurately. In contrast, wells 5 and 6 tell a different story: despite pronounced downhole velocity changes, the surface torque remains deceptively smooth. This disconnect suggests that vibrations are dampened before reaching the surface. A plausible explanation is their steep trajectory profile, transitioning sharply from vertical to horizontal (0–90°), and that the available test data lies entirely in the horizontal section and beyond (see Fig. 9.3b and Fig. 9.3c), as a result the surface signal carries little trace of the underlying downhole dynamics, making accurate prediction far more challenging. A possible way to address this issue is to increase the complexity of the physical model embedded in both methods. A corresponding test with the PSKF on simulated data is presented in the next subsubsection.

9.3.2.3 Prediction of severe downhole stick-slip for sequences with extremely low TVI

Unlike the one-mass lumped torsional model, the two-mass model Eq. (6.8) is able to reproduce downhole stick-slip oscillations with small surface torque variations. We illustrate this here and test the PSKF, embedded with the same two-mass torsional dynamic model used to generate the data.

For this purpose, a sequence is generated at a depth of 19007 ft, along the trajectory of well 7 (see Fig. 3.7), with a constant surface velocity of 60 rpm and the stiction friction model applied (see Section 6.3.2.2). The inertia, stiffness, and normal force parameters of both masses used to generate the sequence are presented in Table 9.5. The true friction coefficients are set to $\mu_s = 0.5$ and $\mu_d = 0.4$. The results are displayed in Fig. 9.8, where Fig. 9.8a shows the 1-Hz surface torque used as input for the PSKF, Fig. 9.8b presents the true and predicted bit rotation speeds, and Fig. 9.9 shows the true and predicted friction coefficients, with initial values

set to 0.4 and 0.3 for the static and dynamic friction, respectively. As can be observed, the PSKF with the two-mass lumped torsional dynamic model is able to reproduce severe downhole vibrations from the 1-Hz smooth surface measurements and converges to the true friction coefficients. Such results cannot be obtained with the PSKF based on the one-mass lumped torsional dynamic model.

Parameter	I_{Top} (kg.m²)	I_1 (kg.m²)	I_2 (kg.m²)	K_1 (N.m)	K_2 (N.m)	F_1^N (N)	F_2^N (N)
Value	2900	2815.57	746.67	75.06	299.65	0	43125.13

Table 9.5: Parameter values of the two-mass lumped torsional model (Eq. (6.8)) used to generate the test sequence shown in Fig. 9.8.

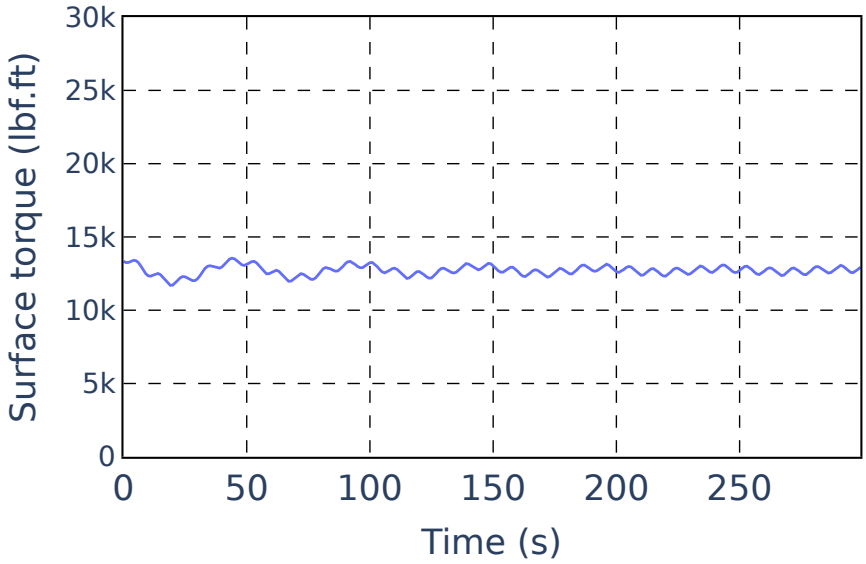
The complexity of the physical model can be further increased, however, as shown in Section 8.8.1.2, increasing the model complexity within the PSKF reduces the basin of attraction of the observer and requires high-frequency measurements to ensure convergence. For this reason, the PSKF with the two-mass model is tested on the data of wells 5 and 6, but it failed to converge when using the 1-Hz surface data. A similar outcome is observed with the PINN, which is also tested on both wells: the optimizer struggles to reduce the loss and gets trapped in a local minimum. With high probability, significantly better results could be achieved by using high-frequency surface measurements as inputs, which are not available in our case, thus limiting our study.

9.3.2.4 Conclusion

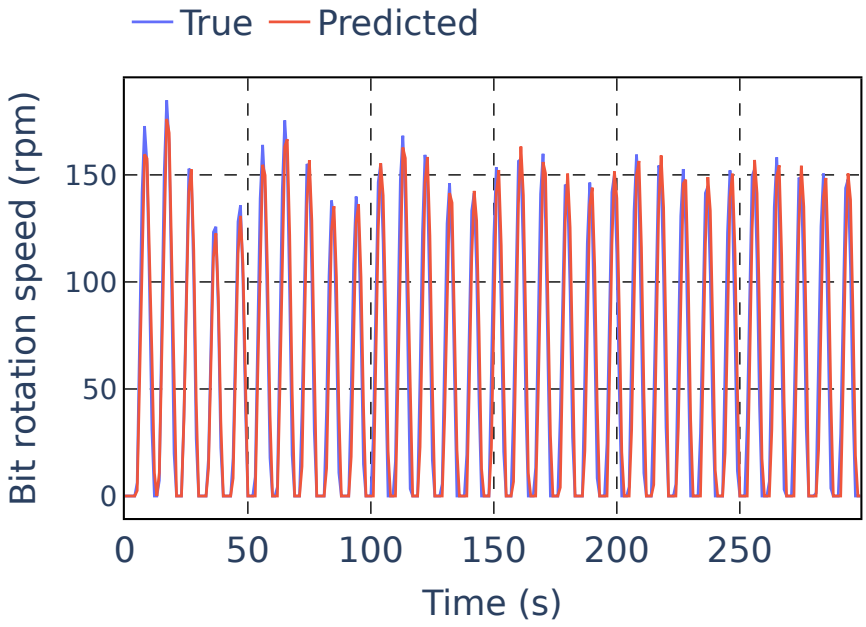
The PSKF and the PINN each present distinct advantages and limitations. The PSKF is an online estimator that updates instantly with incoming data, making it lightweight and fast. In contrast, the PINN requires offline training, about 10 minutes for a 30-second sequence sampled at 1 Hz, which makes it more computationally demanding. Both methods rely on parameter tuning: the PSKF depends on accurate initialization and properly chosen noise covariance matrices, while the PINN requires careful adjustment of the loss function's weighting coefficients. Their limitations also differ: the PINN can suffer from long training times and convergence to local minima, whereas the PSKF may diverge if initial values or measurements are unreliable.

Across both short (Section 9.3.2.1) and long sequences (Section 9.3.2.2), the PINN generally outperforms the PSKF. This is largely due to its robustness to noisy data. The PSKF requires relatively clean and high-quality data to ensure stability and convergence, since measurement noise can directly destabilize the filtering process (see Section 9.3.2.2). In contrast, the PINN is trained by minimizing a loss function that combines data fitting with physics-informed regularization. This joint formulation allows the network to balance between following noisy data and respecting the underlying physical laws, which naturally smooths out noise effects and leads to more reliable estimates.

Nevertheless, the PSKF retains an important advantage: flexibility. Increasing the complexity of the physical model is straightforward in the PSKF, as it only requires modifying the state-space equations. Extending a PINN, on the other hand, demands adding new terms to the loss function and carefully tuning additional coefficients, which quickly increases optimization complexity. Thus, when high-quality, high-frequency data are available, the PSKF remains a competitive choice due to its simplicity and efficiency. By comparison, a PINN with a highly



(a) Surface torque



(b) Bit rotation speed

Figure 9.8: Prediction results of bit rotation speed and friction coefficients for a sequence generated using the two-mass lumped torsional dynamic model (Eq. (6.8)), with the PSKF embedded with the same physical model. The surface rotation speed is constant to 60 rpm. Shown are: (a) Surface torque; and (b) true and predicted bit rotation speed.

complex loss function can become computationally expensive and more prone to convergence

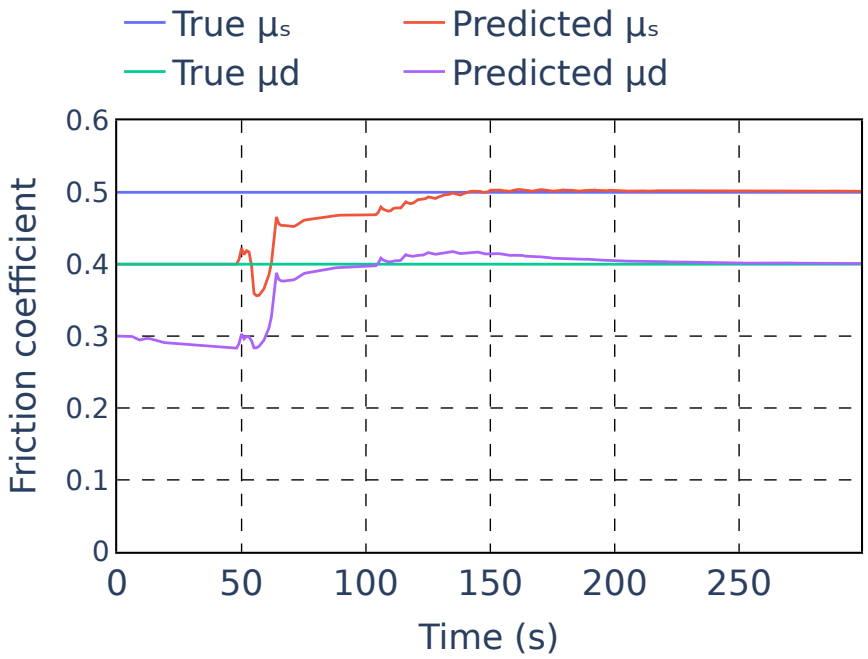


Figure 9.9: True and predicted friction coefficient variations over time at 1 Hz, corresponding to the sequence shown in Fig. 9.8.

issues, such as getting trapped in local minima.

9.4 Conclusion

This chapter evaluates two approaches, PINN and PSKF (introduced in Chapter 7 and Chapter 8), for predicting the downhole bit velocity and friction coefficients using 1-Hz surface measurements and a simplified one-mass lumped torsional dynamic model. The performance of both methods is assessed on PDE-generated simulated sequences as well as field data. On simulated data, MATLAB's `nlgreyest` from the System Identification Toolbox is used as a state-of-the-art benchmark. Both PINN and PSKF outperform `nlgreyest` in predicting friction coefficients when relying on the same embedded physical model, with `nlgreyest` errors exceeding those of PINN by 40.35% for μ_s and 57.75% for μ_d.

For field data, the predicted downhole rotational speed is compared with direct downhole measurements. Both methods are able to reproduce severe stick–slip dynamics observed in some wells, with PINN achieving higher accuracy than PSKF. However, PINN requires considerably more computation time for training and inference, whereas PSKF provides predictions in real-time. In highly inclined wells and within the far end of horizontal sections, where downhole vibrations are weakly transmitted to the surface, both methods fail to capture the downhole dynamics. A potential remedy is to increase the complexity of the embedded physical model, an approach that has already shown success on simulated data, and to employ higher-frequency surface measurements.

Regarding the estimation of friction coefficients from field data, the absence of ground-truth measurements, the limited quantity of available data, and the lack of high-frequency surface and downhole measurements render the analysis inconclusive. Consequently, no significant trends

or conclusions could be drawn. This aspect of the study is therefore been deferred to future work, with the prospect of collecting higher-frequency measurements and a larger dataset to enable a more comprehensive investigation of the variations in the predicted friction coefficients.

Chapter 10

Conclusion - version française

Cette thèse porte sur l'estimation de la sévérité des vibrations de type "stick-slip" au fond du puits lors du forage, en utilisant comme entrée les mesures de surface enregistrées à des fréquences d'échantillonnage faibles. L'étude est organisée en deux grandes parties. Dans la première, l'Indice de Stick-Slip (SSI) est prédit à l'aide d'un réseau de neurones entraîné sur des séries temporelles de mesures de surface. L'objectif principal est d'explorer des techniques permettant d'améliorer la généralisation du modèle sans augmenter la quantité de données d'apprentissage. Pour cela, diverses stratégies sont étudiées, telles que la normalisation des données, l'ajout de variables fondées sur la physique, le transfert d'apprentissage et les méthodes de généralisation de domaine.

La seconde partie traite la prédiction de la vitesse de rotation du trépan et des coefficients de frottement entre la tige de forage et le puits, en s'appuyant sur les mesures de surface combinées à un modèle physique simplifié. Deux approches sont étudiées : un réseau de neurones informé par la physique ou "Physics-Informed Neural Network (PINN)", qui approxime la solution tout en intégrant un modèle de frottement lisse, et le "Projected Salted Kalman Filter (PSKF)", une version améliorée du "Salted Kalman Filter (SKF)" fonctionnant en ligne et tenant explicitement compte du comportement stick-slip à travers un modèle de frottement par adhérence. Les deux méthodes sont évaluées sur des données simulées et réelles, démontrant leur potentiel pour des applications pratiques.

Dans l'ensemble, les résultats ouvrent plusieurs pistes prometteuses pour des travaux futurs.

Défis et opportunités liés aux données

La plupart des défis liés à la prédiction de la sévérité des vibrations et à la détection d'incidents concernent la **qualité et la disponibilité des données**. La première limite est le **déséquilibre entre les mesures de surface et celles du fond du puits**. Les données de surface sont généralement enregistrées pour l'ensemble du puits, tandis que les données du fond ne sont disponibles que sur de courts intervalles. Comme les modèles de la première partie de ce travail sont supervisés, seules les sections se recouvrant peuvent être utilisées, laissant une grande quantité de données inutilisées. Une voie d'amélioration serait d'explorer l'Apprentissage Auto-Supervisé

ou "Self-Supervised Learning (SSL)" [266], qui réduit le besoin de jeux de données annotés en apprenant à prédire une partie de l'entrée à partir d'une autre. Une autre possibilité serait la détection non supervisée d'anomalies [267], permettant d'identifier les événements de stick-slip sévères sans étiquettes. Bien que cette méthode ne prédise pas directement le SSI, elle constitue une piste de recherche prometteuse.

Une seconde limite concerne le **problème de mise à zéro** du poids sur l'outil (WOB) et de la pression différentielle estimés à la surface. Comme expliqué précédemment, le WOB de surface est déduit de la différence entre la charge sur crochet hors du fond et celle sur le fond, la position hors du fond étant identifiée manuellement par le foreur. Cela introduit une première source d'erreur due à l'incertitude dans la détermination de la position adéquate hors du fond. De plus, dans les puits déviés ou horizontaux, les frottements accentuent les distorsions de calcul, entraînant des valeurs inexactes. Un problème similaire apparaît pour la pression différentielle. Une solution potentielle est d'utiliser des algorithmes de mise à zéro automatique, tels que celui proposé par Neufeldt et al. [268], capables de détecter les erreurs dues à une mise à zéro incohérente ou oubliée.

Une troisième limite réside dans le **manque de standardisation** entre les jeux de données issus de différents puits. Les unités et les noms de variables ne sont pas uniformes, obligeant à des ajustements manuels pour chaque jeu de données, ce qui est coûteux en temps et en effort. Une meilleure interopérabilité des systèmes est nécessaire dans l'industrie du forage. Un exemple est la norme "Drilling and Wells Interoperability Standard Industry (D-WIS)", qui promeut une gestion standardisée des données et une intégration facilitée.

D'autres défis, non liés à la qualité des données, concernent l'étape de **détection des opérations** lors du prétraitement des données. Actuellement, les opérations de forage (par exemple, forage sur le fond, remontée ou descente de la tige) sont identifiées manuellement à l'aide de seuils définis par l'utilisateur. Ce processus est subjectif et manque de cohérence. Une amélioration naturelle consisterait en une détection automatique des opérations via des classifieurs d'apprentissage automatique, comme proposé dans [269, 270].

Une autre difficulté concerne les **calculs physiques**, tant pour le calcul des caractéristiques du Chapitre 4, et pour l'estimation des forces de contact normales, de l'inertie, de la rigidité et de l'amortissement dans la tige de forage de la seconde partie de cette thèse. Ces calculs nécessitent une description complète de chaque BHA, comprenant parfois des dizaines de composants. La collecte et la manipulation de ces informations sont chronophages et sujettes à erreur. L'automatisation de ce processus à l'aide de cadres basés sur des grands modèles de langage ou "Large Language Model (LLM)" [271], combinés à des méthodes d'apprentissage profond pour extraire des informations opérationnelles à partir de documents non structurés [272], pourrait faire gagner à la fois du temps et de l'effort. Une autre application prometteuse des LLMs est l'analyse des rapports de forage, riches en informations sur les incidents, leurs causes, leur chronologie et leur contexte [273], ce qui simplifierait la collecte et l'annotation des données pour chaque type d'incident.

Au-delà du modèle de régression de "Stick-Slip Index"

Dans la première partie de cette thèse, l'accent a été mis sur le phénomène de stick-slip, une **vibration de basse fréquence** détectable à la fréquence d'échantillonnage de 1 Hz. Cela exclut les vibrations de plus haute fréquence, telles que le rebond du trépan. Avec l'accès à des données à fréquence plus élevée, les méthodes proposées pourraient être étendues à ces

cas. De même, la détection de coincement de tige pourrait être explorée, bien que la rareté des données hors du fond et la difficulté d'annotation demeurent des obstacles. Même avec de petits jeux de données annotés, les méthodes développées ici pourraient améliorer la capacité de généralisation.

Au Chapitre 4, une **augmentation de données** a été réalisée à partir de données simulées et réelles afin de créer davantage de séquences présentant un stick-slip sévère. Cependant, la génération de grands jeux de données simulées est coûteuse en calcul, tandis que les données réelles sont limitées. Une alternative serait l'utilisation de *TimeGAN* [274], capable de générer des séries temporelles réalistes comportant du stick-slip sévère pour équilibrer le jeu de données et améliorer l'apprentissage.

Une autre direction consiste à développer des **modèles intégrés**. Au lieu de s'appuyer sur une seule architecture, telle que les LSTM, la combinaison de plusieurs pourrait tirer parti de leurs forces respectives. Les couches convolutionnelles sont efficaces pour capturer les motifs locaux, tandis que les couches récurrentes saisissent les dépendances temporelles. Les modèles hybrides ont montré une amélioration en termes de précision, de robustesse et d'interprétabilité [275, 276].

Une application des modèles proposés pour la prédiction du SSI est la **mise à jour en temps réel d'une carte thermique du SSI** en fonction des paramètres de forage, afin de guider leur sélection. En effet, pour améliorer l'efficacité du forage et minimiser les vibrations, l'optimisation en temps réel des paramètres tels que le WOB et la vitesse totale de rotation est essentielle. Les foreurs sont souvent confrontés à un compromis: augmenter le WOB peut déclencher le stick-slip, tandis qu'augmenter la vitesse de rotation peut provoquer des vibrations latérales. Maintenir les deux faibles réduit les vibrations mais limite la vitesse de pénétration (ROP) [277]. Pour identifier les combinaisons optimales de paramètres, une carte thermique peut être générée afin de visualiser les variations du SSI en fonction du WOB et de la vitesse de rotation. Pendant le forage, cette carte peut être mise à jour en temps réel à partir des mesures de surface et des prédictions de SSI issues du modèle entraîné. Les données de surface à 1 Hz sont introduites dans le modèle pour prédire le SSI, et la carte est continuellement affinée à l'aide de ces prédictions et mesures correspondantes. Un processus gaussien ou "Gaussian Process" (GP) peut être employé pour cette mise à jour, car il permet de modéliser de manière flexible les relations complexes entre variables sans supposer de forme fonctionnelle prédéfinie.

Une autre piste, au-delà de la prédiction de la sévérité des vibrations de torsion au fond du puits, est **l'optimisation de la vitesse de pénétration (ROP)**. À partir des données de surface en fonction de la profondeur, un modèle de régression du ROP peut être entraîné pour prédire la vitesse de pénétration, puis couplé à un algorithme d'optimisation afin de déterminer les paramètres de forage optimaux pour chaque profondeur D, tout en respectant les contraintes opérationnelles. Le transfert d'apprentissage pourrait également être appliqué pour tenir compte de la variabilité des formations traversées. En parallèle, la modélisation prédictive des vibrations peut aider à identifier les conditions de résonance. La tige de forage étant continuellement soumise à des excitations externes, une résonance peut apparaître si la vitesse de rotation de surface n'est pas correctement choisie. Pour éviter cela, un modèle physique peut être utilisé afin de calculer les fréquences naturelles de la tige en modes axial, torsionnel et latéral. Ces fréquences correspondent aux vitesses de rotation critiques que l'algorithme d'optimisation doit éviter.

Bien que cette thèse se concentre sur les puits pétroliers, les méthodes pourraient être éten-

dues aux **puits géothermiques**, où la rareté des données est encore plus marquée. Leur application au forage géothermique permettrait de généraliser et d'élargir leur portée.

Méthodes avancées pour l'estimation du frottement

Comme discuté dans cette thèse, un modèle physique simplifié à une masse peut reproduire les variations de la vitesse de rotation du trépan au fond lorsque ces vibrations sont transmises à la surface. Cependant, dans certains cas, les fluctuations au fond sont totalement atténuées lors de la transmission, ne produisant aucune variation observable dans les mesures de surface à basse fréquence. De tels cas ne peuvent être capturés avec le modèle simplifié. Comme montré avec le modèle à deux masses et les données à fréquence plus élevée (voir Section 9.3.2.3), ces cas peuvent être reproduits en simulation sans inclure explicitement un terme d'atténuation dans le modèle. Cela suggère qu'avec des **modèles physiques plus complexes et des données à plus haute fréquence**, il serait possible de reproduire ces variations avec plus de précision.

L'analyse des **variations prédites des coefficients de frottement** à partir de données plus riches constitue un axe clé pour les travaux futurs. Lors de cette analyse, il est important de rappeler que nous utilisons un modèle physique à une masse. Dans ce modèle, le couple de frottement TOB apparaît conjointement avec le couple sur le trépan lié au mécanisme de coupe. Par conséquent, toute sous- ou surestimation du TOB affectera les coefficients de frottement prédits, puisque leurs erreurs s'additionnent. L'erreur d'estimation du TOB peut provenir de deux sources : des inexactitudes dans l'estimation du TOB de surface, liées au problème de mise à zéro, ou du choix du modèle physique utilisé pour représenter l'interaction trépan-roche. Par conséquent, une estimation précise du TOB est essentielle pour garantir que l'analyse des variations prédites des coefficients de frottement soit juste et pertinente.

Pour les PINNs, un **pondérage automatique des termes de la fonction de perte** basé sur des stratégies d'incertitude [219] pourrait améliorer l'apprentissage.

Pour les PSKF, une **analyse formelle de stabilité** demeure un problème ouvert mais fournirait des garanties théoriques plus solides. Tran and Bernard [22] présentent une analyse de convergence pour le cas de temps de saut connus, ce qui diffère de notre cadre, où ces temps sont inconnus.

Une autre piste prometteuse serait **d'étendre le cadre du PSKF aux EDP**. Les travaux présentés dans [155], qui estiment les coefficients de frottement et la vitesse angulaire à partir des seules mesures de surface en se basant sur un modèle torsionnel distribué, reposent sur une conception d'observateur linéaire. Une perspective future serait de combiner ces deux approches afin de développer une formulation PSKF pour les EDP linéaires, où la matrice de saltation pourrait être utilisée pour traiter la non-linéarité associée aux sauts et dériver des lois de mise à jour hybrides pour les coefficients de frottement.

Une autre approche potentielle pour la prédiction des coefficients de frottement et de la vitesse de rotation du trépan consisterait à utiliser une formulation d'optimisation non lisse, où le modèle de frottement par adhérence décrit dans la Section 6.3.2.2 serait représenté non pas via des inclusions différentielles ou des systèmes hybrides, mais au moyen de **contraintes de complémentarité**. Plus précisément, le problème pourrait être formulé comme un programme mathématique à contraintes de complémentarité ou "Mathematical Program with Complementarity Constraints (MPCC)" [278], où le frottement par adhérence serait modélisé sous forme de contraintes de complémentarité au sein du problème d'optimisation.

Chapter 11

Summary and perspectives

This dissertation focuses on estimating the severity of downhole stick-slip vibrations during drilling using surface measurements recorded at low-frequency sampling rates as input. The study is organized into two main parts. In the first part, the Stick-Slip Index (SSI) is predicted using a neural network trained on time series of surface measurements. The primary goal is to explore techniques that improve the generalization of the model without increasing the amount of training data. To achieve this, various strategies are investigated, including data normalization, the inclusion of physics-based features, transfer learning, and Domain Generalization methods. The second part addresses the prediction of bit rotation speed and the friction coefficients between the drillstring and the borehole, leveraging surface measurements alongside a simplified physical model. Two approaches are studied: a Physics-Informed Neural Network (PINN), which approximates the solution while integrating a smooth friction model, and the Projected Salted Kalman Filter (PSKF), an enhanced version of the Salted Kalman Filter (SKF) that operates online and explicitly accounts for stick-slip behavior through a stiction friction model. Both methods are evaluated on simulated and field data, demonstrating their potential for practical applications.

Overall, the results opens several promising directions for future work.

Data-related challenges and opportunities

Most of the challenges for the prediction of vibrations severity and incident detection are related to the **quality and availability of data**. The first limitation is the **imbalance between surface and downhole measurements**. Surface data are usually recorded for entire wells, while downhole data are available only over short intervals. Since the models in the first part of this work are supervised, only overlapping sections could be used, leaving much data unused. One way forward is to explore Self-Supervised Learning (SSL) [266], which reduces the need for labeled datasets by letting the model learn one part of the input from another. Another possibility is unsupervised anomaly detection [267], which could identify severe stick-slip events without requiring labels. While this method does not directly predict the SSI, it remains a promising line of research.

A second limitation concerns the **zeroing problem** of surface-estimated WOB and differential pressure. As explained earlier, surface WOB is inferred from the difference between off-bottom and on-bottom hookload, where the off-bottom position is manually identified by the driller. This introduces the first source of error, due to uncertainty in determining the adequate off-bottom position. In addition, in deviated or horizontal wells, friction further distorts the calculation, leading to inaccurate values. A similar issue arises for differential pressure. A potential solution is to use automatic zeroing algorithms, such as the one proposed by Neufeldt et al. [268], which can detect errors caused by inconsistent or forgotten zeroing.

A third limitation lies in the **lack of standardization** across datasets from different wells. Units and feature names are not consistent, which forces manual adjustments for each dataset, which is time and effort consuming. More interoperable systems are needed in the drilling industry. An example is Drilling and Wells Interoperability Standard Industry (D-WIS), which promotes standardized data handling and easier integration.

Additional challenges, unrelated to data quality, include the **operation detection** step in data preprocessing. Currently, drilling operations (e.g., on-bottom drilling, tripping in/out) are identified manually with user-defined thresholds. This process is inconsistent and subjective. A natural improvement would be automated operation detection using machine learning classifiers, as proposed in [269, 270].

Another difficulty concerns **physical calculations**, both for feature computation in Chapter 4 and for the estimation of normal contact forces, inertia, rigidity, and damping in the drill string of the second part of this thesis. These calculations require complete descriptions of each BHA, sometimes involving dozens of components. Collecting and handling this information is time-consuming and error-prone. Automating the process with Large Language Model (LLM)-based frameworks [271], combined with deep learning methods for extracting operational insights from unstructured documents [272], could save both time and effort. Another promising application of LLMs is the analysis of drilling reports, which contain valuable information about incidents, their causes, timing, and context [273], thereby simplifying data collection and labeling for each specific incident type.

Beyond our Stick-Slip Index Regression model

In the first part of this thesis, the focus has been on stick-slip, a **low-frequency vibration** detectable at the 1 Hz sampling rate. This excludes higher-frequency vibrations such as bit-bounce. With access to higher-frequency data, the proposed methods could be extended to such cases. Similarly, stuck-pipe detection could be explored, although limited off-bottom data and the difficulty of labeling remain challenges. Even in cases with only small labeled datasets, the methods developed here may still improve generalizability.

In Chapter 4, **data augmentation** is performed using both field and simulated data to create more sequences with severe stick-slip. However, generating large simulated datasets is computationally expensive, while field data are limited. An alternative is TimeGAN [274], which can generate realistic time series sequences with severe stick-slip to balance the dataset and improve training.

Another direction is the development of **integrated models**. Instead of relying on a single architecture, such as LSTM, combining multiple ones could leverage their strengths. Convolutional layers are effective for local patterns, while recurrent layers capture temporal dependencies. Hybrid models have been shown to improve accuracy, robustness, and interpretabil-

ity [275, 276].

An application of the proposed models for SSI prediction is **the real-time updating of an SSI heatmap** based on drilling parameters to guide their selection. Indeed, to improve drilling efficiency and minimize vibrations, real-time optimization of drilling parameters such as WOB and total rotation speed is essential. Drillers often face a trade-off: increasing WOB can trigger stick-slip, while raising rotation speed may cause lateral vibrations. Keeping both low reduces vibrations but limits the Rate of Penetration (ROP) [277]. To identify optimal parameter combinations, a heatmap can be generated to visualize SSI variations with respect to WOB and rotation speed, for instance at a fixed depth D. During drilling, this heatmap can be updated in real time using surface measurements and SSI predictions from the trained model. The 1 Hz surface data are fed into the model to predict SSI, and the heatmap is continuously refined using these predictions and the corresponding measurements. A Gaussian Process (GP) can be employed for this update, as it can flexibly model complex relationships between variables without assuming a predefined functional form.

Another potential direction, beyond predicting the severity of downhole torsional vibrations, is **the optimization of the Rate of Penetration (ROP)**. Using depth-based surface data, an ROP regression model can be trained to predict ROP, which can then be coupled with an optimization algorithm to determine the optimal drilling parameters for each depth D, while respecting operational constraints. Transfer learning could also be applied to account for formation variability during drilling. In parallel, predictive vibration modeling can help identify resonance conditions. The drill string is continuously subjected to external excitations that may induce resonance if the surface rotation speed is not properly selected. To prevent this, a physical model can be used to compute the natural frequencies of the drill string in axial, torsional, and lateral modes. These frequencies correspond to critical surface rotation speeds that the optimization algorithm must avoid.

While this dissertation focuses on petroleum wells, the methods could be extended to **geothermal wells**, where data scarcity is even more pronounced. Applying these approaches to geothermal drilling would help generalize and broaden their impact.

Advanced methods for friction estimation

As discussed in this dissertation, a simplified one-mass physical model can reproduce variations in downhole bit rotation speed when these vibrations are transmitted to the surface. However, in some cases, downhole fluctuations are completely attenuated during transmission, resulting in no observable variation in the surface measurements at low frequency. Such cases cannot be captured with the simplified model. As shown with the two-mass lumped model and higher-frequency data (see Section 9.3.2.3), such cases can be reproduced in simulation even without including an explicit attenuation term in the model. This suggests that, with **more complex physical models and higher-frequency data**, it may be possible to capture these variations more accurately.

Analyzing **predicted friction coefficient variations** with richer data is kep for future work. When analyzing the predicted variations of the friction coefficients, it is important to remember that we use a single-mass physical model. In this model, the friction torque appears alongside the TOB related to the cutting mechanism. As a result, any under- or overestimation of the TOB will affect the predicted friction coefficients, since their errors add up. The TOB estimation error may come from two sources: inaccuracies in the surface estimated TOB, linked to the zeroing

problem, or from the choice of the physical model used to represent the bit-rock interaction. Therefore, an accurate TOB estimation is important to ensure that the analysis of the predicted friction coefficient variations is fair and meaningful.

For PINNs, **automatic weighting of loss terms** using uncertainty-based strategies [219] could enhance training.

For PSKFs, **a formal stability analysis** remains an open problem but would provide stronger theoretical guarantees. Tran and Bernard [22] present a convergence analysis for the case of known jump times, which differs from our setting, where the jump times are unknown.

Another promising direction is **to extend the PSKF framework to PDEs**. The work presented in [155], which estimates friction coefficients and angular velocity using only surface measurements based on a distributed torsional model, relies on a linear observer design. A potential future work would be to combine both approaches to develop a PSKF formulation for linear PDEs, where the saltation matrix could be used to handle the nonlinearity associated to jumps and derive hybrid-like update laws for friction coefficients.

Another potential approach for predicting the friction coefficients and bit rotation speed is to use a nonsmooth optimization-based formulation, where the stiction friction model described in Section 6.3.2.2 is represented not through differential inclusions or hybrid systems, but via **complementarity constraints**. Specifically, the problem could be formulated as a Mathematical Program with Complementarity Constraints (MPCC) [278], where the stiction friction is modeled as complementarity constraints within the optimization problem.

Bibliography

[1] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[2] Christoph Glocker. *Set-valued force laws: dynamics of non-smooth systems*, volume 1. Springer Science & Business Media, 2013.

[3] Remco I Leine and Henk Nijmeijer. *Dynamics and bifurcations of non-smooth mechanical systems*, volume 18. Springer Science & Business Media, 2013.

[4] Kamaljyoti Nath, Xuhui Meng, Daniel J Smith, and George Em Karniadakis. Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines. *Scientific Reports*, 13(1):13683, 2023.

[5] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

[6] Giuseppe Ciaburro and Gino Iannace. Machine learning-based algorithms to knowledge extraction from time series data: A review. *Data*, 6(6):55, 2021.

[7] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

[8] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[9] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[12] Chao-Lung Yang, Chen-Yi Yang, Zhi-Xuan Chen, and Nai-Wei Lo. Multivariate time series data transformation for convolutional neural network. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 188–192. IEEE, 2019.

[13] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[14] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[15] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[16] Pauline Bernard. *Observer design for nonlinear systems*, volume 479. Springer, 2019.

[17] Antonio Tornambè. High-gain observers for non-linear systems. *International Journal of Systems Science*, 23(9):1475–1489, 1992.

[18] Keisuke Fujii. Extended kalman filter. *Refernce Manual*, 14:41, 2013.

[19] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE control systems magazine*, 29(2):28–93, 2009.

[20] Guy Ackerson and K Fu. On state estimation in switching environments. *IEEE transactions on automatic control*, 15(1):10–17, 1970.

[21] Yong Liu. Switching observer design for uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 42(12):1699–1703, 1997.

[22] Gia Quoc Bao Tran and Pauline Bernard. Kalman-like observer for hybrid systems with linear maps and known jump times. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1865–1872. IEEE, 2023.

[23] Nathan J Kong, J Joe Payne, George Council, and Aaron M Johnson. The salted kalman filter: Kalman filtering on hybrid dynamical systems. *Automatica*, 131:109752, 2021.

[24] Maurer Anderson, Cooper Hood, and Cook. Deep drilling basic research: volume 4-system description. final report. Technical report, Maurer Engineering Inc., Houston, TX and University of California, 1990.

[25] Thomas Marc Richard. *Self-excited stick-slip oscillations of drag bits*. University of Minnesota, 2001.

[26] Hana Yahia, Thomas Romary, Laurent Gerbaud, Bruno Figluizzi, Florent Di Meglio, Stephane Menand, and Mohamed Mahjoub. Combining machine-learning and physics-based models to mitigate stick-slip in real-time. In *SPE/IADC Drilling Conference and Exhibition*. SPE, 2024.

[27] Hana Yahia, Thomas Romary, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Real-time stick-slip mitigation using combined machine learning and physics based techniques. In *International Petroleum Technology Conference*. IPTC, 2024.

[28] Hana YAHIA, Bruno Figliuzzi, Florent Di Meglio, Stephane Menand, Mohamed MAHJOUB, et al. Domain generalization for time series: Enhancing drilling regression models for stick-slip index prediction. *Transactions on Machine Learning Research*.

[29] Hana Yahia, Florent DI Meglio, Bruno Figliuzzi, Laurent Gerbaud, Stephane Menand, and Mohamed Mahjoub. Physics-informed neural network to estimate downhole torsional vibrations and friction coefficients. In *Abu Dhabi International Petroleum Exhibition and Conference*. SPE, 2025.

[30] Yuelin Shen, Zhengxin Zhang, Jie Zhao, Wei Chen, Mohammad Hamzah, Richard Harmer, and Geoff Downton. The origin and mechanism of severe stick-slip. In *SPE annual technical conference and exhibition*. OnePetro, 2017.

[31] Xiaohua Zhu, Liping Tang, and Qiming Yang. A literature review of approaches for stick-slip vibration suppression in oilwell drillstring. *Advances in Mechanical Engineering*, 6: 967952, 2014.

[32] Brian Armstrong-Helouvry. Stick-slip arising from stribeck friction. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1377–1382. IEEE, 1990.

[33] U Aarsnes and Nathan van de Wouw. Effect of axial and torsional vibrations on a distributed drill string system. In *5th International Colloquium on Nonlinear Dynamics and Control of Deep Drilling Systems, 27-29 June 2022, University of Maryland, College Park, MD, USa*, 2022.

[34] Thomas Richard, Christophe Germay, and Emmanuel Detournay. A simplified model to explore the root cause of stick–slip vibrations in drilling systems with drag bits. *Journal of sound and vibration*, 305(3):432–456, 2007.

[35] Yevhen Kovalyshen. Understanding root cause of stick–slip vibrations in deep drilling with drag bits. *International Journal of Non-Linear Mechanics*, 67:331–341, 2014.

[36] J Ford Brett. The genesis of torsional drillstring vibrations. *SPE Drilling Engineering*, 7 (03):168–174, 1992.

[37] MP Dufeyte and H Henneuse. Detection and monitoring of the slip-stick motion: field experiments. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–21945. SPE, 1991.

[38] Gilles Pelfrene. *Rôle du processus de forabilité des roches dans les vibrations de torsion des systèmes de forage pétrolier*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2010.

[39] Noël Challamel. *Étude de stabilité d'une structure de forage à partir du processus de destruction de la roche*. PhD thesis, ENSMP, 1999.

[40] Junichi Sugiura and Steve Jones. Measurement of mud motor back-drive dynamics, associated risks, and benefits of real-time detection and mitigation measures. *SPE Drilling & Completion*, 36(03):628–646, 2021.

[41] Saket Srivastava. An experimental, modeling and machine learning based investigation of stick-slip vibrations. 2022.

[42] Åge Kyllingstad. A new technique for calculating downhole rotation speed and torque from surface measurements. In *SPE/IADC Drilling Conference and Exhibition*, page D031S016R005. SPE, 2015.

[43] N Challamel, H Sellami, E Chenevez, and L Gossuin. A stick-slip analysis based on rock/bit interaction: theoretical and experimental contribution. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–59230. SPE, 2000.

[44] Jeffrey R Bailey and Stephen M Remmert. Managing drilling vibrations through bha design optimization. *SPE Drilling & Completion*, 25(04):458–471, 2010.

[45] David Watt, Ngoc-Ha Dao, Mohamed Mahjoub, Stephane Menand, and Mark Smith. How motor bend affects stick-slip vibrations: Modeling and case study. In *SPE Annual Technical Conference and Exhibition*, page D031S055R004. SPE, 2022.

[46] Gilles Pelfrene, Hedi Sellami, and Laurent Gerbaud. Mitigating stick-slip in deep drilling based on optimization of pdc bit design. In *SPE/IADC drilling conference and exhibition*, pages SPE–139839. SPE, 2011.

[47] JR Bailey, GS Payette, MT Prim, J Molster, AW Al Mheiri, PG McCormack, and K LeRoy. Mitigating drilling vibrations in a lateral section using a real-time advisory system. In *Abu Dhabi International Petroleum Exhibition and Conference*, page D011S020R004. SPE, 2017.

[48] Prasham Sheth, Indranil Roychoudhury, Crispin Chatar, and José Celaya. A hybrid physics-based and machine-learning approach for stick/slip prediction. In *IADC/SPE International Drilling Conference and Exhibition*. OnePetro, 2022.

[49] Yang Zha and Son Pham. Monitoring downhole drilling vibrations using surface data through deep learning. In *SEG Technical Program Expanded Abstracts 2018*, pages 2101–2105. Society of Exploration Geophysicists, 2018.

[50] Prince Okoli, Juan Cruz Vega, and Roman Shor. Estimating downhole vibration via machine learning techniques using only surface drilling parameters. In *SPE Western Regional Meeting*, page D022S019R005. SPE, 2019.

[51] Maja Ignova, Justo Matheus, Darwin Amaya, and Edward Richards. Recognizing abnormal shock signatures during drilling with help of machine learning. In *SPE Middle East Oil and Gas Show and Conference*, page D031S032R002. SPE, 2019.

[52] Ekaterina Wiktorski, Suranga Geekiyanage, Erik Løken, and Dan Sui. Comparative study of surface and downhole drillstring vibrations measurements on a laboratory-scale drilling rig. In *SPE Norway Subsurface Conference*, page D011S008R002. SPE, 2019.

[53] Behzad Elahifar and Erfan Hosseini. A new approach for real-time prediction of stick–slip vibrations enhancement using model agnostic and supervised machine learning: a case study of norwegian continental shelf. *Journal of Petroleum Exploration and Production Technology*, 14(1):175–201, 2024.

[54] Theresa Baumgartner and Eric van Oort. Pure and coupled drill string vibration pattern recognition in high frequency downhole data. In *SPE Annual Technical Conference and Exhibition*, pages SPE–170955. SPE, 2014.

[55] Chiranth Hegde, Harry Millwater, and Ken Gray. Classification of drilling stick slip severity using machine learning. *Journal of Petroleum Science and Engineering*, 179: 1023–1036, 2019.

[56] Ekaterina Millan, Maurice Ringer, Riadh Boualleg, and Denis Li. Real-time drillstring vibration characterization using machine learning. In *SPE/IADC Drilling Conference and Exhibition*, page D031S016R004. SPE, 2019.

[57] Michael Behounek, Kirt McKenna, Taylor Thetford, Trey Peroyea, Michael Roberts, Julie Pearce, Gary Hickin, Pradeepkumar Ashok, Michael Yi, and Dawson Ramos. Deployment of a hybrid machine learning and physics based drilling advisory system at the rig site for rop optimization. In *SPE/IADC Drilling Conference and Exhibition*, page D021S016R001. SPE, 2023.

[58] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[59] Ramy Saadeldin, Hany Gamal, and Salaheldin Elkatatny. Detecting downhole vibrations through drilling horizontal sections: machine learning study. *Scientific Reports*, 13(1): 6204, 2023.

[60] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[61] Sibarama Panigrahi and HS Behera. Effect of normalization techniques on univariate time series forecasting using evolutionary higher order neural network. *International Journal of Engineering and Advanced Technology*, 3(2):280–285, 2013.

[62] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9):3760–3765, 2019.

[63] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[64] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[65] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[66] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3:1–40, 2016.

[67] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

[68] Justin Salamon and Juan Pablo Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, 24 (3):279–283, 2017.

[69] Ngoc-Ha Dao, Stephane Menand, and Matt Isbell. Mitigating and understanding stick-slip in unconventional wells. In *SPE/IADC International Drilling Conference and Exhibition*. OnePetro, 2019.

[70] Abdessalem Belaid. *Modélisation tridimensionnelle du comportement mécanique de la garniture de forage dans les puits à trajectoires complexes: application à la prédiction des frottements garniture-puits*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2005.

[71] Stephane Menand, Hedi Sellami, Michel Tijani, Olivier Stab, D Dupuis, and Christophe Simon. Advancements in 3d drillstring mechanics: From the bit to the topdrive. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–98965. SPE, 2006.

[72] Adrian MP Braşoveanu and Răzvan Andonie. Visualizing transformers for nlp: a brief survey. In *2020 24th international conference information visualisation (IV)*, pages 270–279. IEEE, 2020.

[73] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[74] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[75] Qiong Ran, Yonghao Zhou, Danfeng Hong, Meiqiao Bi, Li Ni, Xuan Li, and Muhammad Ahmad. Deep transformer and few-shot learning for hyperspectral image classification. *CAAI Transactions on Intelligence Technology*, 8(4):1323–1336, 2023.

[76] Hao Yang, Haoyang Yu, Danfeng Hong, Zhen Xu, Yulei Wang, and Meiping Song. Hyperspectral image classification based on multi-level spectral-spatial transformer network. In *2022 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2022.

[77] Diling Liao, Cuiping Shi, and Liguo Wang. A complementary integrated transformer network for hyperspectral image classification. *CAAI Transactions on Intelligence Technology*, 8(4):1288–1307, 2023.

[78] Zhijia Zhang, Sinan Wu, Xinming Peng, Wanting Wang, and Rui Li. Continuous learning deraining network based on residual fft convolution and contextual transformer module. *IET Image Processing*, 17(3):747–760, 2023.

[79] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[80] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[81] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[82] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *Pattern recognition. ICPR international workshops and challenges: virtual event, January 10–15, 2021, Proceedings, Part III*, pages 694–701. Springer, 2021.

[83] Nanxin Chen, Shinji Watanabe, Jesús Villalba, Piotr Żelasko, and Najim Dehak. Non-autoregressive transformer for speech recognition. *IEEE Signal Processing Letters*, 28: 121–125, 2020.

[84] Mingrui Ma, Lansheng Han, and Chunjie Zhou. Btad: A binary transformer deep neural network model for anomaly detection in multivariate time series data. *Advanced Engineering Informatics*, 56:101949, 2023.

[85] Junwei Shi, Shiqi Wang, Pengfei Qu, and Jianli Shao. Time series prediction model using lstm-transformer neural network for mine water inflow. *Scientific reports*, 14(1):18284, 2024.

[86] Yujie You, Le Zhang, Peng Tao, Suran Liu, and Luonan Chen. Spatiotemporal transformer neural network for time-series forecasting. *Entropy*, 24(11):1651, 2022.

[87] Hailin Li. Time works well: Dynamic time warping based on time weighting for time series data mining. *Information Sciences*, 547:592–608, 2021.

[88] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[89] F Efteland, A Creegan, L Jordan, and C Caraway. The significance of pro-active online monitoring with stick-slip mitigation. In *Abu Dhabi International Petroleum Exhibition and Conference*, page D031S059R001. SPE, 2015.

[90] Daniel Asante Otchere, Tarek Omar Arbi Ganat, Jude Oghenerurie Ojero, Bennet Nii Tackie-Otoo, and Mohamed Yassir Taki. Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions. *Journal of Petroleum Science and Engineering*, 208:109244, 2022.

[91] Daniel Asante Otchere, Tarek Omar Arbi Ganat, Vanessa Nta, Eric Thompson Brantson, and Tushar Sharma. Data analytics and bayesian optimised extreme gradient boosting approach to estimate cut-offs from wireline logs for net reservoir and pay classification. *Applied Soft Computing*, 120:108680, 2022.

[92] Daniel Asante Otchere, Mohammed Abdalla Ayoub Mohammed, Tarek Omar Arbi Ganat, Raoof Gholami, and Zulkifli Merican Aljunid Merican. A novel empirical and deep ensemble super learning approach in predicting reservoir wettability via well logs. *Applied Sciences*, 12(6):2942, 2022.

[93] Chinedu Ejike, Immanuel Frimpong Obuobi, Simon Avinu, Khizar Abid, and Catalin Teodoriu. Investigation and analysis of influential parameters in bottomhole stick–slip calculation during vertical drilling operations. *Energies*, 17(3):622, 2024.

[94] FF Real, DM Lobo, TG Ritto, and FANC Pinto. Experimental analysis of stick-slip in drilling dynamics in a laboratory test-rig. *Journal of Petroleum Science and Engineering*, 170:755–762, 2018.

[95] Eshan K Maitra and Mohammed F Al Dushaishi. How does flow rate affect drillstring vibrations? experimental investigation. 2023.

[96] Alain Berlioz, J Der Hagopian, Régis Dufour, and E Draoui. Dynamic behavior of a drill-string: experimental investigation of lateral instabilities. 1996.

[97] Chao Wang, Wenbo Chen, Zhe Wu, Jun Li, and Gonghui Liu. Stick–slip characteristics of drill strings and the related drilling parameters optimization. *Processes*, 11(9):2783, 2023.

[98] Ehsan Brenjkar, Ebrahim Biniaz Delijani, and Kasra Karroubi. Prediction of penetration rate in drilling operations: a comparative study of three neural network forecast methods. *Journal of Petroleum Exploration and Production*, 11(2):805–818, 2021.

[99] Liping Tang, Xiaohua Zhu, Xudong Qian, and Changshuai Shi. Effects of weight on bit on torsional stick-slip vibration of oilwell drill string. *Journal of Mechanical Science and Technology*, 31:4589–4597, 2017.

[100] Idir Kessai, Samir Benammar, Mohamed Zinelabidine Doghmane, and Kong Fah Tee. Drill bit deformations in rotary drilling systems under large-amplitude stick-slip vibrations. *Applied Sciences*, 10(18):6523, 2020.

[101] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[102] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International conference on machine learning*, pages 222–230. PMLR, 2013.

[103] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE international conference on computer vision*, pages 769–776, 2013.

[104] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.

[105] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

[106] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[107] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[108] Peeyush Singhal, Rahee Walambe, Sheela Ramanna, and Ketan Kotecha. Domain adaptation: challenges, methods, datasets, and applications. *IEEE access*, 11:6973–7020, 2023.

[109] Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5743–5762, 2024.

[110] Yuqi Fang, Pew-Thian Yap, Weili Lin, Hongtu Zhu, and Mingxia Liu. Source-free unsupervised domain adaptation: A survey. *Neural Networks*, page 106230, 2024.

[111] Young-Bum Kim, Karl Stratos, and Dongchan Kim. Adversarial adaptation of synthetic or stale data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1297–1307, 2017.

[112] Brian Xu, Mitra Mohtarami, and James Glass. Adversarial domain adaptation for stance detection. *arXiv preprint arXiv:1902.02401*, 2019.

[113] Xingxing Weng, Yuchun Huang, Yanan Li, He Yang, and Shaohuai Yu. Unsupervised domain adaptation for crack detection. *Automation in Construction*, 153:104939, 2023.

[114] Dimitrios Kollias, Anastasios Arsenos, and Stefanos Kollias. Domain adaptation explainability & fairness in ai for medical image analysis: Diagnosis of covid-19 based on 3-d chest ct-scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4907–4914, 2024.

[115] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[116] Zheng Li, Yun Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. End-to-end adversarial memory network for cross-domain sentiment classification. In *IJCAI*, pages 2237–2243, 2017.

[117] Ruihao Xia, Chaoqiang Zhao, Meng Zheng, Ziyan Wu, Qiyu Sun, and Yang Tang. Cmda: Cross-modality domain adaptation for nighttime semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21572–21581, 2023.

[118] Aakanksha Naik and Carolyn Rose. Towards open domain event trigger identification using adversarial domain adaptation. *arXiv preprint arXiv:2005.11355*, 2020.

[119] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11749–11756, 2020.

[120] Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve. *Machine Learning*, 112(7):2685–2721, 2023.

[121] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and S Yu Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8): 8052–8072, 2022.

[122] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (4):4396–4415, 2022.

[123] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

[124] Chao Zhao and Weiming Shen. Adversarial mutual information-guided single domain generalization network for intelligent fault diagnosis. *IEEE Transactions on Industrial Informatics*, 19(3):2909–2918, 2022.

[125] Xiang Li, Wei Zhang, Hui Ma, Zhong Luo, and Xu Li. Domain generalization in rotating machinery fault diagnostics using deep neural networks. *Neurocomputing*, 403:409–420, 2020.

[126] Mingxin Liu, Jiong Mu, Zitong Yu, Kun Ruan, Baiyi Shu, and Jie Yang. Adversarial learning and decomposition-based domain generalization for face anti-spoofing. *Pattern Recognition Letters*, 155:171–177, 2022.

[127] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8484–8493, 2020.

[128] Bo-Qun Ma, He Li, Yun Luo, and Bao-Liang Lu. Depersonalized cross-subject vigilance estimation with adversarial domain generalization. In *2019 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2019.

[129] Huailiang Zheng, Rixin Wang, Yuantao Yang, Yuqing Li, and Minqiang Xu. Intelligent fault identification based on multisource domain generalization towards actual diagnosis scenario. *IEEE Transactions on Industrial Electronics*, 67(2):1293–1304, 2019.

[130] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[131] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[132] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[133] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[134] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[135] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[136] Theresa Baumgartner. What is wrong with my drilling data? current state and developments of downhole dynamics measurement tools. 2018.

[137] EE Maidla and AK Wojtanowicz. Field comparison of 2-d and 3-d methods for the borehole friction evaluation in directional wells. In *SPE Annual Technical Conference and Exhibition*, pages SPE–16663. SPE, 1987.

[138] HA Locke, JB Johnson, and DC Jewkes. Extended reach appraisal well: a case history. In *SPE International Oil and Gas Conference and Exhibition in China*, pages SPE–17564. SPE, 1988.

[139] Robello Samuel. Friction factors: What are they for torque, drag, vibration, bottom hole assembly and transient surge/swab analyses? *Journal of Petroleum Science and Engineering*, 73(3-4):258–266, 2010.

[140] Andrew Wu and Geir Hareland. Calculation of friction coefficient and downhole weight on bit with finite element analysis of drillstring. In *ARMA US Rock Mechanics/Geomechanics Symposium*, pages ARMA–2012. ARMA, 2012.

[141] Frank Reiber, Bart E Vos, and Svein E Eide. On-line torque & drag: A real-time drilling performance optimization tool. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–52836. SPE, 1999.

[142] GW Halsey, A Kyllingstad, TV Aarrestad, and D Lysne. Drillstring torsional vibrations: Comparison between theory and experiment on a full-scale research drilling rig. In *SPE Annual Technical Conference and Exhibition*, pages SPE–15564. SPE, 1986.

[143] Å Kyllingstad and GWJ Halsey. A study of slip/stick motion of the bit. *SPE drilling engineering*, 3(04):369–373, 1988.

[144] PC Kriesels, WJG Keultjes, P Dumont, Issam Huneidi, OO Owoeye, and RA Hartmann. Cost savings through an integrated approach to drillstring vibration control. In *SPE/IADC Middle East Drilling Technology Conference and Exhibition*, pages SPE–57555. SPE, 1999.

[145] Graham Mensa-Wilmot, Martin Booth, and Andrew Mottram. New pdc bit technology and improved operational practices saves $1 m in central north sea drilling program. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–59108. SPE, 2000.

[146] Ahmed A Elgibaly, Mohammed Shehata Farhat, Eric W Trant, and Mohammed Kelany. A study of friction factor model for directional wells. *Egyptian journal of petroleum*, 26 (2):489–504, 2017.

[147] Ulf Jakob F Aarsnes and Roman J Shor. Torsional vibrations with bit off bottom: Modeling, characterization and field data validation. *Journal of Petroleum Science and Engineering*, 163:712–721, 2018.

[148] Ulf Jakob F Aarsnes and Nathan van de Wouw. Axial and torsional self-excited vibrations of a distributed drill-string. *Journal of Sound and Vibration*, 444:127–151, 2019.

[149] Belem Saldivar, Sabine Mondié, Jean-Jacques Loiseau, and Vladimir Rasvan. Stick-slip oscillations in oillwell drilstrings: distributed parameter and neutral type retarded model approaches. *IFAC Proceedings Volumes*, 44(1):284–289, 2011.

[150] Carlos Canudas-de Wit, Francisco R Rubio, and Miguel Angel Corchero. D-oskil: A new mechanism for controlling stick-slip oscillations in oil well drillstrings. *IEEE Transactions on Control Systems Technology*, 16(6):1177–1191, 2008.

[151] Thomas Richard, Christophe Germay, and Emmanuel Detournay. Self-excited stick–slip oscillations of drill bits. *Comptes rendus MECANIQUE*, 332(8):619–626, 2004.

[152] C A Johancsik, D B Friesen, and Rapier Dawson. Torque and drag in directional wells-prediction and measurement. *Journal of Petroleum technology*, 36(06):987–992, 1984.

[153] John E McCormick, Melissa Frilot, and TzuFang Chiu. Torque and drag software model comparison: Impact on application and calibration of field data. In *SPE Brasil Offshore*, pages SPE–143623. SPE, 2011.

[154] Chengda Lu, Min Wu, Xin Chen, Weihua Cao, Chao Gan, and Jinhua She. Downhole-friction-estimation-based rotary speed control for drillstring system with stick-slip vibrations. In *2017 11th Asian Control Conference (ASCC)*, pages 2618–2623. IEEE, 2017.

[155] Ulf Jakob F Aarsnes, Jean Auriol, Florent Di Meglio, and Roman J Shor. Estimating friction factors while drilling. *Journal of Petroleum Science and Engineering*, 179:80–91, 2019.

[156] Jean Auriol, Ulf Jakob F Aarsnes, and Roman Shor. Self-tuning torsional drilling model for real-time applications. In *2020 American Control Conference (ACC)*, pages 3091–3096. IEEE, 2020.

[157] Florent Di Meglio, Pierre-Olivier Lamare, and Ulf Jakob F Aarsnes. Robust output feedback stabilization of an ode–pde–ode interconnection. *Automatica*, 119:109059, 2020.

[158] Andreas Raue, Marcel Schilling, Julie Bachmann, Andrew Matteson, Max Schelke, Daniel Kaschek, Sabine Hug, Clemens Kreutz, Brian D Harms, Fabian J Theis, et al. Lessons learned from quantitative dynamical modeling in systems biology. *PloS one*, 8 (9):e74335, 2013.

[159] Klaus Schittkowski. *Numerical data fitting in dynamical systems: a practical introduction with applications and software*, volume 77. Springer Science & Business Media, 2002.

[160] Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[161] Junbo Zhao, Antonio Gómez-Expósito, Marcos Netto, Lamine Mili, Ali Abur, Vladimir Terzija, Innocent Kamwa, Bikash Pal, Abhinav Kumar Singh, Junjian Qi, et al. Power system dynamic state estimation: Motivations, definitions, methodologies, and future work. *IEEE Transactions on Power Systems*, 34(4):3188–3198, 2019.

[162] Emmanuel D Blanchard, Adrian Sandu, and Corina Sandu. Parameter estimation method using an extended kalman filter. 2007.

[163] Laliberte Loron and G Laliberte. Application of the extended kalman filter to parameters estimation of induction motors. In *1993 Fifth European Conference on Power Electronics and Applications*, pages 85–90. IET, 1993.

[164] Amir Noabahar Sadeghi, Kutluk Bilge Arıkan, and Mehmet Efe Özbek. Torsional model of the drill string, and real-time prediction of the bit rotational speed and the torque on bit, in an oil well drilling tower. *Journal of Petroleum Science and Engineering*, 195: 107814, 2020.

[165] Jean Auriol, Roman Shor, Silviu-Iulian Niculescu, and Nasser Kazemi. Estimating drill string friction with model-based and data-driven methods. In *2022 American Control Conference (ACC)*, pages 3464–3469. IEEE, 2022.

[166] Giancarlo Bernasconi and Massimiliano Vassallo. Estimation of drill-bit rotation velocity with surface and downhole dynamics measurements. In *SEG International Exposition and Annual Meeting*, pages SEG–2001. SEG, 2001.

[167] Kursat Rasim Mestav, Jaime Luengo-Rozas, and Lang Tong. State estimation for unobservable distribution systems via deep neural networks. In *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[168] M Zhang, ME Kaya, N Habib, A Groh, A Small, Y Xue, and J Walthall. Real-time downhole rpm range prediction for improved stick-slip detection using ensemble machine learning. In *SPE/IADC Drilling Conference and Exhibition*, page D022S011R002. SPE, 2025.

[169] Johan Dirk Jansen and Leon Van Den Steen. Active damping of self-excited torsional vibrations in oil well drillstrings. *Journal of sound and vibration*, 179(4):647–668, 1995.

[170] TG Ritto, RR Aguiar, and S Hbaieb. Validation of a drill string dynamical model and torsional stability. *Meccanica*, 52:2959–2967, 2017.

[171] Thiago G Ritto and Maryam Ghandchi-Tehrani. Active control of stick-slip torsional vibrations in drill-strings. *Journal of Vibration and Control*, 25(1):194–202, 2019.

[172] Lucas P Volpi, Daniel M Lobo, and Thiago G Ritto. A stochastic analysis of the coupled lateral–torsional drill string vibration. *Nonlinear Dynamics*, 103:49–62, 2021.

[173] Eva Maria Navarro-Lopez and Rodolfo Suarez. Practical approach to modelling and controlling stick-slip oscillations in oilwell drillstrings. In *Proceedings of the 2004 IEEE International Conference on Control Applications, 2004.*, volume 2, pages 1454–1460. IEEE, 2004.

[174] Fereidoun Abbassian and VA Dunayevsky. Application of stability approach to torsional and lateral bit dynamics. *SPE Drilling & Completion*, 13(02):99–107, 1998.

[175] JF Brett. The genesis of torsional drillstring vibrations (spe 21943). *SPE Drilling Engineering*, 7(3):168–174, 1992.

[176] Abdessalem Belaid. *Modélisation tridimensionnelle du comportement mécanique de la garniture de forage dans les puits à trajectoires complexes: application à la prédiction des frottements garniture-puits*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2005.

[177] Ingvar Andersson. Stick-slip motion in one-dimensional continuous systems and in systems with several degrees of freedom. 1981.

[178] Leon van den Steen. Suppressing stick-slip-induced drillstring oscillations: a hyperstability approach. 1998.

[179] Fereidoun Abbassian and VA Dunayevsky. Application of stability approach to torsional and lateral bit dynamics. *SPE Drilling & Completion*, 13(02):99–107, 1998.

[180] N Challamel, H Sellami, E Chenevez, and L Gossuin. A stick-slip analysis based on rock/bit interaction: theoretical and experimental contribution. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–59230. SPE, 2000.

[181] Parimal Arjun Patil and Catalin Teodoriu. Model development of torsional drillstring and investigating parametrically the stick-slips influencing factors. *Journal of Energy Resources Technology*, 135(1):013103, 2013.

[182] Panagiotis Athanasiou and Yaser Hadi. Simulation of oil well drilling system using distributed–lumped modelling technique. *Modelling*, 1(2):175–197, 2020.

[183] Dou Xie, Zhiqiang Huang, Yachao Ma, Vahid Vaziri, Marcin Kapitaniak, and Marian Wiercigroch. Nonlinear dynamics of lump mass model of drill-string in horizontal well. *International Journal of Mechanical Sciences*, 174:105450, 2020.

[184] LC Cunha Lima, RR Aguiar, TG Ritto, and S Hbaieb. Analysis of the torsional stability of a simplified drillstring. In *Proceedings of the XVII international symposium on dynamic problems of mechanics*, 2015.

[185] Liping Tang, Baolin Guo, Xiaohua Zhu, Changshuai Shi, and Yunlai Zhou. Stick–slip vibrations in oil well drillstring: A review. *Journal of Low Frequency Noise, Vibration and Active Control*, 39(4):885–907, 2020.

[186] Liping Tang and Xiaohua Zhu. Effects of the difference between the static and the kinetic friction coefficients on a drill string vibration linear approach. *Arabian Journal for Science and Engineering*, 40:3723–3729, 2015.

[187] Liping Tang, Xiaohua Zhu, Changshuai Shi, and Jianqiang Sun. Investigation of the damping effect on stick-slip vibration of oil and gas drilling system. *J Vib Eng Technol*, 4(1):79–86, 2016.

[188] E Omojuwa, S Osisanya, and R Ahmed. Dynamic analysis of stick-slip motion of drillstring while drilling. In *SPE Nigeria Annual International Conference and Exhibition*, pages SPE–150755. SPE, 2011.

[189] Thomas Marc Richard. *Self-excited stick-slip oscillations of drag bits*. University of Minnesota, 2001.

[190] Mohammad Javad Moharrami, Clóvis de Arruda Martins, and Hodjat Shiri. Nonlinear integrated dynamic analysis of drill strings under stick-slip vibration. *Applied Ocean Research*, 108:102521, 2021.

[191] Jeong-Kweon Seo. A pretraining domain decomposition method using artificial neural networks to solve elliptic pde boundary value problems. *Scientific Reports*, 12(1):13939, 2022.

[192] Taco de Wolff, Hugo Carrillo, Luis Martí, and Nayat Sanchez-Pi. Assessing physics informed neural networks in ocean modelling and climate change applications. In *AI: Modeling Oceans and Climate Change Workshop at ICLR 2021*, 2021.

[193] Zheyuan Hu, Ameya D Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (xpinns) improve generalization? *arXiv preprint arXiv:2109.09444*, 2021.

[194] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):6136, 2021.

[195] Tim De Ryck, Ameya D Jagtap, and Siddhartha Mishra. Error estimates for physics-informed neural networks approximating the navier–stokes equations. *IMA Journal of Numerical Analysis*, 44(1):83–119, 2024.

[196] Yuyao Chen, Lu Lu, George Em Karniadakis, and Luca Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8):11618–11633, 2020.

[197] Ameya D Jagtap, Zhiping Mao, Nikolaus Adams, and George Em Karniadakis. Physics-informed neural networks for inverse problems in supersonic flows. *Journal of Computational Physics*, 466:111402, 2022.

[198] GP Purja Pun, R Batra, R Ramprasad, and Y Mishin. Physically informed artificial neural networks for atomistic modeling of materials. *Nature communications*, 10(1): 2339, 2019.

[199] Vikas Dwivedi and Balaji Srinivasan. A normal equation-based extreme learning machine for solving linear partial differential equations. *Journal of Computing and Information Science in Engineering*, 22(1):014502, 2022.

[200] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[201] Stephanie Yang, Hsueh-Chih Chen, Chih-Hsien Wu, Meng-Ni Wu, and Cheng-Hong Yang. Forecasting of the prevalence of dementia using the lstm neural network in taiwan. *Mathematics*, 9(5):488, 2021.

[202] Xu Liu, Xiaoya Zhang, Wei Peng, Weien Zhou, and Wen Yao. A novel meta-learning initialization method for physics-informed neural networks. *Neural Computing and Applications*, 34(17):14511–14534, 2022.

[203] Bin Huang and Jianhui Wang. Applications of physics-informed neural networks in power systems-a review. *IEEE Transactions on Power Systems*, 38(1):572–588, 2022.

[204] Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Solving the wave equation with physics-informed deep learning. *arXiv preprint arXiv:2006.11894*, 2020.

[205] Rongxi Gou, Yijie Zhang, Xueyu Zhu, and Jinghuai Gao. Bayesian physics-informed neural networks for the subsurface tomography based on the eikonal equation. *IEEE transactions on geoscience and remote sensing*, 61:1–12, 2023.

[206] Yin Yuan, Weiqing Li, Lihan Bian, and Junkai Lei. A prediction model for pressure and temperature in geothermal drilling based on physics-informed neural networks. *Electronics*, 13(19):3869, 2024.

[207] Zhaopeng Zhu, Zihao Liu, Xianzhi Song, Shuo Zhu, Mengmeng Zhou, Gensheng Li, Shiming Duan, Baodong Ma, Shanlin Ye, and Rui Zhang. A physics-constrained data-driven workflow for predicting bottom hole pressure using a hybrid model of artificial neural network and particle swarm optimization. *Geoenergy Science and Engineering*, 224:211625, 2023.

[208] Jeanne Redaud, Maxime Darrin, Nasser Kazemi, and Jean Auriol. Physics-informed dual architecture neural networks for enhanced estimation of drilling dynamics. In *IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium*, pages 1038–1042. IEEE, 2024.

[209] Zaharaddeen Karami Lawal, Hayati Yassin, Daphne Teck Ching Lai, and Azam Che Idris. Physics-informed neural network (pinn) evolution and beyond: A systematic literature review and bibliometric analysis. *Big Data and Cognitive Computing*, 6 (4):140, 2022.

[210] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

[211] Matthieu Barreau, Miguel Aguiar, John Liu, and Karl Henrik Johansson. Physics-informed learning for identification and state reconstruction of traffic density. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2653–2658. IEEE, 2021.

[212] Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism. *arXiv preprint arXiv:2009.04544*, 2020.

[213] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18(153):1–43, 2018.

[214] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[215] Barak A Pearlmutter and Jeffrey Mark Siskind. Reverse-mode ad in a functional framework: Lambda the ultimate backpropagator. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 30(2):1–36, 2008.

[216] Wonyeol Lee, Hangyeol Yu, Xavier Rival, and Hongseok Yang. On correctness of automatic differentiation for non-differentiable functions. *Advances in neural information processing systems*, 33:6719–6730, 2020.

[217] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

[218] Zhiwei Fang. A high-efficient hybrid physics-informed neural networks based on convolutional neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5514–5526, 2021.

[219] Jiaqi Hua, Yingguang Li, Changqing Liu, Peng Wan, and Xu Liu. Physics-informed neural networks with weighted losses by uncertainty evaluation for accurate and stable prediction of manufacturing systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[220] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115 (34):8505–8510, 2018.

[221] Chen Cheng, Peng-Fei Xu, Yong-Zheng Li, and Guang-Tao Zhang. Deep learning based on pinn for solving 2 d0f vortex induced vibration of cylinder with high reynolds number. *arXiv preprint arXiv:2106.01545*, 2021.

[222] Akshansh Mishra. Transport equation based physics informed neural network to predict the yield strength of architected materials. *arXiv preprint arXiv:2312.00003*, 2023.

[223] Kamaljyoti Nath, Xuhui Meng, Daniel J Smith, and George Em Karniadakis. Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines. *Scientific Reports*, 13(1):13683, 2023.

[224] Chenhao Si, Ming Yan, Xin Li, and Zhihong Xia. Complex physics-informed neural network. *arXiv preprint arXiv:2502.04917*, 2025.

[225] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[226] Aleks Gollu and Pravin Varaiya. Hybrid dynamical systems. In *Proceedings of the 28th IEEE Conference on Decision and Control,*, pages 2708–2712. IEEE, 1989.

[227] Allen Back, John Guckenheimer, and Mark Myers. A dynamical simulation facility for hybrid systems. In *International Hybrid Systems Workshop*, pages 255–267. Springer, 1991.

[228] Roger W Brockett. Hybrid models for motion control systems. In *Essays on Control: Perspectives in the Theory and its Applications*, pages 29–53. Springer, 1993.

[229] Pravin Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on automatic control*, 38(2):195–207, 2002.

[230] Pablo A Iglesias. On the stability of sampled-data linear time-varying feedback systems. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 1, pages 219–224. IEEE, 1994.

[231] Michael S Branicky. Stability of switched and hybrid systems. In *Proceedings of 1994 33rd IEEE conference on decision and control*, volume 4, pages 3498–3503. IEEE, 1994.

[232] Elena Bossolini, Morten Brøns, and Kristian Uldall Kristiansen. Canards in stiction: on solutions of a friction oscillator by regularization. 16(4):2233–2258, 2017.

[233] Mohammad Ziaee and Farzad Hejazi. Developed steady-state response for a new hybrid damper mounted on structures. *Plos one*, 18(8):e0290248, 2023.

[234] Arjan J Van Der Schaft and Hans Schumacher. *An introduction to hybrid dynamical systems*, volume 251. springer, 2007.

[235] G Davrazos and NT Koussoulas. A review of stability results for switched and hybrid systems. In *Mediterranean Conference on Control and Automation*. Citeseer, 2001.

[236] Fayssal Arichi, Jeremy Van Gorp, Mohamed Djemai, Michael Defoort, and Brahim Cherki. Hybrid state estimation for a class of switched system using petri nets. In *2014 European Control Conference (ECC)*, pages 2534–2539. IEEE, 2014.

[237] Robert Shorten, Fabian Wirth, Oliver Mason, Kai Wulff, and Christopher King. Stability criteria for switched and hybrid systems. *SIAM review*, 49(4):545–592, 2007.

[238] Fulvio Forni, Andrew R. Teel, and Luca Zaccarian. Follow the bouncing ball: Global results on tracking and state estimation with impacts. *IEEE Transactions on Automatic Control*, 58(6):1470–1485, 2013. doi: 10.1109/TAC.2013.2237952.

[239] L. Menini and A. Tornambè. Velocity observers for linear mechanical systems subject to single non-smooth impacts. *Systems Control Letters*, 43(3):193–202, 2001. ISSN 0167-6911. doi: https://doi.org/10.1016/S0167-6911(01)00097-4. URL https://www.sciencedirect.com/science/article/pii/S0167691101000974.

[240] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE control systems magazine*, 29(2):28–93, 2009.

[241] Pauline Bernard and Ricardo G. Sanfelice. Semiglobal high-gain hybrid observer for a class of hybrid dynamical systems with unknown jump times. *IEEE Transactions on Automatic Control*, 69(9):5804–5819, 2024. doi: 10.1109/TAC.2024.3355324.

[242] Gia Quoc Bao Tran, Sergio Garcia, Pauline Bernard, Ricardo G Sanfelice, and Florent Di Meglio. Towards gluing kkl observer for hybrid systems with unknown jump times (full version). In *63rd IEEE Conference on Decision and Control*, 2024.

[243] Gia Quoc Bao Tran. *Observer Design for Hybrid Systems*. Theses, Mines Paris - PSL, November 2024. URL https://hal.science/tel-04975149.

[244] Michael Bloesch, Marco Hutter, Mark A Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, and Roland Siegwart. State estimation for legged robots: Consistent fusion of leg kinematics and imu. 2013.

[245] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4):402–430, 2020.

[246] Yuan Gao, Chengzhi Yuan, and Yan Gu. Invariant extended kalman filtering for hybrid models of bipedal robot walking. *IFAC-PapersOnLine*, 54(20):290–297, 2021.

[247] J Joe Payne, Nathan J Kong, and Aaron M Johnson. The uncertainty aware salted kalman filter: State estimation for hybrid systems with uncertain guards. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8821–8828. IEEE, 2022.

[248] Wu Wen and Hugh F Durrant-Whyte. Model based active object localisation using multiple sensors. In *Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91*, pages 1448–1452. IEEE, 1991.

[249] Wu Wen and Hugh F Durrant-Whyte. Model-based multi-sensor data fusion. In *Proceedings 1992 IEEE international conference on robotics and automation*, pages 1720–1726. IEEE, 1992.

[250] Raphael Hauser. Kalman filtering with equality and inequality state constraints. Technical report, 2007.

[251] Dan Simon and Tien Li Chia. Kalman filtering with state equality constraints. *IEEE transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

[252] Nachi Gupta. Kalman filtering in the presence of state space equality constraints. In *2007 Chinese Control Conference*, pages 107–113. IEEE, 2007.

[253] Dan Simon and Donald L Simon. Aircraft turbofan engine health estimation using constrained kalman filtering. *J. Eng. Gas Turbines Power*, 127(2):323–328, 2005.

[254] Nathan J Kong, J Joe Payne, James Zhu, and Aaron M Johnson. Saltation matrices: The essential tool for linearizing hybrid dynamical systems. *Proceedings of the IEEE*, 2024.

[255] Simon J Julier and Joseph J LaViola. On kalman filtering with nonlinear equality constraints. *IEEE transactions on signal processing*, 55(6):2774–2784, 2007.

[256] Dan Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.

[257] C Yung and E Blasch. Kalman filtering with nonlinear state contraints. In *Proc. FUSION'06: 9th Int. Conf. Inform. Fusion*, pages 10–14, 2006.

[258] Nancy L Nihan and Gary A Davis. Recursive estimation of origin-destination matrices from input/output counts. *Transportation Research Part B: Methodological*, 21(2):149–163, 1987.

[259] AH Mohamed and KP Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.

[260] Ali Almagbile, Jinling Wang, and Weidong Ding. Evaluating the performances of adaptive kalman filter methods in gps/ins integration. *Journal of Global Positioning Systems*, 9(1):33–40, 2010.

[261] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. In *2017 IEEE power & energy society general meeting*, pages 1–5. IEEE, 2017.

[262] Jinling Wang. Stochastic modeling for real-time kinematic gps/glonass positioning. *Navigation*, 46(4):297–305, 1999.

[263] András Retzler, Jan Swevers, Joris Gillis, and Zsolt Kollár. Shooting methods for identification of nonlinear state-space grey-box models. In *2022 IEEE 17th International Conference on Advanced Motion Control (AMC)*, pages 207–212. IEEE, 2022.

[264] Marah Engels, Gerwald Lichtenberg, and Steffi Knorn. An approach to parameter identification for boolean-structured multilinear time-invariant models. In *2024 European Control Conference (ECC)*, pages 2077–2082. IEEE, 2024.

[265] Elin Dahlborg, Per Norrlund, and Linn Saarinen. Kaplan turbine model validation for large grid frequency disturbances. *IEEE Transactions on Energy Conversion*, 36(2): 611–618, 2020.

[266] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9 (1):2, 2020.

[267] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[268] Adam C Neufeldt, Stephen W Lai, and Sean D Kristjansson. An algorithm to automatically zero weight on bit and differential pressure and resulting improvements in data quality. *SPE Drilling & Completion*, 35(04):535–552, 2020.

[269] Arghad Arnaout, R Fruhwirth, Bilal Esmael, and Gerhard Thonhauser. Intelligent real-time drilling operations classification using trend analysis of drilling rig sensors data. In *SPE Kuwait International Petroleum Conference and Exhibition*, pages SPE–163302. SPE, 2012.

[270] Adriane BS Serapiao, Rogerio M Tavares, Jose Ricardo P Mendes, and Ivan R Guilherme. Classification of petroleum well drilling operations using support vector machine (svm). In *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, pages 145–145. IEEE, 2006.

[271] MX Lee and Z Wang. Retrieving operation insights with genai llm: Comparative analysis and workflow enhancement. In *Abu Dhabi International Petroleum Exhibition and Conference*, page D011S020R004. SPE, 2024.

[272] Lucas P Cinelli, José FL de Oliveira, Vinicius M de Pinho, Wesley L Passos, Rafael Padilla, Patrick F Braz, Breno Galves, Domenica P Dalvi, Gabriela Lewenfus, Jonathas O Ferreira, et al. Automatic event identification and extraction from daily drilling reports using an expert system and artificial intelligence. *Journal of Petroleum Science and Engineering*, 205:108939, 2021.

[273] Zhiwei Ma, Javier E Santos, Greg Lackey, Hari Viswanathan, and Daniel O'Malley. Information extraction from historical well records using a large language model. *Scientific Reports*, 14(1):31702, 2024.

[274] Yunfei Zhang, Zhihua Zhou, Junwei Liu, and Jianjuan Yuan. Data augmentation for improving heating load prediction of heating substation based on timegan. *Energy*, 260: 124919, 2022.

[275] Diaa Salman, Cem Direkoglu, Mehmet Kusaf, and Murat Fahrioglu. Hybrid deep learning models for time series forecasting of solar power. *Neural Computing and Applications*, 36(16):9095–9112, 2024.

[276] Mujie Sui, Cheng Zhang, Li Zhou, Shuhan Liao, and Changsong Wei. An ensemble approach to stock price prediction using deep learning and time series models. In *2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 793–797. IEEE, 2024.

[277] Sean Xianping Wu, Luis Paez, Uyen Partin, and Mukul Agnihotri. Decoupling stick-slip and whirl to achieve breakthrough in drilling performance. In *SPE/IADC Drilling Conference and Exhibition*, pages SPE–128767. SPE, 2010.

[278] Patrick T Harker and Jong-Shi Pang. Existence of optimal solutions to mathematical programs with equilibrium constraints. *Operations research letters*, 7(2):61–64, 1988.

SI metric conversion factors

Imperial	**Value**	**SI Unit**
inch	0.025 400	m
ft	0.304 800	m
lb	0.453 592	kg
lbf	4.448 222	N
psi (lbf/in^2)	6894.757 000	Pa
lbf.ft	1.355 820	$\mathrm{N \cdot m}$
rpm	0.104 720	$\mathrm{rad \cdot s^{-1}}$

RÉSUMÉ

Cette thèse étudie des méthodes permettant d'estimer la sévérité des vibrations de type "stick-slip" au fond du puits lors du forage, en utilisant comme entrée des mesures de surface échantillonnées à basse fréquence. Deux approches sont proposées : la première vise à prédire le "Stick-Slip Index (SSI)", un indicateur couramment utilisé dans l'industrie pour évaluer la sévérité des vibrations torsionnelles, directement à partir des mesures de surface. La seconde consiste à reconstruire la variation de la vitesse de rotation du trépan à partir de ces mêmes mesures. Une fois cette variation obtenue, elle permet d'inférer la sévérité des vibrations torsionnelles au fond du puits. En conséquence, la thèse est structurée en deux grandes parties, chacune étant consacrée à l'une de ces approches.
Dans la première partie, un modèle d'apprentissage automatique est utilisé pour prédire le SSI à partir de séquences temporelles de mesures de surface. L'objectif est d'entraîner un modèle capable de bien généraliser sur différents puits présentant des conditions géologiques variées. À cette fin, plusieurs stratégies sont explorées, notamment la normalisation des données, l'intégration de caractéristiques dérivées de lois physiques, l'application de l'apprentissage par transfert ou "transfer learning (TL)", ainsi que l'augmentation de données afin d'obtenir un jeu de données plus équilibré. L'influence de l'architecture du modèle est étudiée en comparant les réseaux récurrents à mémoire court et long terme ou "Long Short Term Memory (LSTM)" et les modèles de type "Transformer". En outre, des stratégies avancées de généralisation, telles que la généralisation de domaine adversariale ou "Adversarial Domain Generalization (ADG)" et la minimisation du risque invariant ou "Invariant Risk Minimization (IRM)", sont employées afin d'améliorer la robustesse face aux changements de domaine et à la variabilité opérationnelle.

Dans la seconde partie, deux méthodes hybrides sont étudiées pour prédire la vitesse de rotation du trépan et les coefficients de frottement entre la tige de forage et le puits, à partir de mesures de surface échantillonnées à 1 Hz et d'un modèle dynamique torsionnel non linéaire simplifié. La première méthode repose sur un réseau de neurones informé par la physique ou "Physics-Informed Neural Network (PINN)", une approche hors ligne utilisée pour résoudre le problème d'identification du système, le modèle physique non linéaire étant intégré dans la fonction de perte. La seconde méthode est une approche en ligne, issue de la théorie du contrôle, basée sur une version améliorée du filtre de "Salted Kalman Filter (SKF)" adaptée aux systèmes hybrides non linéaires. Dans les deux cas, les méthodes sont évaluées sur des données simulées et réelles afin d'en évaluer l'applicabilité pratique et la robustesse.

MOTS CLÉS

Stick-slip, Généralisation en apprentissage automatique, Séries temporelles, Réseau de neurones informé par la physique, Apprentissage par transfert, Identification de systèmes, Filtre de Kalman, Système hybride, Dynamiques non lisses.

ABSTRACT

This thesis investigates methods for estimating the severity of downhole stick-slip vibrations during drilling, using surface measurements with a low-frequency sample rate as input. Two approaches are proposed: The first focuses on predicting the Stick-Slip Index (SSI), an industry-standard indicator for assessing the severity of torsional vibrations, directly from surface measurements. The second seeks to reconstruct the variation of the bit rotation speed from surface measurements. Once this variation is obtained, it provides information about the severity of downhole torsional vibrations. Accordingly, the thesis is structured into two main parts, each dedicated to one of these approaches.
In the first part, a data-based machine learning model is used to predict the SSI from time-series sequences of surface measurements. The objective is to train a model that generalizes well across different wells with varying geological conditions. To this end, several strategies are explored, including data normalization, the incorporation of engineered features derived from physical laws, the application of transfer learning, and the augmentation of data to achieve a more balanced dataset. The influence of model architecture is examined through comparisons between Long Short-Term Memory (LSTM) networks and Transformer-based models. Furthermore, advanced generalization strategies, such as Adversarial Domain Generalization (ADG) and Invariant Risk Minimization (IRM), are employed to enhance robustness to domain shifts and operational variability.

In the second part, two hybrid-based methods are investigated to predict the bit rotation speed and the friction coefficients between the drill string and the borehole using 1-Hz surface measurements and a simplified nonlinear torsional dynamic model. The first method relies on a Physics-Informed Neural Network (PINN), an offline approach used to solve the system identification problem with the nonlinear physical model incorporated into the loss function. The second method is an online, control-theoretic approach based on an enhanced version of the Salted Kalman Filter (SKF) adapted to hybrid nonlinear systems. In both cases, the methods are evaluated on simulated and field data to assess their practical applicability and robustness.

KEYWORDS

Stick-slip, Machine learning generalization, Time series, Physics-Informed Neural Network, Transfer learning, System identification, Kalman filter, Hybrid system, Non-smooth dynamics.