

RAPPORT DE STAGE

DÉVELOPPEMENT D'UN SYSTÈME DE SUIVI DU TRAFIC PAR PONT INSTRUMENTÉ

Tutrice IFSTTAR : Franziska SCHMIDT

Tuteurs ESIEE : Jean-François BERCHER, Giovanni CHIERCHIA

Table des matières

Résumé de stage	2
Français	2
Anglais	2
Descriptif de l'entreprise	3
Objectifs et contexte de la mission	4
Travail réalisé et résultats obtenus	5
Introduction et environnement de travail	5
Chargement de la data	7
Calibration	8
Calcul de la ligne d'influence à partir de plusieurs camions	8
Pondération des signaux issus des camions par niveau de bruit	11
Régularisation appliquée à la calibration pour plusieurs camions	13
Normalisation de l'échelle des signaux	15
Calibration avec tri sur la vitesse des camions	17
Facteurs d'amplitude sur la ligne d'influence selon la vitesse des camions	18
Optimisation de la calibration	20
Conservation de l'énergie	24
Interpolation de la ligne d'influence selon les vitesses des camions	25
Pesée en marche (PEM)	30
Première estimation des poids	30
Décalage des pics sur les signaux	34
Procédure d'évaluation des statistiques sur nos méthodes	38
Evaluation des résultats sur des données simulées	47
Détermination des solutions de l'équation dynamique du pont par <i>physic informed neural network</i> (PINN)	51
Principe	51
Résolution	51
Bilan d'expérience technique, organisationnel et humain	55
Annexes	56
Annexe 1 - Bibliographie	56

Résumé de stage

Français

Le pesage en marche par ponts instrumentés consiste à enregistrer les signaux de déformation ou de contrainte lors de passage de camions par des capteurs. Il est possible d'utiliser le modèle physique B-WIM (Bridge Weighing In Motion) pour estimer la ligne d'influence du pont puis leurs poids. Tel était le premier objectif du stage.

Plus précisément, il fallait estimer au mieux la ligne d'influence du pont à l'aide des camions de calibration puis l'utiliser pour estimer les poids des camions de trafic.

Ces calculs sont cependant sujet à incertitudes à cause de divers paramètres, liés à l'ouvrage ou aux mesures.

Une fois les résultats obtenus satisfaisants, le but était de retrouver les solutions de l'équation dynamique du pont, décrivant la ligne d'influence lorsqu'elles sont évaluées en son milieu, par physic based deep learning.

Diverses campagnes de mesure de pesage par ponts instrumentés ont été menées, avec enregistrement des signaux temporels et des données concernant les camions.

Nous avons exclusivement travaillé sur les données du pont de Senlis.

Anglais

Weighing in motion by instrumented bridges consists of recording strain or stress signals when trucks pass through load cells. It is possible to use the physical model BWIM (Bridge- Weighing In Motion) to estimate the line of influence of the bridge and trucks weights. This was the first objective of the course.

More precisely, it was necessary to best estimate the line of influence of the bridge using the calibration trucks and then use it to estimate the weights of the traffic trucks.

However, these calculations are subject to uncertainties due to various parameters related to the structure or the measurements.

Once the results obtained were satisfactory, the goal was to find the solutions to the dynamic equation of the bridge, describing the line of influence when evaluated in its middle (of the bridge), by physics-based deep learning.

Various instrumented bridge weighing measurement campaigns were carried out, with recording of time signals and truck data.

We worked exclusively on the data from the Senlis bridge.

Descriptif de l'entreprise

L'IFSTTAR (Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux) a été créé en 2011 après la fusion du LCPC (Laboratoire Central des Ponts et Chaussées) et de l'INRETS (Institut National de Recherche sur les Transport et leur Sécurité). Cet établissement public est géré conjointement par le Ministère de l'Environnement, de l'Énergie et de la Mer et le Ministère de l'enseignement supérieur et de la recherche.

L'institut est composé de six antennes réparties en France (Lille, Marne-la-vallée, Versailles, Nantes, Lyon, Marseille). Son siège social se situe à Marne-la-Vallée.

L'institut se décompose en cinq départements :

- MAST : Matériaux et structures
- GERS : Géotechnique, environnement, risques naturels et sciences de la terre
- COSYS : Composants et systèmes
- TS2 : Transport, santé et sécurité
- AME : Aménagement, mobilités et environnement

Acteur majeur de la recherche européenne, l'IFSTTAR mène des travaux de recherche et d'expertise au service des ministères, administrations, ainsi qu'auprès d'entreprises telles que EDF ou Eiffage.

Les chercheurs répondent aussi aux appels à projets de l'ANR (Agence Nationale de la Recherche), de l'ADEME, de Horizon 2020, programme européen pour la recherche et l'innovation...

Parallèlement, l'IFSTTAR a à cœur la transmission du savoir scientifique. Aussi il participe activement à la formation à la recherche, initiale et continue en encadrant des subventions CIFRE aux doctorants et des activités d'enseignements avec des écoles d'ingénieurs et universités partenaires (ENTPE, ESIEE, ESTP, UPEM...).

Certaines missions sont aussi un moyen de pallier les difficultés de financement de la recherche. En effet, seuls les salaires des employés et l'entretien des locaux sont désormais subventionnés par l'Etat.

L'institut oriente ses recherches vers les grands défis sociétaux tels que le changement climatique et l'amélioration de la santé publique. Ainsi, il met à profit ses compétences en génie civil, mobilité et sécurité routière afin de penser et aménager des territoires durables et promouvoir une mobilité éco-responsable.

L'objectif est ainsi de promouvoir le développement des sciences et des technologies dans des domaines variés tels que le génie urbain, le génie civil, les infrastructures mais aussi la mobilité des personnes et des biens.

L'EMGCU (Expérimentation et Modélisation pour le Génie Civil et Urbain) a été créé en 2018 en fusionnant les laboratoires EMMS (Expérimentation et Modélisation des Matériaux et des Structures) et SDOA (Sécurité et Durabilité des Ouvrages d'Art). Il fait partie du département MAST et se spécialise dans l'expérimentation et la modélisation de toute construction en dehors des bâtiments d'habitation classiques. Le laboratoire conduit des recherches et expertises sur des thématiques centrées autour du cycle de vie et de la sécurité des infrastructures.

Objectifs et contexte de la mission

Dans un contexte de constante amélioration des technologies et d'une convention avec la DGITM (Direction Générale des Infrastructures, des Territoires et la Mer) sur le contrôle de sanction automatisé de surcharges, l'IFSTTAR a lancé un projet visant à contrôler les charges sur les essieux des poids-lourds (PL). Diverses technologies existantes sont alors implémentées, testées, comparées, dont celle qui consiste à utiliser les ponts comme balance pour peser le trafic : B-WIM (Bridge-Weighing In Motion), c'est-à-dire, que le pesage est effectué en marche par pont instrumenté.

C'est ce modèle qui a été utilisé dans la première partie du stage pour l'estimation des lignes d'influences puis des poids. Il existe déjà aujourd'hui des systèmes permettant la surveillance et l'enregistrement des charges sur essieux en marche en chaussées, en installant des capteurs en chaussée. Néanmoins, le B-WIM se présente comme une bonne alternative aux autres systèmes car il est transportable, facile à poser, nécessite peu de moyens humains et son installation est plus sûre et non intrusive.

Le principe est simple, afin de déterminer les charges sur les essieux des poids lourds (PL) en marche, une mesure de la déformation de l'ouvrage est effectuée en fonction du temps avec des capteurs de déformation. Le pont fait alors office de balance. (Voir Fig 1)

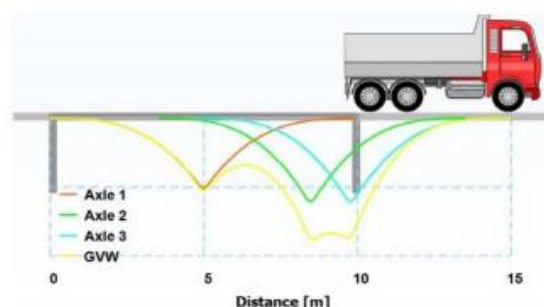


Fig.1 - Représentation des déformations en B-WIM

Cela impose une calibration préliminaire du système avec des PL dont les caractéristiques de poids et de longueur sont connues, permettant dans un premier temps de déterminer la fonction de transfert entre charges appliquées et déformations mesurées (appelée ligne d'influence en résistance des matériaux).

Les charges sur les essieux en marche sont finalement calculées par calcul inverse via la fonction de transfert précédemment calculée.

L'entreprise qui commercialise le système, fournit également un logiciel permettant la visualisation du calcul des charges sur chacun des essieux et indique les PL comme étant en surcharge ou non.








Cependant, d'après les études de l'IFSTTAR, les résultats de ce calcul inverse sont non conformes aux exigences de la DGITM (moins de 5% d'erreur entre poids réel et poids calculé, pour les essieux, les groupes d'essieux et le camion entier).

Le premier objectif est donc de passer sous ce seuil d'erreur. Lorsque les résultats seront satisfaisants, le deep learning sera utilisé pour calculer les solutions de l'équation dynamique du pont (équation différentielle) afin d'évaluer la ligne d'influence théorique ou, à l'inverse, trouver des caractéristiques du pont.

Travail réalisé et résultats obtenus

Introduction et environnement de travail

Pour atteindre les objectifs énoncés dans la partie précédente, le travail a été réalisé en **python** sur une distribution **Linux Ubuntu 20.04** via des **jupyter notebooks (.ipynb)**. De nombreuses bibliothèques ont été utilisées, les plus importantes étant :

- **NumPy**, pour l'utilisation de tableaux multidimensionnels  NumPy
- **SciPy**, pour l'implémentation de fonctions mathématiques complexes (interpolation, minimisation notamment)  SciPy
- **matplotlib** et **seaborn**, pour les affichages de courbes/graphes  matplotlib **Seaborn**
- **Scikit-learn**, pour les algorithmes de machine learning  learn
- **DeepXDE** et **SCIANN** (basés sur **Tensorflow**), pour la partie PINN  TensorFlow
- **GitLab** et **Git** : pour versionner le code et partager les ressources  git  GitLab
- les **fonctions** et **classes** fournies au début du stage

Les fichiers et données à disposition sont :

- **1_intro.ipynb** : Notebook décrivant succinctement le modèle B-WIM.
- **2_senlis.ipynb** : Notebook présentant les fonctions **python** permettant de lire les données, effectuer la calibration et l'estimation des poids.
- **4_calibration.ipynb** : Notebook contenant différentes manières d'estimer la ligne d'influence.

- **data** : Dossier racine contenant les signaux des poids lourds enregistrés par les capteurs.
- **data/senlis/Calibration - Description.txt** : Fichier contenant les informations des camions de calibration.
- **script** : Dossier contenant le script permettant de convertir le format .events en .CSV.
- **bwim.py** : script contenant des fonctions utiles à la calibration et la pesée en marche
- **plot.py** : script contenant les fonctions d'affichage des signaux et de la data.
- **data.py** : script contenant la fonction permettant de lire la data.
- **utils.py** : script contenant des fonctions utiles.

Les fichiers mis à notre disposition sont les :

- **.nswd** : (format propriétaire, résultats correspondant aux résultats rendus par le système de calcul de poids)
- **.event** : (format propriétaire, résultats correspondant aux signaux de déformation des différents PL)

Le stage se déroulera ainsi (Fig.2) :



Fig.2 - Déroulement du stage

Chargement de la data

Dans un premier temps, il fallait pouvoir utiliser l'ensemble données concernant les camions à notre disposition :

- Ceux du dossier de calibration
- Ceux du dossier de trafic

Pour cela, la fonction existante permettant de charger les poids-lourds (PL) de calibration a été modifiée pour qu'elle charge également ceux de trafic.

Il faut noter que les données brutes sont au format .event (format propriétaire, résultats correspondant aux signaux de déformation des différents PL), nécessitant de préciser sous forme de dictionnaire les poids des essieux. (Voir Fig.3). Il a donc fallu implémenter ce dictionnaire à l'aide d'un fichier texte précisant les poids des essieux des PL de trafic.

```
senlis weights modified = {  
    ###Trafic###  
    'PL1 2015-10-01-09-45-52-576': np.array([7.35, 11.25, 8.20, 8.10, 8.25]),  
    'PL2 2015-10-01-09-55-36-371': np.array([7.65, 14.10, 6.05, 6.20, 6.30]),  
    'PL3 2015-10-01-10-22-34-654': np.array([7.40, 11.60, 8.35, 8.50, 8.50]),  
    'PL4 2015-10-01-10-35-43-279': np.array([7.60, 12.35, 7.80, 7.70, 7.80]),  
    'PL5 2015-10-01-10-59-39-060': np.array([7.35, 10.50, 9.30, 9.70, 9.45]),  
    'PL6 2015-10-01-11-05-36-540': np.array([7.30, 14.30, 6.00, 6.60, 6.70]),  
    'PL7 2015-10-01-11-14-06-712': np.array([7.05, 13.95, 7.25, 7.55, 7.90]),  
    'PL8 2015-10-01-11-42-22-222': np.array([6.40, 14.05, 6.85, 6.75, 6.65]),  
    'PL9 2015-10-01-14-42-22-908': np.array([8.20, 12.00, 8.35, 8.40, 8.35]),  
    'PL10 2015-10-01-14-58-50-212': np.array([6.75, 12.45, 8.00, 8.10, 8.00]),  
    'PL11 2015-10-01-15-00-57-020': np.array([7.15, 11.40, 8.05, 8.00, 8.45]),  
    'PL12 2015-10-01-15-10-30-906': np.array([7.40, 10.80, 8.80, 9.05, 8.75]),  
    'PL13 2015-10-01-15-19-48-718': np.array([8.15, 12.20, 7.30, 8.30, 8.80]),  
    'PL14 2015-10-01-15-34-03-982': np.array([6.20, 5.05, 7.95, 7.65, 7.40, 7.65]),  
    'PL15 2015-10-01-15-36-13-519': np.array([5.35, 8.25, 8.10, 7.75, 7.80, 7.60]),  
    ###CALIBRATION###  
    '2015-07-02-10-54-53-222': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-07-02-16-36-41-533': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-09-29-10-28-52-687': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-09-56-08-457': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-09-28-46-125': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-15-52-01-990': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-07-02-17-05-11-125': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-07-02-16-08-40-480': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-07-02-15-40-32-876': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-07-02-11-54-10-861': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-07-02-11-24-46-054': np.array([6.03315, 10.8891, 9.17235, 8.58375, 8.77995]), #semi_trailor  
    '2015-09-29-15-23-14-921': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-14-55-32-562': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-14-28-15-093': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-14-08-38-369': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-12-17-33-320': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-13-33-05-960': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-11-50-31-328': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
    '2015-09-29-11-08-45-998': np.array([6.16068, 10.8499, 9.08406, 6.98472, 7.0632]), #cal_sep_2015  
}
```

Fig.3 - Définition des poids des essieux

Les PL de calibration ont chacun le même nombre d'essieux et le même poids total.

Les poids des PL de trafic ne sont pas censés être connus. Ils le sont dans ce cas, ce qui permettra par la suite de les utiliser pour la calibration si nous manquons de données de calibration.

Les PL sont stockés sous forme de **namedTuple** "truck", contenant les attributs de chaque PL (cf Annexe 2 et 3) dans des tableaux **NumPy**.

Calibration

Calcul de la ligne d'influence à partir de plusieurs camions

Une fois les données utilisables dans un script python, la calibration se présentait comme la première tâche à réaliser.

La calibration est définie comme le calcul de la ligne d'influence du pont à partir des données mesurées lors du passage des camions. Plus, précisément, nous utilisons le modèle physique B-WIM (Bridge-weighing-in-motion).

Dans les fonctions fournies existait un ensemble de fonctions permettant la calibration à partir d'un unique PL. (Voir Fig.4)

```
def calibration(truck, length, l2_reg=None, tv_reg=None):# pour le moment aucune régularization
    A, b = prepare_least_squares(truck, length)#retourne T et y
    A, b = prepare_regularization(A, b, l2_reg, tv_reg)#Aucune régularization pour le moment
    influence, _, _ = np.linalg.lstsq(A, b, rcond=None)# Retourne la solution des moindres carrés de ||y - Th|| avec h notre
    ligne d'influence (donc donne h)
    influence_bundle = time_to_meter_interpolation(truck, influence)#
    return influence_bundle
```

Fig.4 - Fonction de calibration pour un camion

Bien que théoriquement la ligne d'influence soit unique en chaque point du pont, différents facteurs (distance latérale de passage du PL par rapport au capteur, structure imparfaite du pont, usure du pont, précisions des outils de mesure etc..) altèrent les résultats obtenus. La calibration à partir d'un PL ne se généralise donc pas bien. (Voir Fig.4 et 5)

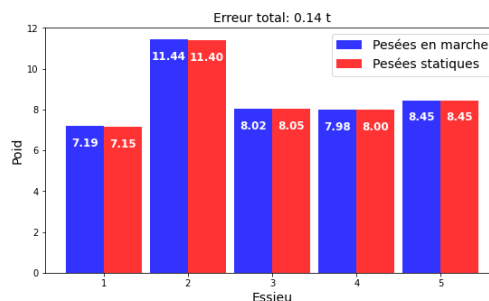


Fig.4 - Résultat de la pesée en marche (PEM) sur le camion de calibration

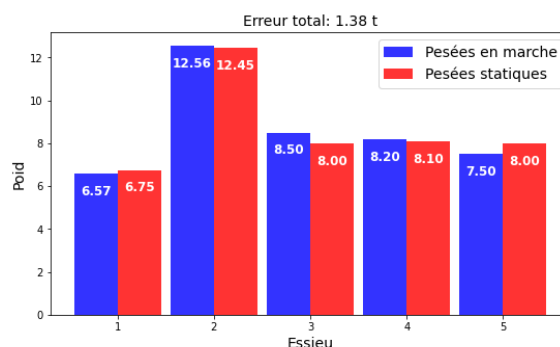


Fig.5 - Résultat de la PEM sur un autre camion que celui de calibration

Les résultats peuvent d'autant plus varier que les PL de trafic sont susceptibles d'avoir plus d'essieux que ceux de calibration ou un poids total différent.

Le problème majeur réside ainsi dans la généralisation de nos résultats. Cette généralisation est essentielle pour la mise en œuvre réelle de ces méthodes et l'atteinte de l'objectif final (moins de 5% d'erreur de précision).

Ce constat nous a amené à estimer la ligne d'influence à partir de plusieurs PL.

Mathématiquement, cela revient à passer de la minimisation pour un camion (voir Fig.6) à la minimisation pour plusieurs camions (voir Fig.7).

$$\underset{\mathbf{h} \in \mathbb{R}^M}{\text{minimize}} \quad \|\mathbf{T}\mathbf{h} - \mathbf{y}\|^2,$$

Fig.6 - Calcul de la ligne d'influence pour un camion

Avec :

- $\mathbf{T} = \sum_{a=1}^A w_a \mathbf{D}_a$.
- \mathbf{h} : ligne d'influence
- \mathbf{w}_a : poids par essieu
- \mathbf{D}_a : matrice de Toeplitz

$$\underset{\mathbf{h} \in \mathbb{R}^M}{\text{minimize}} \quad \|\tilde{\mathbf{T}}\mathbf{h} - \tilde{\mathbf{y}}\|^2$$

Fig.7 - Calcul de la ligne d'influence pour plusieurs camions

Avec :

$$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{T}_1 \\ \dots \\ \mathbf{T}_S \end{bmatrix} \quad \tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_1 \\ \dots \\ \mathbf{y}_S \end{bmatrix}$$

L'implémentation d'une fonction permettant cela se compose de deux parties :

- Modification de la fonction retournant le signal (y) et la matrice T (telle que décrite précédemment) en une fonction retournant une liste de signaux (y tilde) et la liste des T correspondants (T tilde).
- Modification de la fonction effectuant la minimisation

Initialement, la calibration utilisait la méthode des moindres carrés grâce à une fonction de la bibliothèque **NumPy** : `numpy.linalg.lstsq()`

Elle n'est cependant pas utilisable avec des matrices ou des tableaux multidimensionnels dont la dimension est strictement supérieure à 1.

Avec **SciPy**, le paramétrage de la minimisation est bien plus intéressant pour ce cas. Avec le bon algorithme de minimisation, il sera par la suite possible d'ajouter des contraintes ou des limites par exemple. L'intérêt principal est de pouvoir minimiser une fonction (qui retournera donc notre somme) à partir de ses paramètres (fixes et variables).

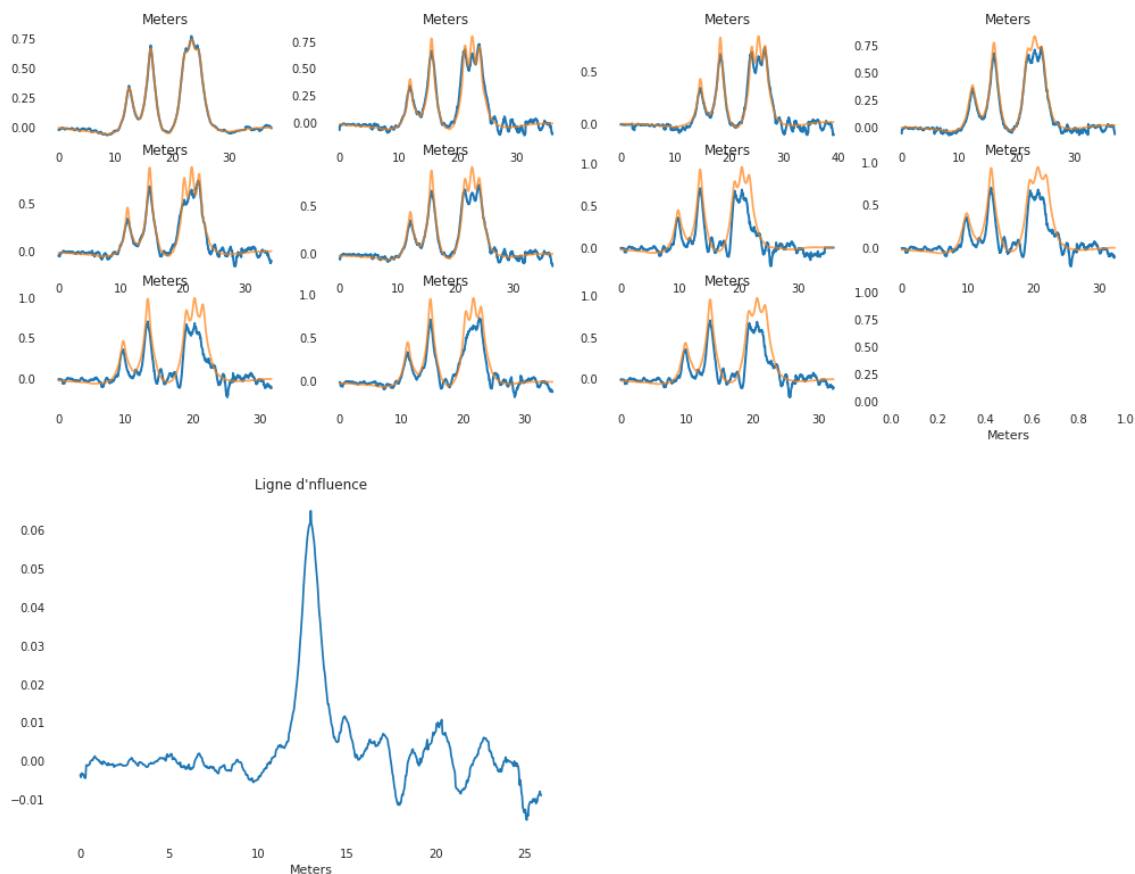


Fig.8 - Reconstruction des signaux des camions de calibration via calibration (capteur 6) sur ce même ensemble

Les résultats pour les autres capteurs sont disponibles sur le Notebook N1 (en plus de l'ensemble du code).

Les résultats sont légèrement améliorés mais ne sont pas satisfaisants pour une utilisation pratique réelle. Il existe donc des paramètres qui n'ont pas été pris en compte et qu'il sera important de déterminer puis modifier pour obtenir les résultats souhaités.

Il était alors possible de suspecter le manque de précision des capteurs, une fréquence d'échantillonnage trop faible, ou encore le niveau de bruit des signaux pouvant impacter la qualité des résultats.

Pondération des signaux issus des camions par niveau de bruit

Pour s'assurer que le niveau de bruit du signal ne soit pas la cause principale des problèmes précédemment observés, il a été ensuite décidé de pondérer les signaux par niveau de bruit dans la minimisation.

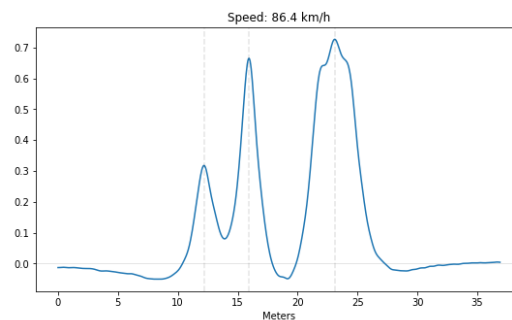


Fig.9 - Exemple de signal d'un camion

Il fallait isoler les cinq premiers mètres (Voir Fig.10), correspondants à la partie du signal la plus bruitée, calculer le niveau de bruit et en déduire un coefficient pour la pondération.

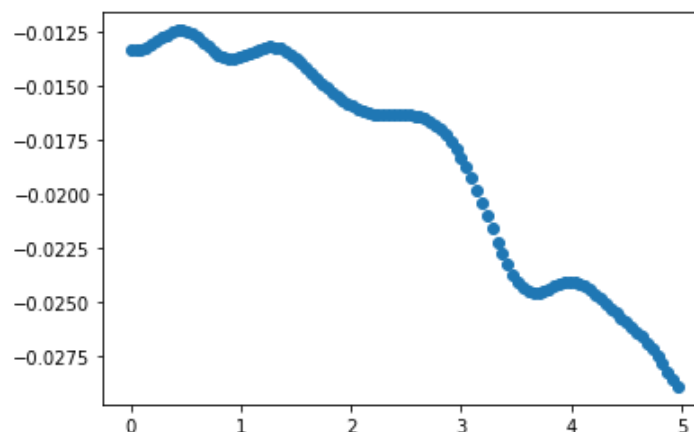


Fig.10 - Isolement des cinq premiers mètres d'un signal

La pente observée nous a conforté dans l'idée d'utiliser une régression linéaire (Voir Fig.11) puis de calculer le niveau de bruit à partir du RSS (residual sum of squares).

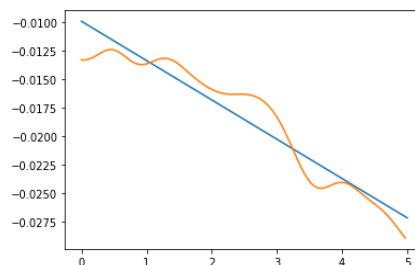


Fig.11 - Régression linéaire sur cet extrait de signal

Une fonction calculait alors le niveau de bruit comme :

$$\text{Niveau de bruit (NB)} = \sqrt{RSS}.$$

Avec RSS le residual sum square de notre régression

Les signaux étaient alors pondérés par $0.001/\text{NB}$ (0.001 pour obtenir des coefficients centrés autour de 1). (Voir Fig.12)

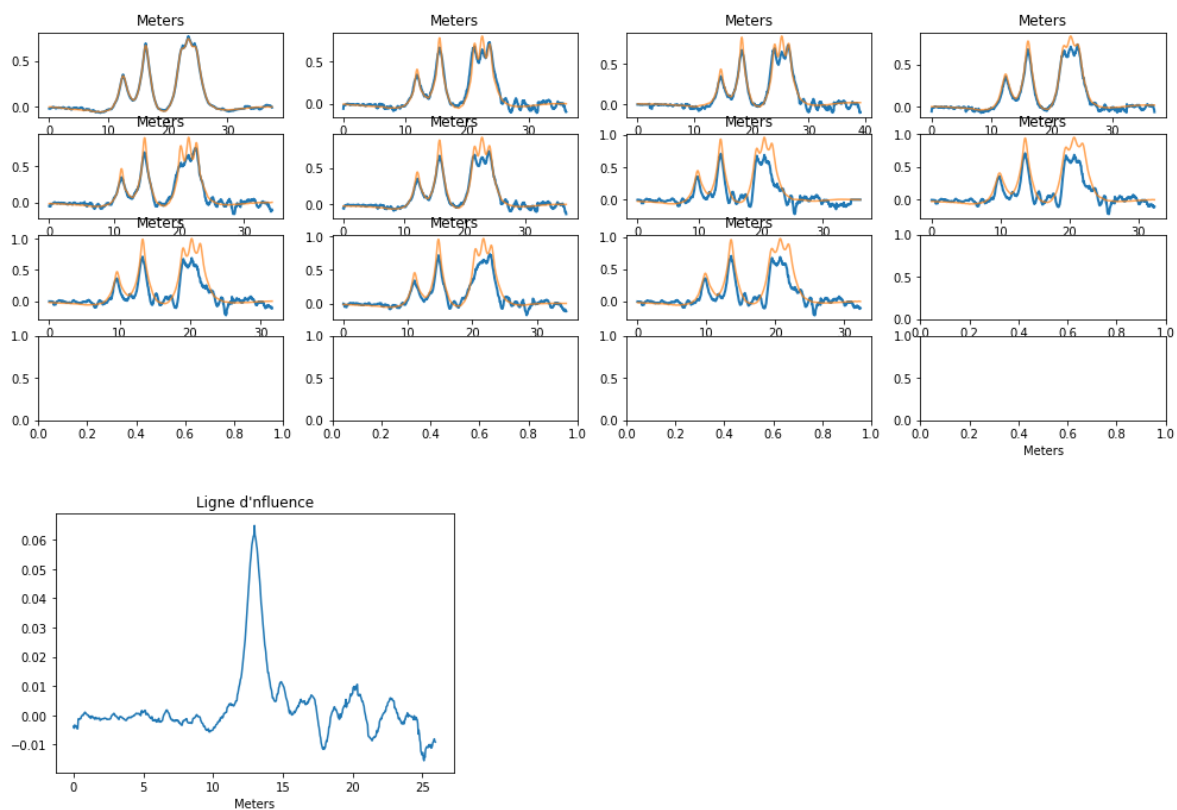


Fig.12 - Reconstruction des signaux des camions de calibration via calibration sur cet ensemble avec pondération par niveau de bruit des signaux

Le niveau de bruit des signaux n'était donc pas le problème majeur. Aucune différence n'est constatée.

Régularisation appliquée à la calibration pour plusieurs camions

Après l'implémentation d'une calibration pour plusieurs camions, l'ajout possible d'une régularisation à cette méthode a également été implémentée, plus précisément les méthodes Total Variation et L2.

Pour rappel :

- **Total Variation (TV)** : Part du principe que les signaux bruités ont une variation totale élevée (c'est-à-dire que l'intégrale du gradient absolu du signal est élevée). Selon ce principe, la réduction de la variation totale du signal, à condition qu'elle soit proche du signal original, permet de supprimer les détails indésirables tout en préservant les détails importants.
- **L2** : Régularisation qui ajuste à la baisse les poids au sein d'un modèle proportionnellement à la somme des carrés de leurs valeurs afin de contrer le surajustement. Cela permet donc de réduire la variance de nos paramètres en ajoutant un biais.

Les détails du code sont disponibles dans le Notebook N3.

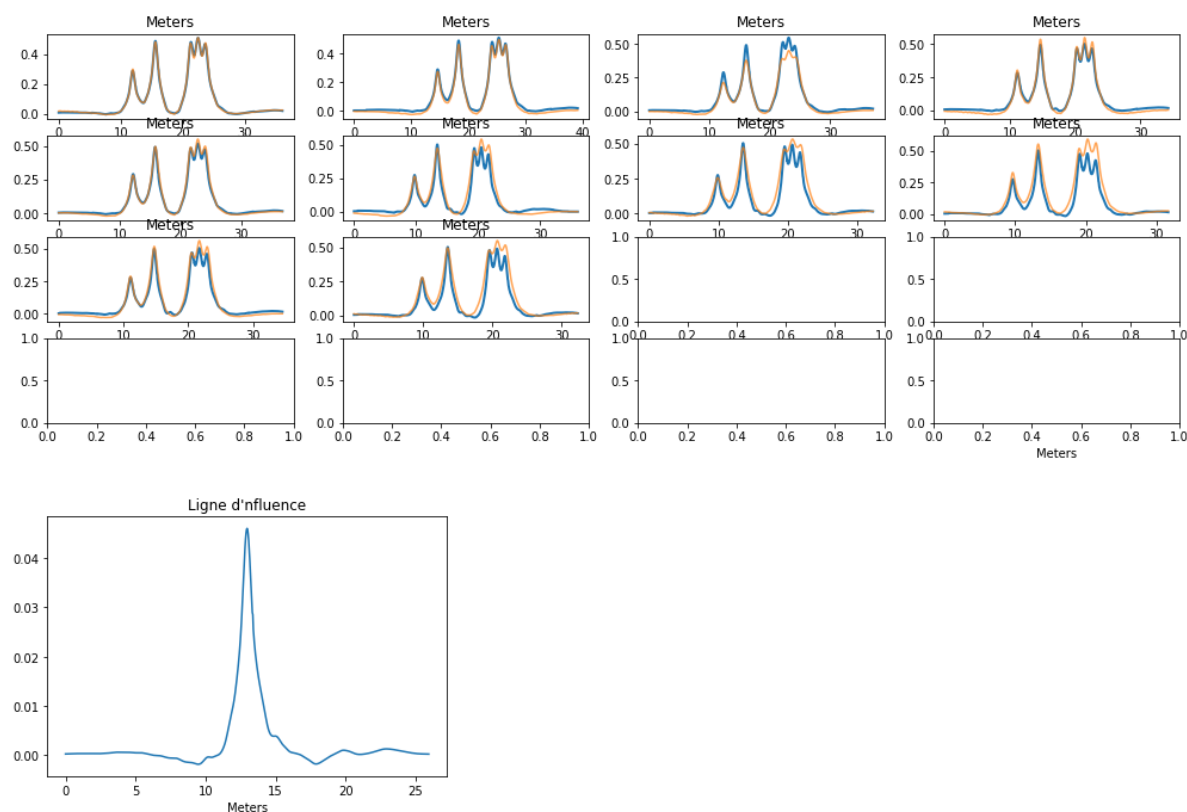


Fig.13- Reconstruction des signaux des camions de calibration avec calibration sur cet ensemble et régularisation TV

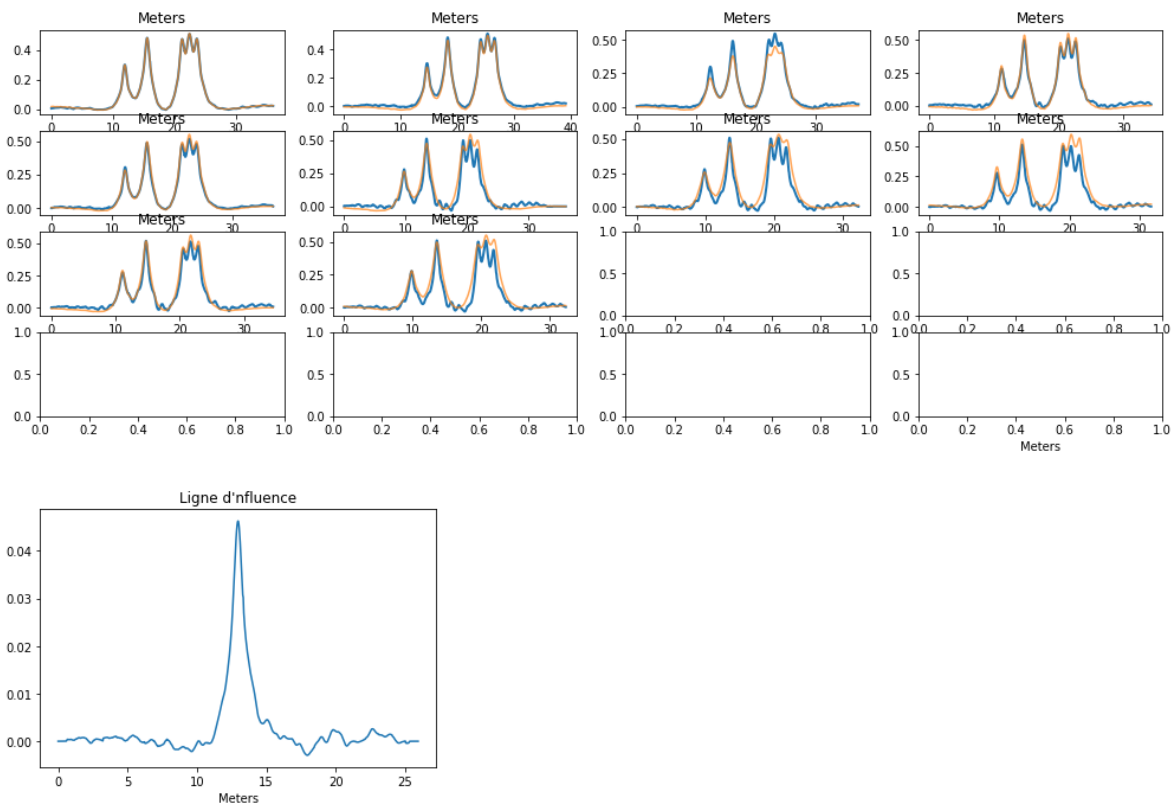


Fig.14- Reconstruction des signaux des camions de calibration avec calibration sur cet ensemble et régularisation L2

La régularisation TV apporte une réelle plus-value sur la reconstruction des signaux. La régularisation L2 quant à elle ne change pas de façon significative les résultats.

La régularisation par splines n'a pas été implémentée bien qu'elle puisse donner des résultats intéressants.

Normalisation de l'échelle des signaux

Une hypothèse formulée concernant les problèmes observés sur les résultats concernait la nécessité de normaliser l'échelle des signaux pour s'affranchir des différentes vitesses des PL de calibration. En effet, la variation de vitesse dans l'ensemble de PL peut poser des soucis d'échelle dus à l'échantillonnage des capteurs.

Dans un premier temps, les signaux des PL ont été interpolés avec la liste des distances associées (vitesse du PL * liste des temps du PL). Il fallait ensuite reconstruire le signal à partir de cette fonction sur les points (d'un intervalle) souhaités.

Cette étape était réalisée dans la fonction de préparation des listes des signaux et des matrices de Toeplitz utiles à la calibration.

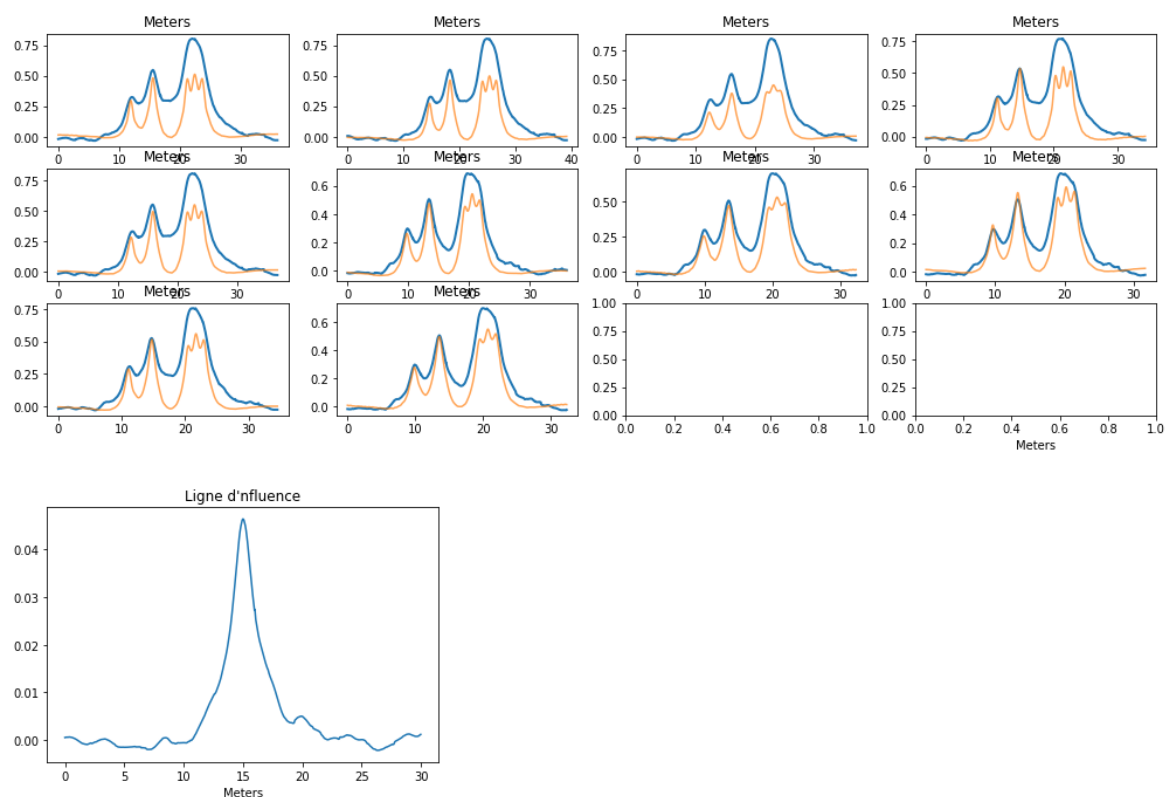


Fig.15 - Reconstruction des signaux après calibration avec normalisation des signaux en suivant la première méthode

Les résultats obtenus (Voir Fig.15) avec cette méthode de normalisation de l'échelle des signaux ne sont pas satisfaisants non plus. Il semblerait qu'il y ait un problème sur la largeur du pic principal de la ligne d'influence.

Pour s'assurer que l'erreur ne soit due à une mauvaise implémentation, une deuxième tentative de normalisation a été entreprise en intégrant cette fois la normalisation du signal dans la création du **namedTuple** Truck.

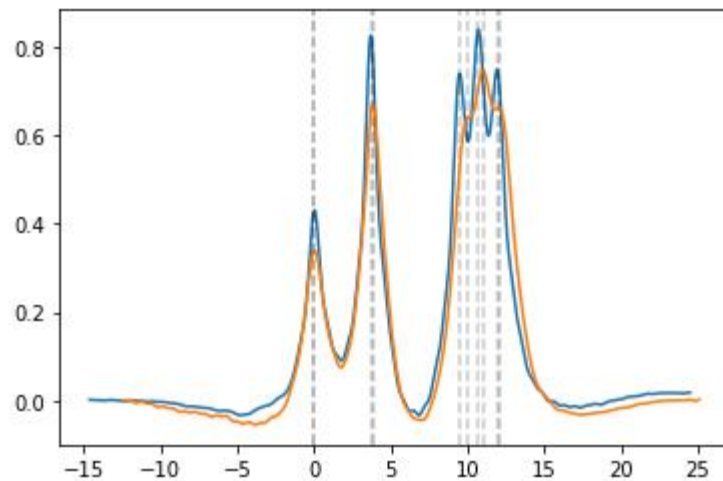


Fig.16 - Comparaison de deux signaux des nouveaux named Tuple Truck (normalisation intégrée)

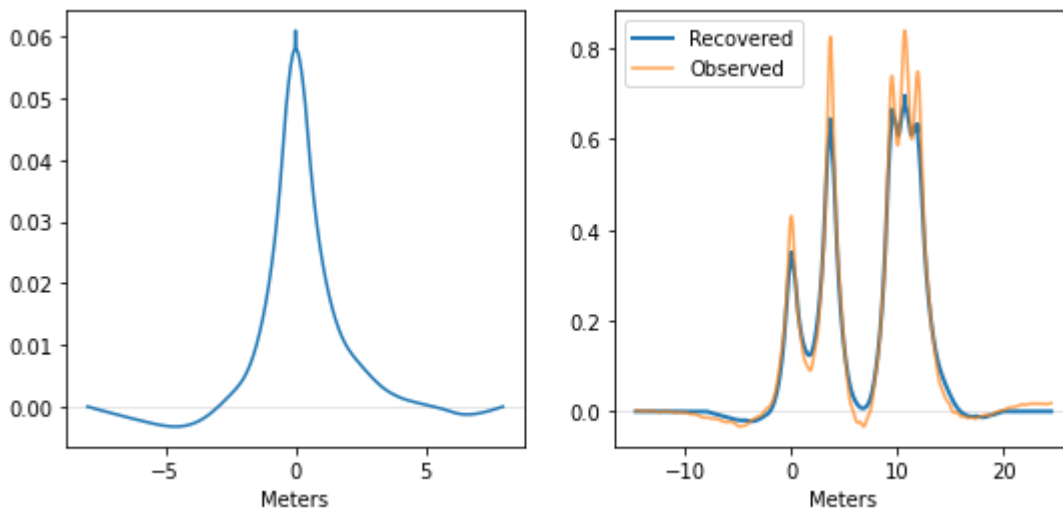


Fig.17 - Reconstruction du signal après calibration avec normalisation du signal intégrée

Pour évaluer cette méthode a été réalisé la pesée en marche sur ces résultats :

- **Sur les camions de calibration :**

Erreur totale absolue moyenne mesurée (en tonnes) : **8.057 t**

- **Sur les camions de trafic :**

Erreur totale absolue moyenne mesurée (en tonnes) : **5.23 t**

L'échelle n'est donc pas la solution permettant la généralisation.

Calibration avec tri sur la vitesse des camions

La constatation faite sur les résultats précédents concerne la vitesse des camions. Les résultats semblent se dégrader pour des vitesses élevées.

Cela peut s'expliquer par l'échantillonnage. Il est cependant compliqué d'expliquer pourquoi la ligne d'influence calculée sur les PL roulant à 70km/h ne se généralise pas.

Elle est calculée à partir des signaux les plus fiables (basse vitesse) et donne les meilleurs résultats sur les PL roulant à cette vitesse (généralisation réussie par tranche de vitesse). Partant du principe qu'il n'y ait qu'une ligne d'influence par point (donc par capteur), il y a un problème entre la théorie et les observations.

.

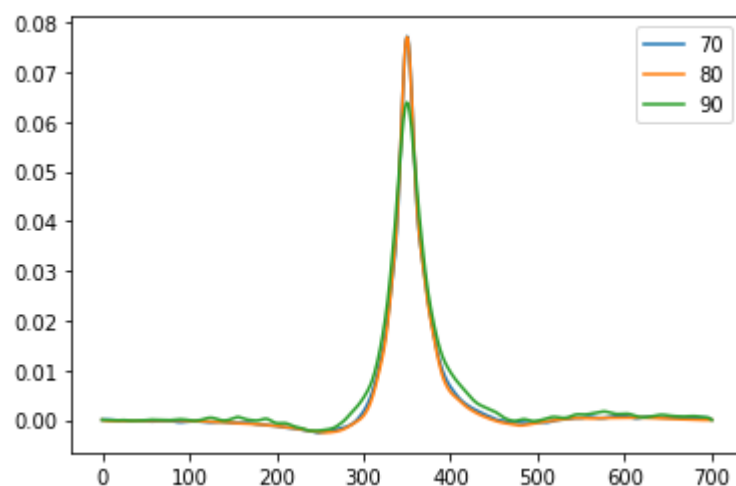


Fig.18 - Lignes d'influence calculée pour des camions roulant à différentes vitesses (70 km/h, 80 km/h et 90 km/h)

Il y a un facteur d'amplitude (Voir Fig.18) entre les lignes d'influence calculées sur des camions roulant à différentes vitesses.

De plus, ces facteurs ne semblent pas évoluer linéairement mais semblent plutôt suivre une courbe parabolique (maximum d'amplitude pour la ligne d'influence des camions roulant à 80 km/h). Ce facteur d'amplitude selon les vitesses peut être la clé pour la suite des recherches.

Facteurs d'amplitude sur la ligne d'influence selon la vitesse des camions

Dans un premier temps, le problème d'amplitude a tenté d'être résolu directement dans la minimisation (de la calibration). Cette fois, au lieu de minimiser une fonction ne dépendant que de la ligne d'influence, il fallait minimiser une fonction dépendant de la ligne d'influence et d'un ensemble de coefficients pondérant les signaux (et de manière induite leur amplitude). Le tout a été implémenté dans un seul tableau pour simplifier le code. Il suffisait alors en début de fonction de séparer ce tableau de deux. Un pour la ligne d'influence et l'autre pour les coefficients (autant que de PL).

```
Alphas finaux : [0.99999277 1.00023322 1.00039534 0.9997654 0.99971593 0.99924522  
0.99906682 0.99841124 0.99910643 0.99887528]  
Alphas finaux : [1.00000383 0.99985646 0.99971732 0.99898155 0.99959821 0.99952971  
0.99940864 0.99905438 0.9991801 0.99874882 0.99894142 0.99902309]  
Alphas finaux : [0.99995068 0.99973542 0.99840037 0.99910812 0.99913218 0.99871379  
0.9972225 0.99759873 0.99712686 0.99705102 0.99723104]  
Alphas finaux : [1.00000045 1.00004677 0.99991587 0.99953333 0.99963691 0.99951472  
0.9995326 ]
```

Fig.19- Coefficients calculés par minimisation sur l'ensemble des camions de calibration pour les quatre capteurs

Les coefficients calculés (Voir Fig.19) sont tous centrés autour de 1, mais ne corrigent pas le problème que nous observions. Ils ne correspondent pas aux différences d'amplitude observées (allant jusqu'à 0.04)

Il était alors intéressant d'observer l'influence de la vitesse sur l'échantillonnage de nos signaux afin de s'assurer que le problème n'était pas dû aux capteurs. (Voir Fig.19)

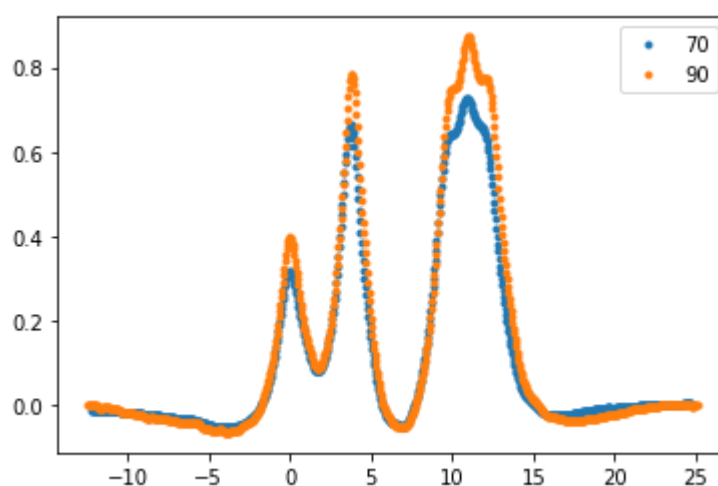


Fig.20- Comparaison d'un signal d'un camion roulant à 70 km/h avec celui d'un camion roulant à 90 km/h

La vitesse ne semble induire qu'un léger décalage en amplitude et en largeur (*Voir Fig.20*).

En faisant le quotient de nos amplitudes maximales (pic le plus haut) entre chaque signal des PL, il était possible d'observer une sorte de corrélation linéaire entre vitesse et facteur d'amplitude.

(*Voir Fig.21*)

Ci-dessous l'exemple obtenu en utilisant un PL comme référence et en calculant le quotient expliqué précédemment par rapport aux autres camions :

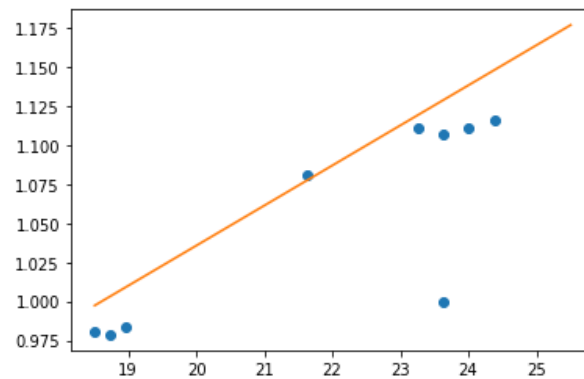


Fig.21 - Rapports d'amplitude des camions de calibration par rapport à un autre camion de calibration selon la vitesse

Optimisation de la calibration

Comme la plupart des hypothèses concernant l'erreur sur la généralisation s'avéraient être peu concluantes (justes pour la plupart mais non suffisantes), il a alors été entrepris d'optimiser au mieux la méthode de calibration afin obtenir les résultats les plus précis possibles, avec des indicateurs de performance (metrics).

Dans un premier temps, une hypothèse a été émise concernant le calcul du poids total d'un PL à partir de la ligne d'influence calculée et de son signal. En effet, le poids total peut être décrit comme le rapport de **l'aire sous le signal du camion** et de **l'aire sous la ligne d'influence**. Cette méthode permettait alors d'ajouter une contrainte dans la fonction de minimisation sur le poids total.

Malheureusement, cette méthode, dépendant de la qualité de la ligne d'influence, retournait des poids totaux calculés variant énormément, parfois à plus de 40% du poids total réel. (Voir Fig.22 et 23)

```
Poids réel : 40.14256
Poids estimé : 41.255534643303626
Différence (en valeur absolue): 1.1129746433036232

Poids réel : 40.14256
Poids estimé : 40.8933289690534
Différence (en valeur absolue): 0.7507689690533965

Poids réel : 40.14256
Poids estimé : 38.505897396720684
Différence (en valeur absolue): 1.6366626032793192
```

Fig.22 - Poids totaux calculés par cette méthode par calibration sur le camion

```
Poids réel : 43.3
Poids estimé : 52.92951082380425
Différence (en valeur absolue): 9.629510823804253

Poids réel : 43.05
Poids estimé : 52.67940088082072
Différence (en valeur absolue): 9.629400880820725

Poids réel : 44.800000000000004
Poids estimé : 53.30839762825562
Différence (en valeur absolue): 8.508397628255615
```

Fig.23 - Poids totaux calculés par cette méthode par calibration sur l'ensemble des camions de calibration

Il a tout de même été tenté d'intégrer cette contrainte à la minimisation avec des intervalles autour du poids total estimé plus ou moins grands. Les variations de qualité des résultats ne permettaient pas d'établir une méthode stable.

Les différences entre les lignes d'influence calculées sur différents capteurs ont ensuite été observées. (Voir Fig.24)

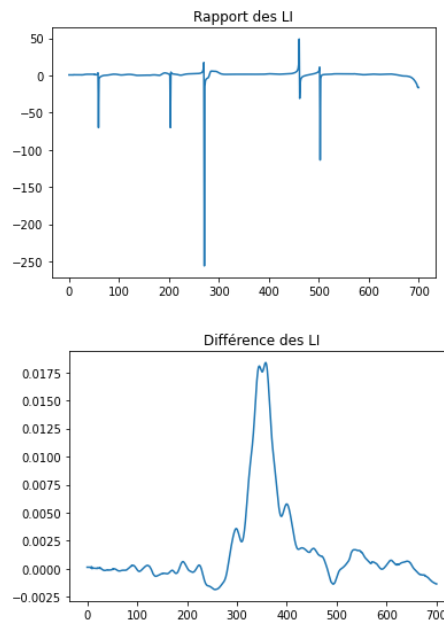


Fig.24 - Rapport et différence des lignes d'influence calculées sur tous les camions de calibration sur les capteurs 3 et 4

Seul un facteur d'amplitude entre les lignes d'influence est observable (normal).

Pour évaluer de façon précise les performances, il fallait alors mettre en place des indicateurs de mesure (métriques) :

- Une métrique sur la reconstruction du signal : distance euclidienne entre signal reconstruit et signal réel. (Voir Fig.25)
- Une autre sur la pesée en marche : erreurs totales et par essieu.

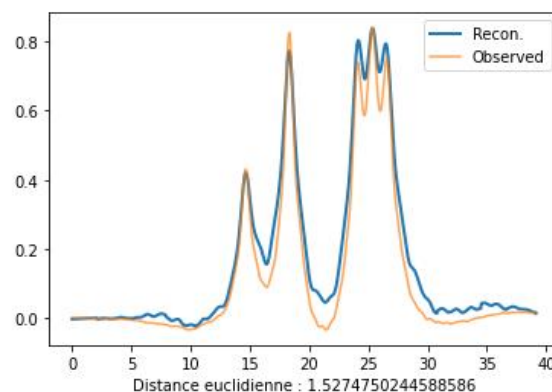


Fig.25 - Exemple de distance euclidienne entre signal réel et reconstruit

Ces métriques ont alors été évaluées sur l'ensemble des PL de calibration pour chaque capteur, pour déterminer les capteurs avec les meilleurs signaux :

- Distance euclidienne moyenne sur le capteur 3 : **3.39**
- Distance euclidienne moyenne sur le capteur 4 : **3.24**
- Distance euclidienne moyenne sur le capteur 6 : **1.87**
- Distance euclidienne moyenne sur le capteur 7 : **4.79**

Il apparait que le capteur 6 (certainement grâce à son positionnement sur la chaussée) est celui à utiliser pour la suite des calculs.

Il a ensuite été tenté de rééchantillonner la ligne d'influence (plutôt que les signaux, vu dans la partie normalisation). La fonction **resample** de la librairie **scipy.signal** a été utilisée dans ce but. (Voir Fig.26)

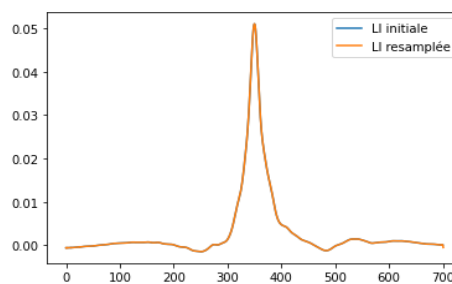


Fig.26 - Exemple de LI ré échantillonnée (sur 5000 points au lieu de 701 puis remis à l'échelle)

Les résultats (sur le critère de la distance euclidienne entre signal reconstruit et réel) n'en ont pas été améliorés (ou peu).

Finalement, le calcul d'une ligne d'influence moyenne entre deux capteurs dans l'espoir de compenser le problème de position latérale des camions par rapport aux capteurs a également été entrepris. Seule la création d'ensembles composés des camions issus de différents capteurs était nécessaire. (Voir Fig.27)

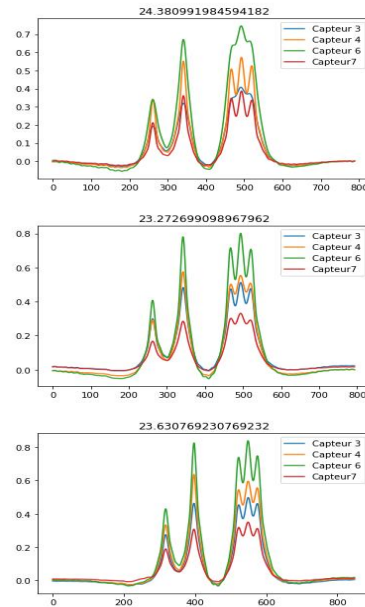


Fig.27 - Exemples de signaux de camions de calibration pour nos quatre capteurs

Les capteurs 4 et 6 semblent être ceux à garder pour obtenir les signaux les plus ‘forts’.

Voici quelques résultats à comparer :

- Distance euclidienne moyenne (signal reconstruit/réel) capteur 4 sur les camions de calibration avec régularisation total variation : **1.74**
- Distance euclidienne moyenne (signal reconstruit/réel) capteur 4 sur les camions de trafic avec régularisation total variation : **1.48**
- Distance euclidienne moyenne (signal reconstruit/réel) capteur 4 et 6 sur les camions de calibration avec régularisation total variation : **2.87**
- Distance euclidienne moyenne (signal reconstruit/réel) capteur 4 et 6 sur les camions de calibration avec régularisation total variation : **2.28**

Les résultats permettent donc d’écarter cette piste également.

Après avoir réalisé ces divers tests, la seule certitude fût qu’il fallait utiliser le capteur 6 pour la suite des calculs. Le reste des expériences n’ont donné aucune solution/piste de recherche majeure.

Conservation de l'énergie

Une idée émise dans le but de contraindre la minimisation lors de la calibration (comme avec le poids total) était d'observer une sorte de conservation de l'énergie entre les PL, en décrivant l'énergie comme :

$$E = \sqrt{\text{aire}(\text{signal}^2)}$$

L'énergie ne semble pas se conserver et semble évoluer en fonction de la vitesse du PL :

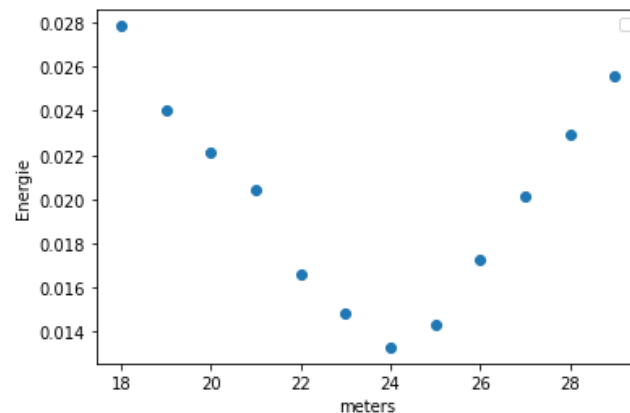


Fig.28 - Evolution de l'énergie selon la vitesse

Il a été essayé de pondérer les signaux utiles à la calibration par ces facteurs, sans résultat. Cette piste n'a pas été plus creusée mais pourrait mériter de l'être dans le futur.

Interpolation de la ligne d'influence selon les vitesses des camions

A ce stade, le problème majeur que représentaient ces facteurs d'amplitude entre les lignes d'influence selon la vitesse des PL n'avait pas de solution.

Il a alors été entrepris d'interpoler un ensemble de lignes d'influence, calculées pour chaque PL, avec les vitesses de ceux-ci.

Dans un premier temps, les PL ont été triés par vitesse dans l'idée de trouver une loi d'évolution des lignes d'influence selon les vitesses.

Cependant, le peu de données à disposition ne permettait pas de faire cela correctement.

L'interpolation a donc été réalisée à partir de chaque camion individuellement.

En prenant l'ensemble des PL de calibration était obtenue une fonction donnant la ligne d'influence en fonction de la vitesse (en m/s), dont l'amplitude variait. (Voir Fig.29)

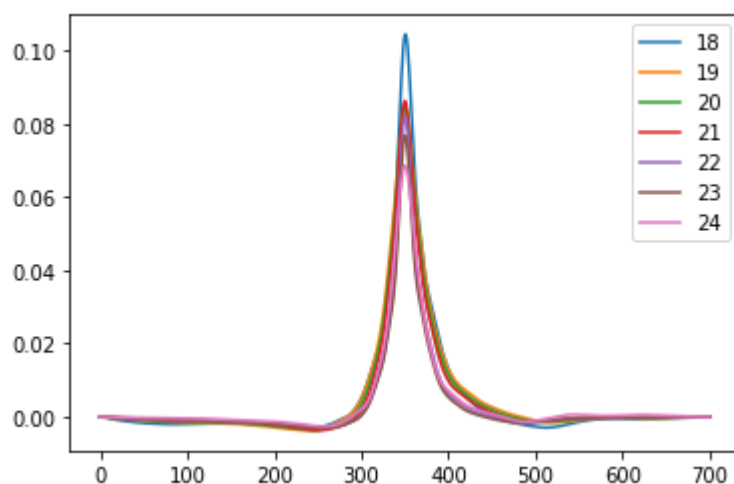


Fig.29 - Ligne d'influence interpolée selon la vitesse des camions (de 18 à 24 m/s)

Des résultats bien plus satisfaisants ont été obtenus, mais l'augmentation de la vitesse continuait de dégrader fortement les résultats. (Voir Fig.30)

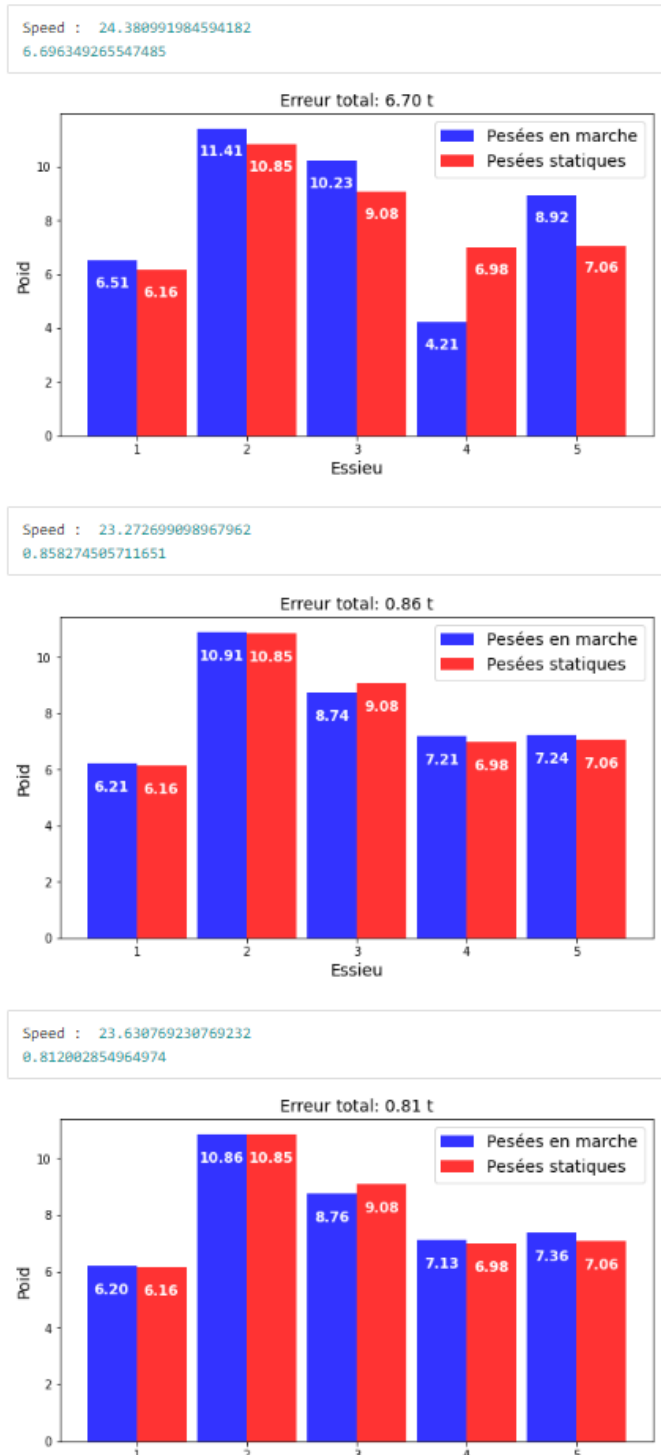


Fig.30- Résultats de l'interpolation sur la pesée en marche pour différentes vitesses

Ce problème pouvait être dû au manque de PL roulant à grande vitesse (plus de 24m/s) dans l'ensemble de calibration.

Les PL de trafic possédaient des vitesses bien plus élevées en moyenne que ceux de calibration. Comme les poids des PL de l'ensemble de trafic étaient connus, certains ont été utilisés pour l'interpolation, en complément des PL de calibration. (Voir Fig.28)

Les meilleurs résultats (sans autre optimisation) ont été obtenus de cette façon.

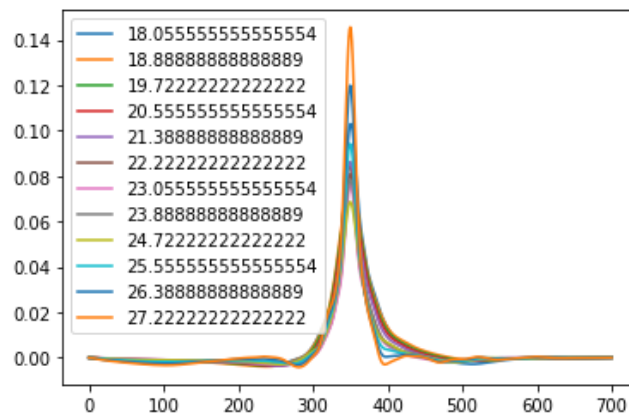


Fig.31 - Interpolation de la LI selon les vitesses avec les camions de calibration et quelques camions de trafic roulant à plus de 24 m/s

- Erreur totale moyenne (en tonnes) sur les camions de calibration : **1.78 t**
- Erreur totale moyenne (en tonnes) sur les camions de trafic : **2.78 t**

Sachant que le poids total des PL s'approchait pour la plupart de 41 tonnes, les résultats correspondaient à une erreur de plus ou moins **4%** pour les PL de calibration et **7%** pour les PL de trafic.

Pour améliorer les résultats, différentes expérimentations ont été réalisées en complément. Les fonctions d'interpolation ont été ré implémentées à la main pour les comparer et garder la meilleure.

- **Interpolation linéaire ré implémentée**

Erreur totale moyenne (en tonnes) sur les camions de calibration : **116 t**

- **Interpolation polynomiale ré implémentée**

Erreur totale moyenne (en tonnes) sur les camions de calibration : **3.2 t**

Erreur totale moyenne (en tonnes) sur les camions de trafic : **21.54 t**

- **Par machine learning :**

Il a également été tenté d'utiliser une *ridge régression* pour prédire l'évolution de la ligne d'influence selon la vitesse. (Voir Fig.32)

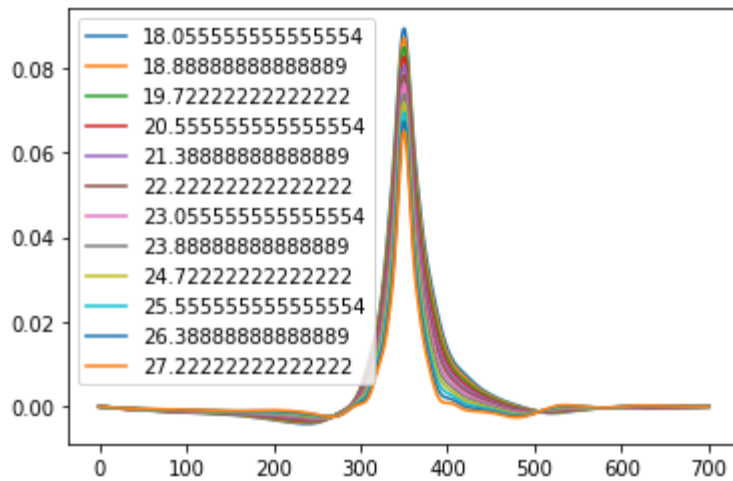


Fig.32 - Interpolation par ridge regression de la LI selon les vitesses

Erreur totale moyenne (en tonnes) sur les camions de calibration : **3.73 t**

L'interpolation marchait presque parfaitement pour les vitesses entre 18m/s et 23 m/s mais les résultats variaient de manière significative pour des vitesses supérieures.

Les maximums d'amplitude des lignes d'influence interpolée selon les vitesses ont alors été observés. (Voir Fig.33)

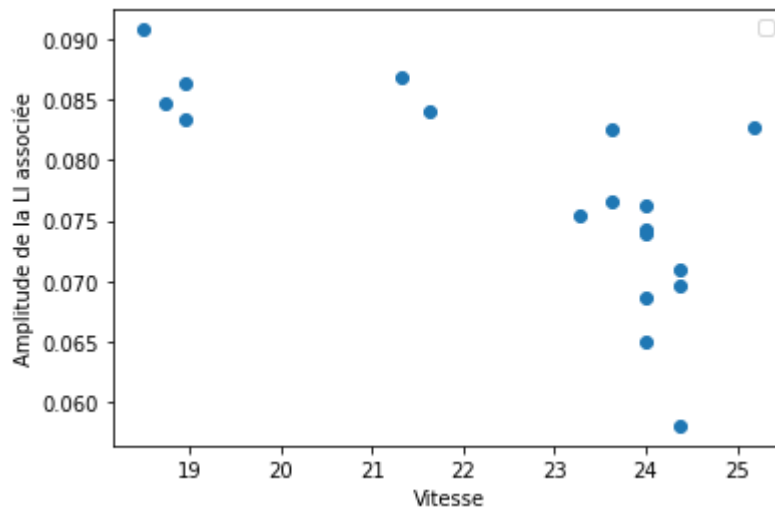


Fig.33 - Maximums d'amplitude des LI interpolées selon les vitesses

Autour de 23-24 m/s, il y a beaucoup de disparité, l'interpolation est donc compliquée à ce niveau. Ce phénomène peut être dû à notre faible quantité de données, qui n'est donc pas forcément représentative de la réalité. Avec une quantité de données bien plus importantes, cette méthode

donnerait assurément de meilleurs résultats (pour rappel, les données de 25 PL étaient utilisables).

Pour finir avec l'interpolation, une interpolation 2D a été implémentée, pour déterminer l'évolution de la ligne d'influence en fonction de la distance et du temps. Cela n'a pas marché. *(Tous les résultats du rapport sont observables en détail dans les Notebooks).*

De plus, une idée a été émise concernant la calibration selon le poids total des PL. En effet, si les poids des PL utilisés en calibration variaient de façon significative, il faudrait les trier par poids pour faire différentes calibrations. Pour automatiser cette tâche, un algorithme de clustering (apprentissage non supervisé) a été implémenté avec **sklearn** pour trier les PL en fonction de leur liste des poids par essieux (en trois catégories). *(Voir Fig.34)*

```
Poids total : 20
Peu chargé

Poids total : 22
Peu chargé

Poids total : 24
Peu chargé

Poids total : 26
Peu chargé

Poids total : 28
Normal

Poids total : 30
Normal

Poids total : 32
Très chargé

Poids total : 34
Très chargé

Poids total : 36
Très chargé

Poids total : 38
Très chargé
```

Fig.34 - Résultats du clustering en trois catégories selon les poids sur essieux

Pesée en marche (PEM)

Première estimation des poids

Une fois la calibration pour plusieurs camions fonctionnelle, il a fallu implémenter les fonctions permettant la pesée en marche pour observer les résultats qui nous intéressent (bien que dans les résultats précédents étaient déjà visibles des résultats de PEM). Jusque-là, seul le critère de la distance euclidienne entre signal reconstruit et signal observé était utilisé (bien que nous ayons déjà observé des résultats de PEM dans la partie précédente pour des questions de compréhension).

Toujours en utilisant le modèle physique B-WIM, les poids sont isolés en partant de la ligne d'influence (d'où l'importance d'une calibration efficace et précise).

En utilisant les mêmes notations que pour la minimisation correspondante à la calibration, la pesée en marche revient à :

$$\underset{\mathbf{w} \in \mathbb{R}^4}{\text{minimize}} \quad \|\mathbf{H}\mathbf{w} - \mathbf{y}\|^2$$

Avec :

$$\mathbf{H} = [\mathbf{D}_1 \mathbf{h} \quad \mathbf{D}_2 \mathbf{h} \quad \dots \quad \mathbf{D}_A \mathbf{h}].$$

(cf page 9 pour les autres notations)

Plusieurs cas sont alors à différencier :

- Lorsque la calibration et la déconvolution est faite sur le même PL, le résultat est presque parfait (comme observé précédemment)
- Lorsque la calibration est effectuée sur un PL et que la déconvolution est faite sur les autres PL, les résultats varient énormément. Le phénomène de dégradation des résultats selon la vitesse est également visible. (Voir Fig.35 et 36)

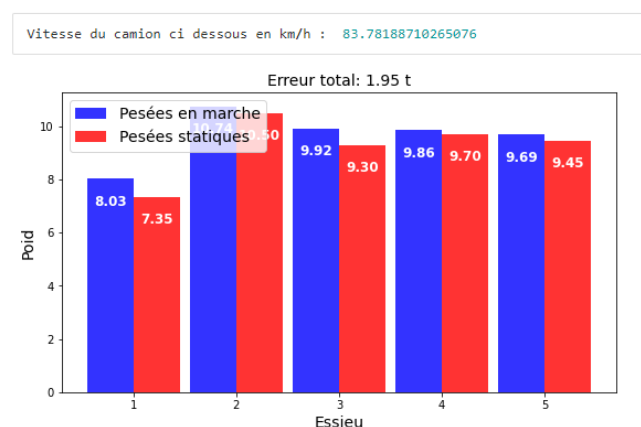


Fig.35 - Résultat sur la PEM avec calibration sur un camion de calibration avec vitesse proche du camion

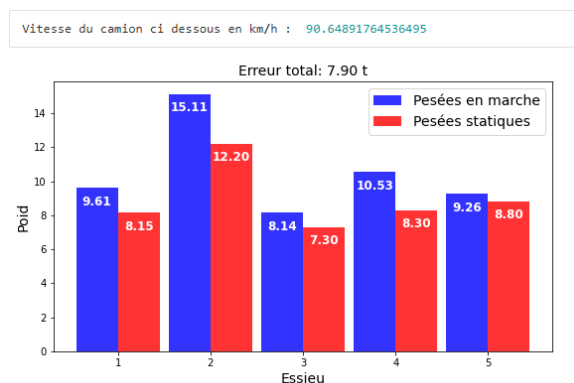


Fig.36 - Résultat sur la PEM avec calibration sur un camion de calibration avec vitesse éloignée du camion

Erreur totale moyenne (en tonnes) sur les camions de calibration : **12.156 t**

Erreur totale moyenne (en tonnes) sur les camions de trafic : **5.74 t**

- Lorsque la calibration est réalisée sur l'ensemble des PL de calibration, les résultats sont légèrement meilleurs mais varient de la même manière.

Erreur totale moyenne (en tonnes) sur les camions de calibration : **12.074 t**

Erreur totale moyenne (en tonnes) sur les camions de trafic : **5.68 t**

Comme les résultats semblent dépendre de la vitesse (mais que la normalisation n'a pas apporté la solution espérée), les camions ont été triés en trois groupes de vitesse (70 km/h, 80km/h, 90 km/h).

Les résultats sont nettement meilleurs. Le tri par vitesse donne des résultats (erreurs totales en tonnes) satisfaisants. Plus la vitesse est faible, meilleurs ils sont (certainement dus à l'échantillonnage du capteur).

- **Pour 70 km/h :**

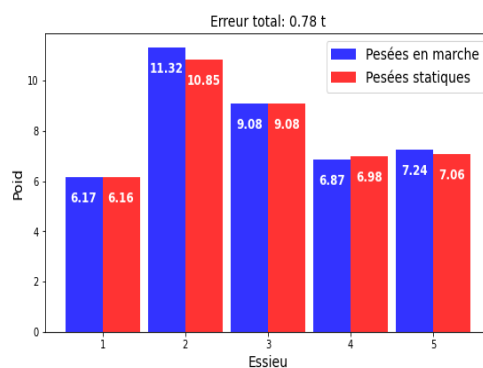
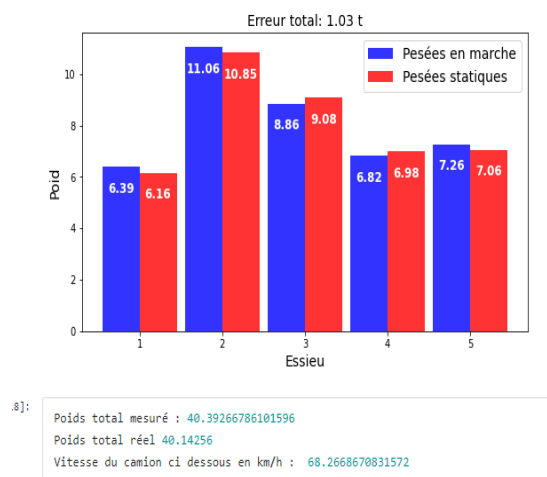


Fig.37 - PEM sur des camions roulant à 70 km/h après calibration sur des camions roulant à cette vitesse

- **Pour 80 km/h :**

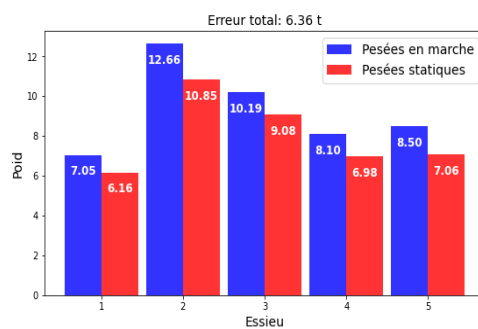
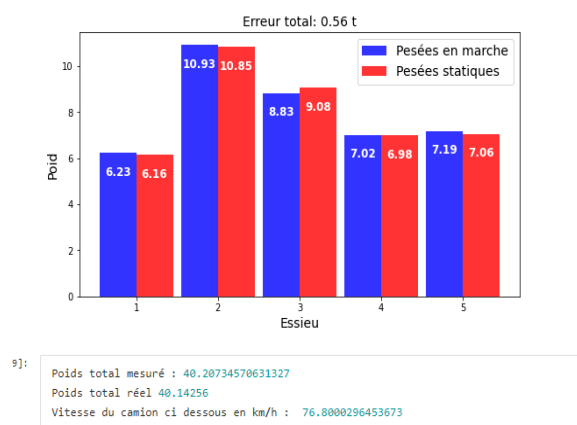


Fig.38 - PEM sur des camions roulant à 80 km/h après calibration sur des camions roulant à cette vitesse

- Pour 90km/h :

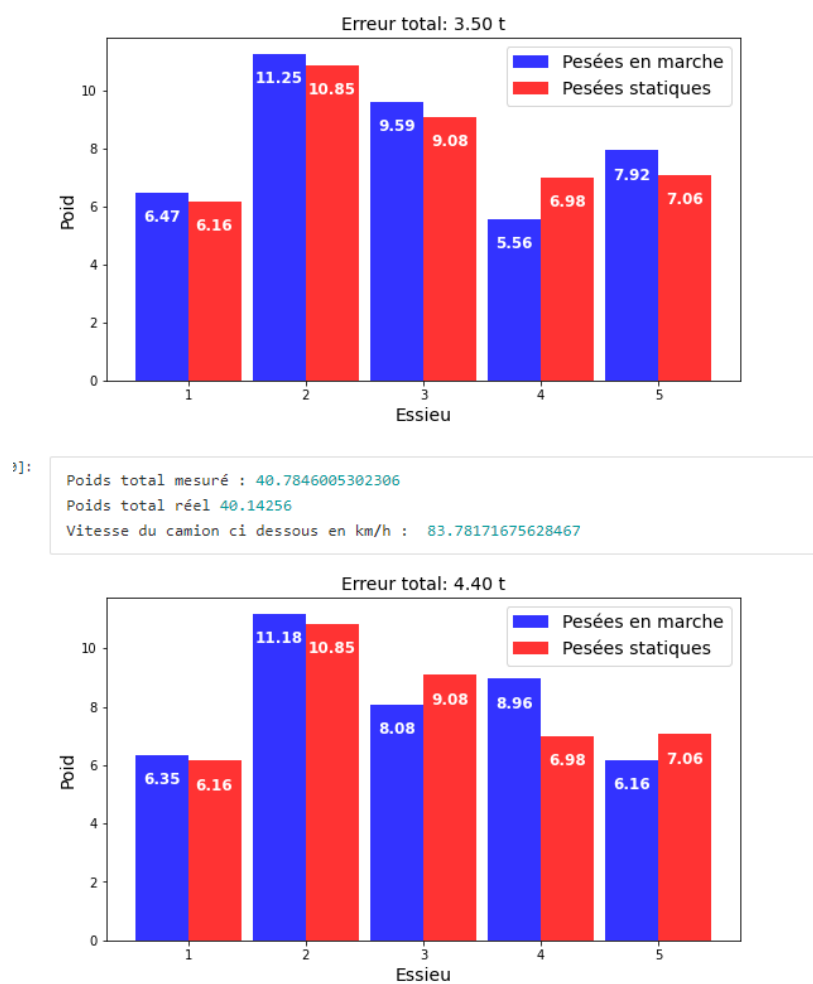


Fig.39- PEM sur des camions roulant à 90 km/h après calibration sur des camions roulant à cette vitesse

La vitesse joue un rôle majeur, comme constaté dans la partie calibration.

Décalage des pics sur les signaux

Une idée supplémentaire pour améliorer les résultats découlait du constat que la fréquence d'échantillonnage pouvait entraîner un déplacement de la position réelle des pics des signaux (correspondants aux passages des essieux sur les capteurs). Il était alors imaginable de décaler les pics des PL sur un certain intervalle puis garder la meilleure combinaison.

- **Décalage global**

Dans un premier temps, les pics ont tous été décalés du même décalage pour corroborer cette idée. (Voir Fig.40)

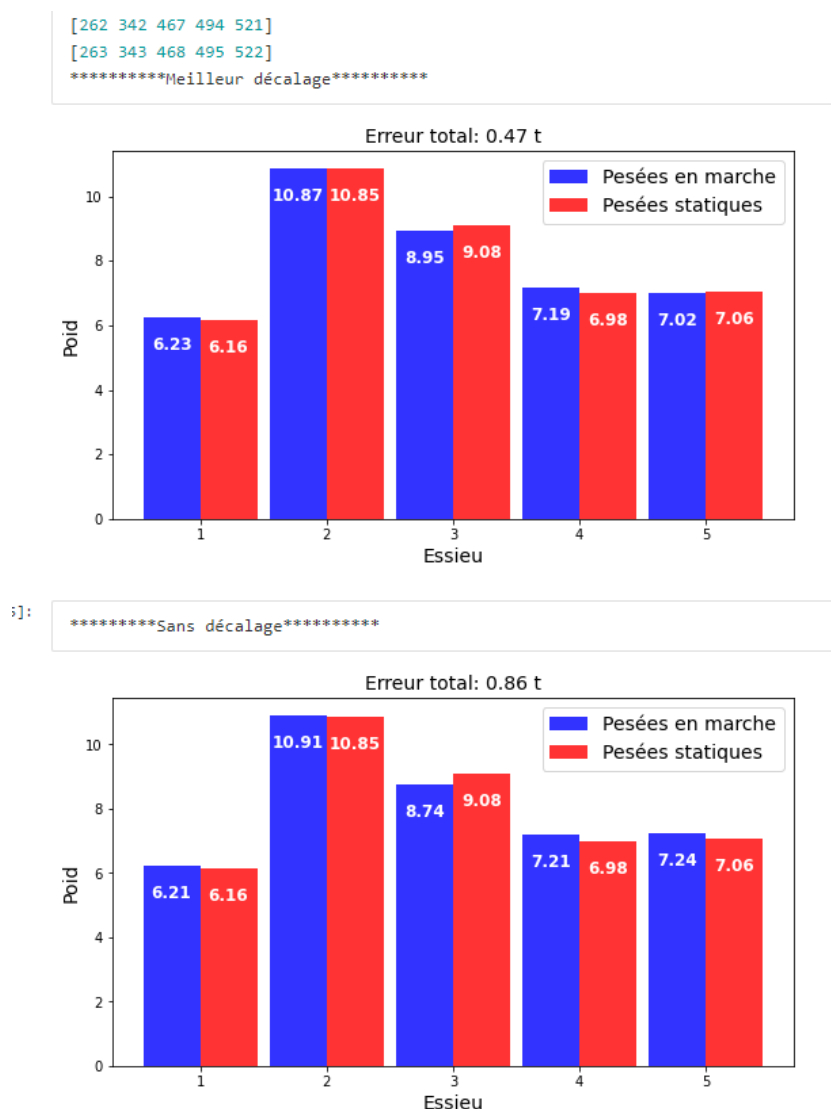


Fig.40 - Comparaison de la PEM avec et sans décalage global

Le décalage global apporte déjà une amélioration conséquente.

Erreur totale (en tonnes) sur les camions de calibration : **1.24 t**

Pourcentage d'erreur totale moyen sur les camions de calibration : **3%**

Erreur totale (en tonnes) sur les camions de trafic : **2.28 t**

Pourcentage d'erreur totale moyen sur les camions de trafic : **5.56%**

- **Décalage local**

Chaque pic a alors été décalé indépendamment des autres. Pour ne pas trop alourdir les calculs et obtenir des résultats dans un temps raisonnable, nous avons réduit le décalage sur un intervalle centré sur chaque pic (intervalle de [pic -2, pic+2]), ce qui correspondait à nombre de combinaisons déjà élevé.

Les résultats, longs à calculer, en ont été encore améliorés (problème de sauvegarde des résultats, il faudrait relancer les calculs sur une grosse machine).

Une variante a été implémentée pour permettre un décalage des limites si la combinaison trouvée comporte une/plusieurs valeurs aux limites de l'intervalle de décalage.

Pour pallier au problème du temps de calcul, la bibliothèque **nevergrad** a été utilisée pour trouver rapidement un minimum local dans l'ensemble des combinaisons décrit ci-dessus. Les résultats ne sont donc pas reproductibles (minimum local) et l'algorithme retourne au pire des cas les pics initiaux si aucune combinaison améliorante n'a été trouvée.

Le critère d'évaluation a également été modifié, en utilisant la distance euclidienne entre signal reconstruit et signal mesuré plutôt que l'erreur sur les poids. Cela permettra par la suite de déployer cette méthode (décalage des pics) sur des PL dont les poids ne sont pas connus.

En incluant cette méthode à la fonction de calibration (utilisation des pics décalés plutôt que des pics initiaux), nous obtenons (calibration et déconvolution sur le même PL) :

Erreur totale (en tonnes) sur les camions de calibration : **0.369 t**

Pourcentage d'erreur totale moyen sur les camions de calibration : **0.25%**

Erreur totale (en tonnes) sur les camions de trafic : **0.171 t**

Pourcentage d'erreur totale moyen sur les camions de trafic : **0,018%**

Un problème rencontré sur certains PL était l'impossibilité de les utiliser dans les calculs car ils comportaient moins de pics que d'essieux. Cela était dû à une mauvaise mesure du signal par les capteurs. (Voir Fig.41)

Nous avons alors utilisé ce principe de décalage des pics pour trouver le/les pics manquants au cas où nous rencontrions ce problème.

Il suffisait donc de créer une liste avec le bon nombre d'essieux et de tester différentes valeurs puis de garder la meilleure (toujours sur le critère de reconstruction du signal). (Voir Fig.42)

```
Peaks : [260 340 493]
Weights : [ 6.16068 10.8499  9.08406  6.98472  7.0632 ]

<IPython.core.display.Javascript object>
```

```
Peaks : [275 358 488 508 531]
Weights : [6.2  5.05 7.95 7.65 7.4  7.65]
```

Fig.41 - Exemples de camions inutilisables pour nos calculs

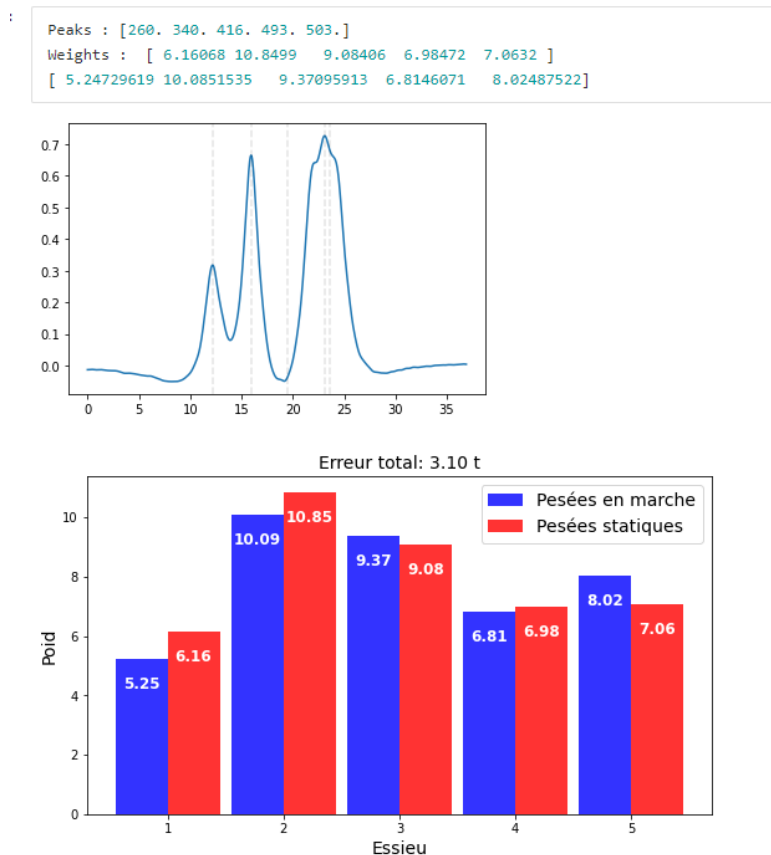


Fig.42 - Exemple de résultat de PEM sur un camion initialement inutilisable

Pour simplifier le calcul et éviter de chercher les meilleurs résultats sur toutes les positions de pics possibles (c'est à dire pour connaître à l'avance la position relative du pic manquant), l'utilisation du critère de la largeur des pics a été employée pour voir si le plus large pouvait correspondre à deux (ou plus) pics superposés.

Ce critère n'était pas suffisant. (Voir Fig.43)

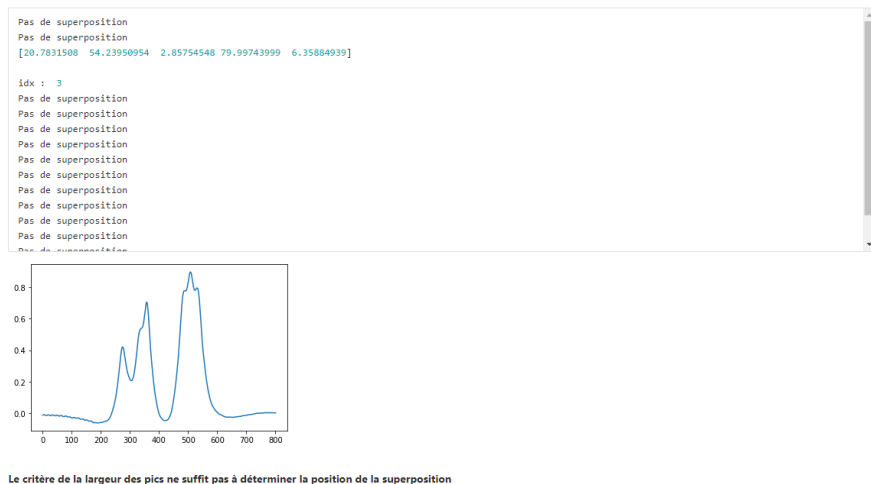


Fig.43- Test de l'utilisation du critère du pic le plus large

Le principe de test exhaustif des positions des pics a été utilisé sur des signaux ne comportant aucune anomalie (sorte de méthode autonome).

Toujours sur le critère du signal reconstruit et avec une calibration sur chaque PL, des résultats quasiment parfaits ont été obtenus. (Voir Fig.44)

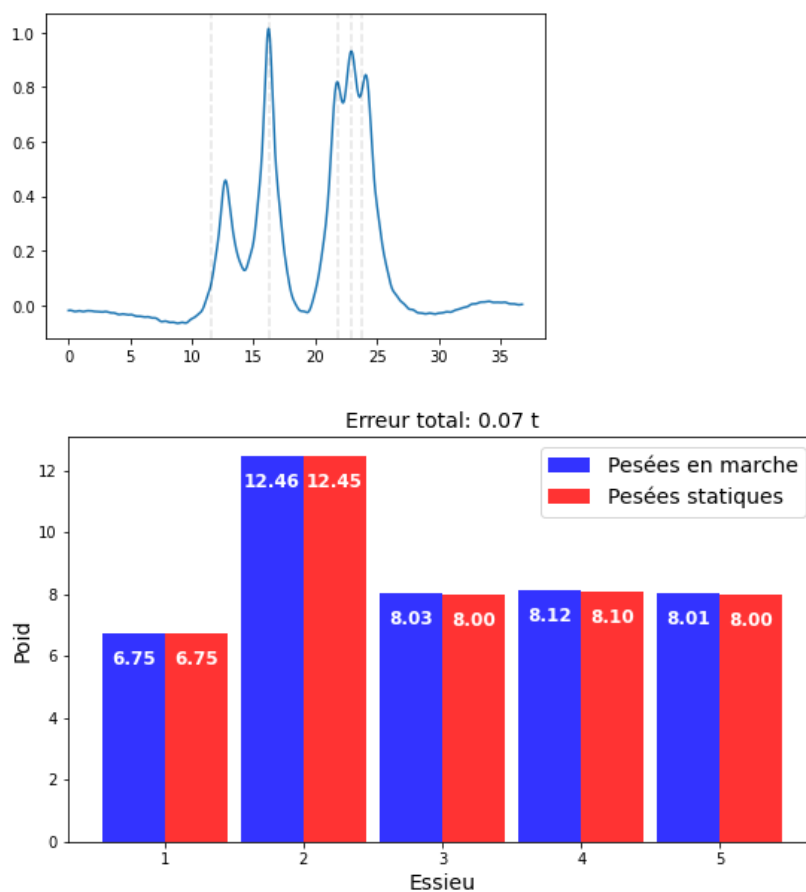


Fig.44- PEM avec décalage pic par pic après calibrations sur le même camion

Comme l'objectif était d'utiliser cette méthode en déploiement, l'utilisation d'une ligne d'influence 0 (calculée à partir d'un camion aléatoirement tiré) a été tentée, en espérant que le décalage compenserait la différence avec la meilleure ligne d'influence pour chaque PL.

Le même problème qui a été rencontré initialement : selon les vitesses, les résultats (bien que améliorés), variaient énormément.

Pour finir, la surestimation du nombre d'essieux a été réalisée pour voir si la méthode précédente était 'saine'. En demandant donc à l'algorithme de trouver une combinaison de pics comportant un pic de plus que d'essieux, le pic en trop calculés s'alignait à tous les coups sur l'un des pics existants. (Voir Fig.45)

[266. 367. 367. 527. 559. 591.]

Fig.45 - Résultats de la surestimation du nombre de pics

Procédure d'évaluation des statistiques sur nos méthodes

A l'issue de toutes ces recherches, il était temps de mettre en place une procédure pour évaluer de manière précise ces différentes méthodes.

Pour cela a été mis en place une procédure de statistiques fonctionnant comme suit :

- **Création de deux sous-ensembles**

A partir de l'ensemble des PL à notre disposition sont créés, selon une répartition prise en paramètre, deux sous-ensembles de PL de manière aléatoire (utilisation de la fonction ***random.shuffle()*** de la bibliothèque ***NumPy*** pour mélanger les indices).

- **Calibration**

Le premier sous ensemble sert à la calibration (calcul de la fonction d'interpolation donnant la ligne d'influence en fonction de la vitesse). Les lignes d'influence individuelles servant à l'interpolation sont calculées avec une régularisation total variation.

- **Pesée en marche**

La fonction interpolée (ligne d'influence/vitesse) est utilisée pour réaliser la pesée en marche des PL du second sous-ensemble. La fonction prend en paramètre le décalage des pics souhaité (si nous en voulons un).

Plus précisément, la pesée peut se faire sans décalage des pics, avec un décalage sur les combinaisons de l'intervalle [pic-2,pic+2] pour chaque pic ou avec un décalage pic par pic (méthode issue de la séparation de deux pics superposés).

- **Récupération des métriques**

Une fois ces trois étapes terminées, nous récupérons les métriques qui nous intéressent, les stockons dans des tableaux ***NumPy*** et les sauvegardons au format ***.npy***.

Les données sont ensuite étudiées dans un autre notebook simplement en chargeant ces données au format ***.npy***.

Les métriques récupérées sont :

- L'erreur totale (en tonne) sur chaque camion
- L'erreur totale en valeur absolue (en tonne) sur chaque camion
- L'erreur (en tonne) sur chaque essieu

Il est facilement possible de récupérer d'autres métriques en les ajoutant à la fonction.

Plusieurs choses ont alors été observées :

- **Procédure sans décalage des pics**

Les premiers résultats observés sont ceux de cette procédure (au moins itérée 30 fois pour considérer nos résultats comme des statistiques) sans effectuer de décalage des pics. Les résultats pour plusieurs répartitions de PL de calibration/PEM ont été calculés pour ensuite observer l'évolution des intervalles de confiance (à 95%) selon ces répartitions. Voici les résultats (sur les erreurs totales) :

- Pour 10% de PL de calibration

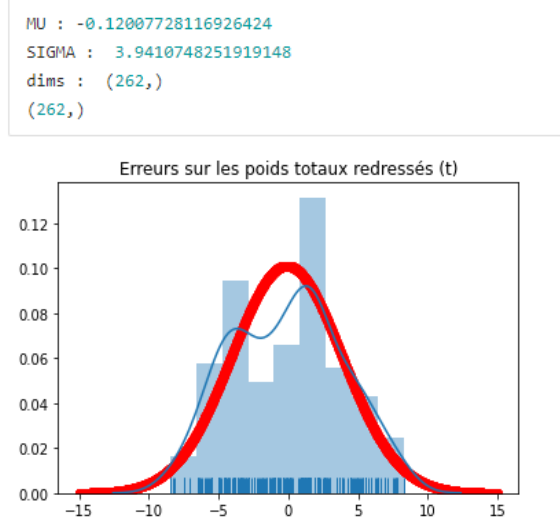


Fig.46 - Procédure avec 10% de camions de calibration

- Pour 20% de PL de calibration

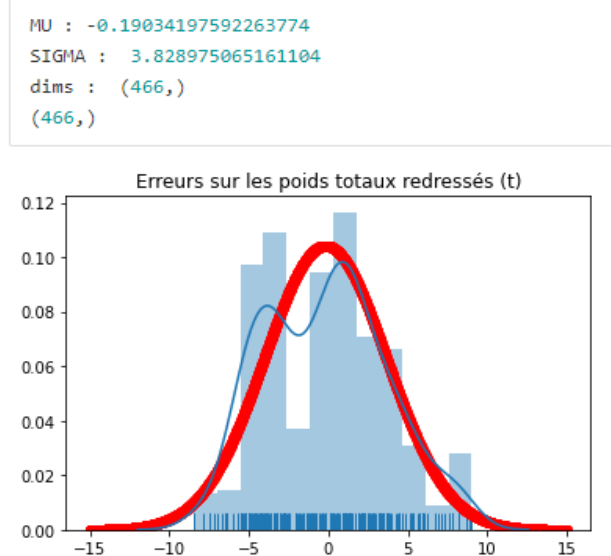


Fig.47 - Procédure avec 20% de camions de calibration

- Pour 30% de PL de calibration

```
MU : 0.16032629368421647
SIGMA : 3.744239464638809
dims : (434,)
(434,)
```

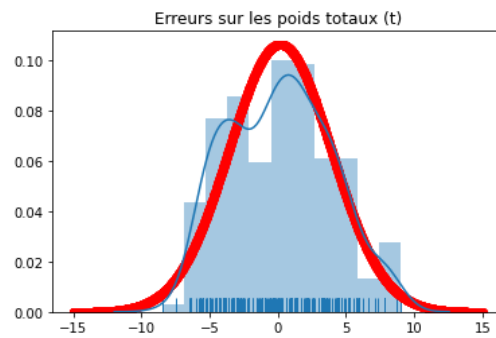


Fig.48 - Procédure avec 30% de camions de calibration

- Pour 40% de PL de calibration

```
MU : -0.07515429479057788
SIGMA : 3.719427856656481
dims : (394,)
(394,)
```

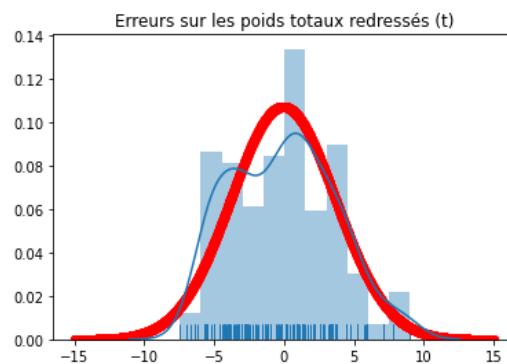


Fig.49 - Procédure avec 40% de camions de calibration

- Pour 50% de PL de calibration

```

MU : -0.05791139054548957
SIGMA : 3.7977474805471045
dims : (342,)
(342,)

```

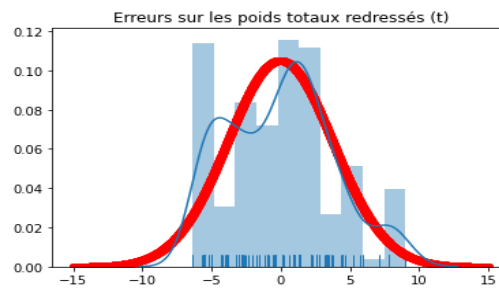


Fig.50 - Procédure avec 50% de camions de calibration

- Pour 60% de PL de calibration

```

MU : 0.6678147518731834
SIGMA : 3.957563357106753
dims : (300,)
(300,)

```

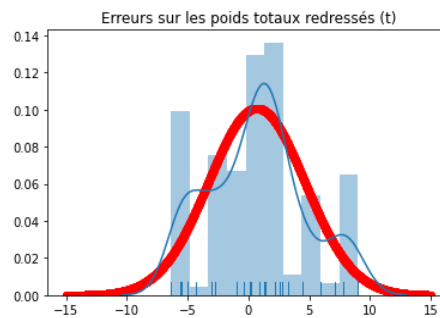


Fig.51 - Procédure avec 60% de camions de calibration

- Pour 70% de PL de calibration

```

MU : 0.3152516874220737
SIGMA : 4.228123772818377
dims : (210,)
(210,)

```

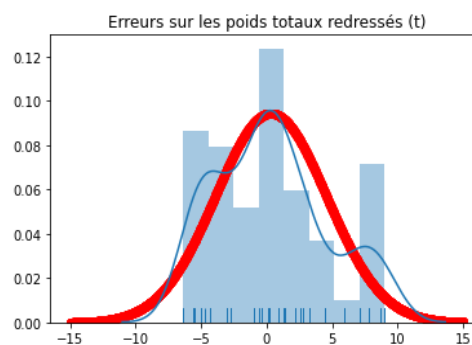


Fig.52 - Procédure avec 70% de camions de calibration

- Pour 80% de PL de calibration

```

MU : 0.040023931600414876
SIGMA : 4.4708912119361575
dims : (150,)
(150,)

```

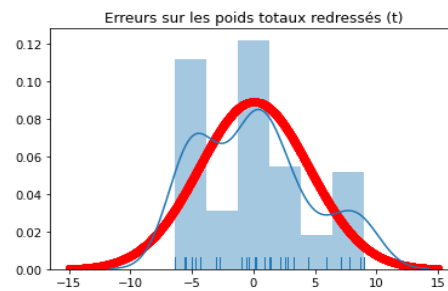


Fig.53 - Procédure avec 80% de camions de calibration

- Pour 90% de PL de calibration

```

MU : -0.11820339866902453
SIGMA : 4.578770480051701
dims : (60,)
(60,)

```

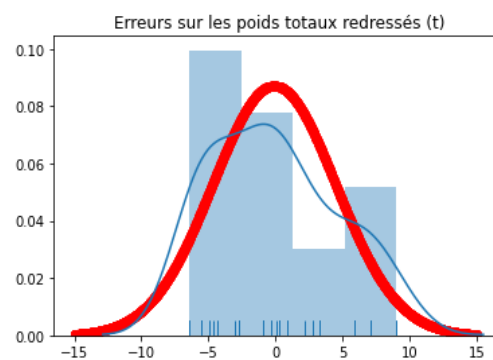


Fig.54 - Procédure avec 90% de camions de calibration

- Et les lois d'évolution de certains paramètres en fonction de la répartition :

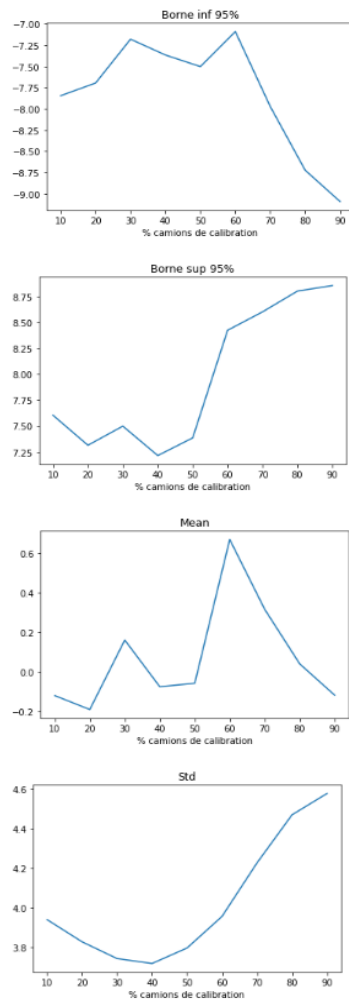


Fig.55 - Evolution de la borne inférieure/supérieure des intervalles de confiance à 95%, de la moyenne et de l'écart-type en fonction de la répartition camions de calibration/trafic

Du fait de la faible quantité de données, il est difficile d'extraire des informations de ces évolutions. Lorsque la répartition devient grande, la variance augmente (par exemple 90% = 1 seul camion de pesée).

Une répartition autour de 40% de camions de calibration semble donner le meilleur compromis. Cette procédure a alors été reproduite avec les deux types de décalage de pics sur cette répartition (40%) (car les calculs sont extrêmement lourds).

Pour rappel, le décalage se fait sur le critère de la distance euclidienne entre signal reconstruit et signal observé. Les poids des PL n'ont pas besoin d'être connus à l'avance et il est donc possible de déployer cette méthode sur de nouveaux camions.

- Procédure avec décalage sur les combinaisons :

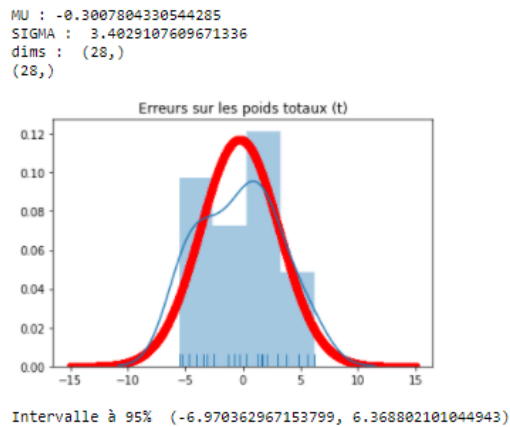


Fig.56 - Procédure avec 40% de camions de calibration et décalage des pics testé sur un ensemble de combinaisons

- Procédure avec décalage pic par pic :

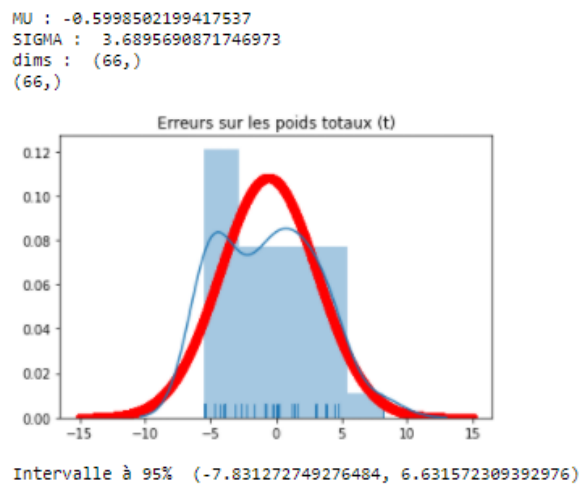


Fig.57 - Procédure avec 40% de camions de calibration et décalage pic par pic

Finalement, pour observer les meilleurs résultats, une fonction interpolée bien choisie (calculée avec les camions de calibration et quelques camions de trafic roulant à grande vitesse) a été utilisée. Un décalage pic par pic a été appliqué.

Voici les résultats (donc considérés comme les meilleurs) :

MU : 0.7879672243192921
 SIGMA : 1.5069335652986349
 dims : (24,)
 (24,)

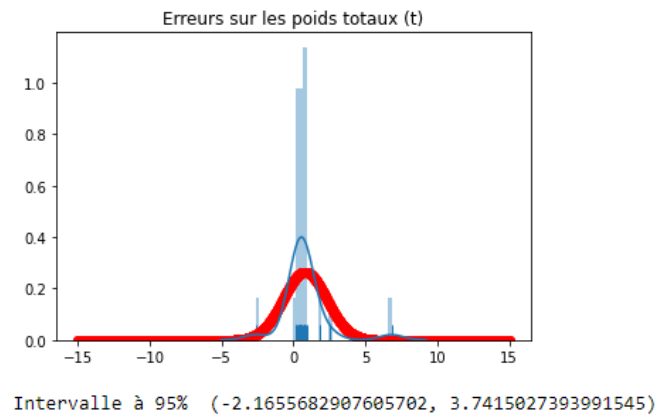


Fig.58- Procédure avec une ligne d'influence donnée (calculée sur un ensemble de PL bien choisi)

La moyenne des erreurs totales redressées (valeurs absolues) est de **1.05 tonne** (erreur moyenne de **2,5%**). (Poids moyen de l'ensemble de tous les camions : 41.92t).

- Erreur totale absolue moyenne (en tonne) sur l'essieu 1 : **1.05 t (15%)**
- Erreur totale absolue moyenne (en tonne) sur l'essieu 2 : **2.02 t (17%)**
- Erreur totale absolue moyenne (en tonne) sur l'essieu 3 : **2.26 t (26%)**
- Erreur totale absolue moyenne (en tonne) sur l'essieu 4 : **1.02 t (13%)**
- Erreur totale absolue moyenne (en tonne) sur l'essieu 5 : **0.91 t (12%)**

Evaluation des résultats sur des données simulées

Pour s'assurer que la procédure est robuste, elle a été réalisée sur des données simulées (générées).

Pour la ligne d'influence, le produit d'un cosinus et d'une gaussienne a été utilisé. (Voir Fig.59)

Elle a ensuite été pondérée par un coefficient tiré des lignes d'influence calculées par expérimentation pour que la suite des éléments générés soient utilisables.

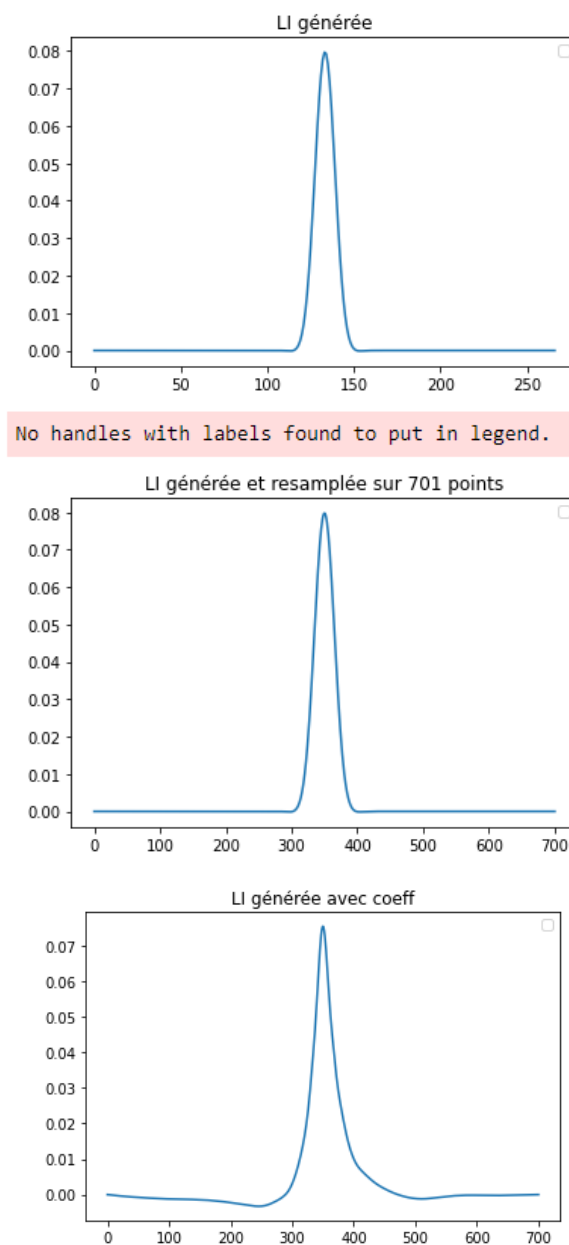


Fig.59 - Ligne d'influence générée (sans et avec coefficient)

Pour la configuration des essieux (poids par essieu), une distribution normale a été utilisée. (Voir Fig.60)

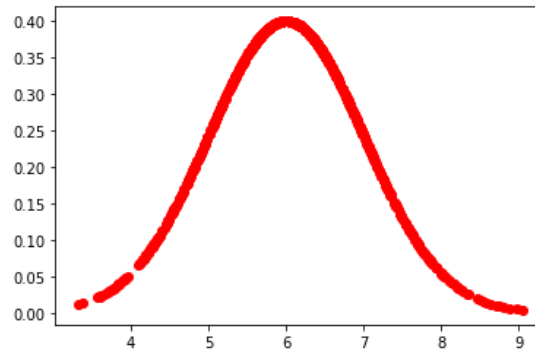


Fig.60 - Distribution normale utilisée pour générer la configuration des essieux

Cela fait, une pondération a été ajoutée selon l'emplacement de l'essieu pour que coller au mieux avec la réalité (premiers essieux moins lourds que le second etc).

Le tout devait correspondre à un poids total fixé en paramètre. (Voir Fig.61)

`[3.94682248 10.89253811 6.78084414 8.74727158 10.56943952]`

Fig.61 - Exemple de poids par essieu générés

Pour l'emplacement des essieux (et donc des pics), les moyennes et écarts-type de ceux des PL à notre disposition ont été utilisés pour générer aléatoirement (distribution normale) des emplacements.

Pour générer le signal, il suffisait alors de reprendre le modèle B-WIM pour obtenir un signal ressemblant à ceux observés. (Voir Fig.62)

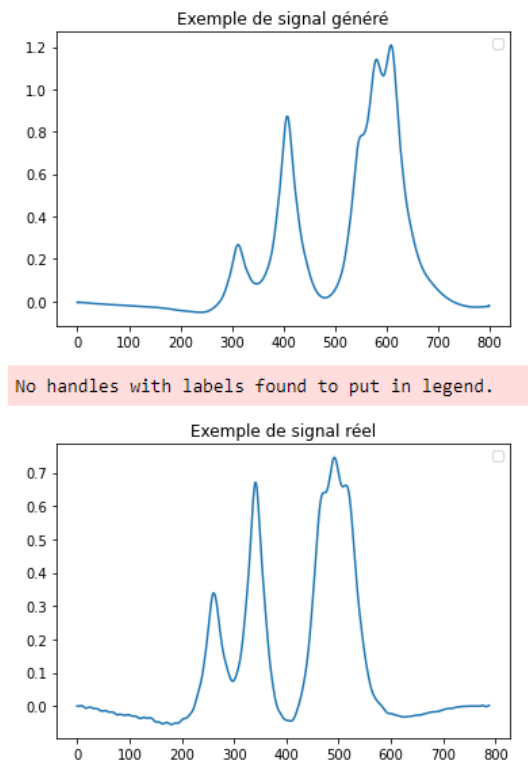


Fig.62 - Comparaison d'un signal observé et généré

Une fois que l'ensemble des attributs décrivant un camion existait, nous avons créé un nouveau `namedTuple` avec des attributs générés et relancer la procédure décrite précédemment. (Voir Fig.63)

```
MU : 0.004422653566077487
SIGMA : 0.006205839404011655
dims : (531,)
(531,)
```

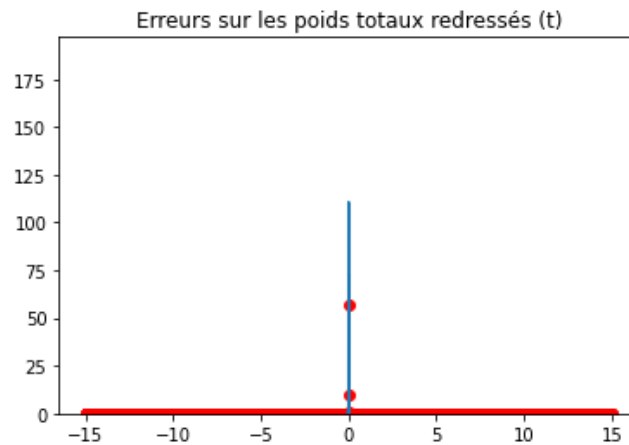


Fig.63 - Résultats de la procédure sur les données simulées

Pour observer l'effet du niveau de bruit d'un signal sur les résultats, nous avons ensuite ajouté du bruit à nos signaux (à différents niveaux) pour observer l'évolution des résultats. (Voir Fig.64 et 65)

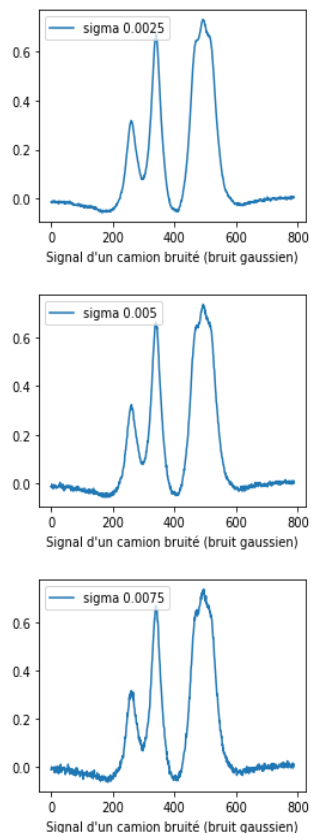


Fig.64 - Signaux générés avec différents niveaux de bruit gaussien (sigma)

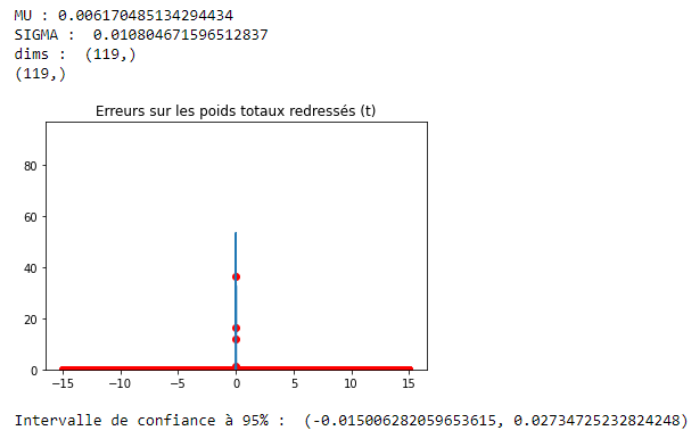


Fig.65 – Résultats de la procédure sur les données simulées avec un bruit gaussien sur les signaux tel que l'écart-type=0.005

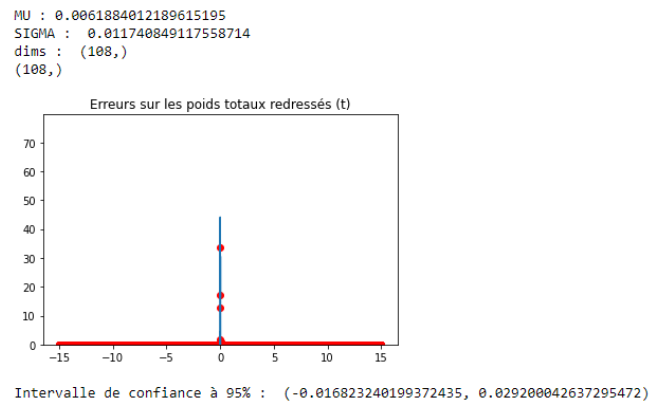


Fig.66 – Résultats de la procédure sur les données simulées avec un bruit gaussien sur les signaux tel que l'écart-type=0.02

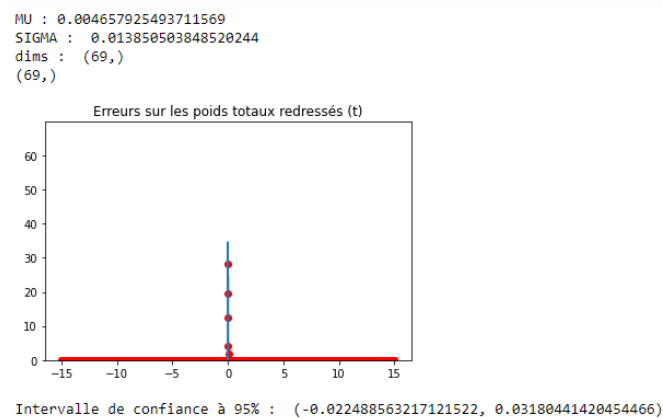


Fig.67 – Résultats de la procédure sur les données simulées avec un bruit gaussien sur les signaux tel que l'écart-type=0.03

Finalement, les intervalles de confiance à 95% s'élargissent bien proportionnellement au niveau de bruit des signaux.

Détermination des solutions de l'équation dynamique du pont par *physic informed neural network* (PINN)

Principe

Finalement, la dernière tâche de ce stage fût de tenter de résoudre l'équation différentielle décrivant l'équation dynamique du pont (qui correspond à la ligne d'influence lorsqu'elle est évaluée en son milieu).

Pour cela, une méthode relativement récente a été employée : le *physic informed deep learning*. Son principe réside dans l'évaluation des solutions de l'équation différentielle par un réseau de neurones.

Pour rappel, l'équation différentielle à résoudre s'écrit sous la forme :

$$\begin{aligned} EI \frac{\partial^4 u}{\partial x^4}(x, t) + \rho A \frac{\partial^2 u}{\partial t^2}(x, t) &= \delta(x), \\ 0 &\leq x \leq L \\ 0 &\leq t \leq \frac{L}{v} \\ u(0, t) &= u(L, t) = 0 \\ \frac{\partial^2 u}{\partial t^2}(0, t) &= \frac{\partial^2 u}{\partial t^2}(L, t) = 0, \end{aligned} \tag{3}$$

où:

- x est l'abscisse curviligne le long de la poutre,
- t le temps,
- $u(x, t)$ est le déplacement vertical de la poutre en x et t ,
- $\delta(x)$ est le dirac en x .

La ligne d'influence qui nous intéresse est donc celle du déplacement à mi-portée, soit $u(x)(\frac{L}{2}, t)$.

Fig.68- Équation dynamique du pont

Avec :

- **E** : Module d'Young
- **p** : masse volumique de la poutre
- **A** : surface de la poutre
- **I** : Inertie de la poutre
- **L** : Longueur de la poutre

Résolution avec SCIANN

Nous avons dans un premier temps utiliser SCIANN, une bibliothèque développée par le MIT sur la base de **Tensorflow** permettant de résoudre ce genre de problème.

Le problème rencontré résidait dans l'utilisation d'une fonction Dirac unitaire ($\delta(x)$) difficilement intégrable avec cette librairie.

Les résultats sont toutefois accessibles sur le Notebook N18.

Résolution avec DeepXDE

DeepXDE, une autre bibliothèque **opensource**, a alors été utilisée, permettant plus de modifications de la part de l'utilisateur (bien que plus compliquée à appréhender).

Après avoir défini l'équation à résoudre, il a fallu décrire la forme des solutions de cette équation comme un réseau de neurones. (Voir Fig.69)

```
net = dde.maps.FNN([2] + [32]*2 + [64] + [32] + [1], "tanh", "Glorot normal")
```

Fig.69 - Neural network représentant la solution de l'équation différentielle

Il s'agit d'un réseau classique feedforward dont les couches sont représentées ci-dessus. La fonction d'activation utilisée est tanh (notamment pour l'amplitude de sa dérivée, permettant d'éviter l'évanescence du gradient dans l'algorithme de rétropropagation. (Voir Fig.70) D'autant plus que les valeurs manipulées sont de l'ordre du milliè.

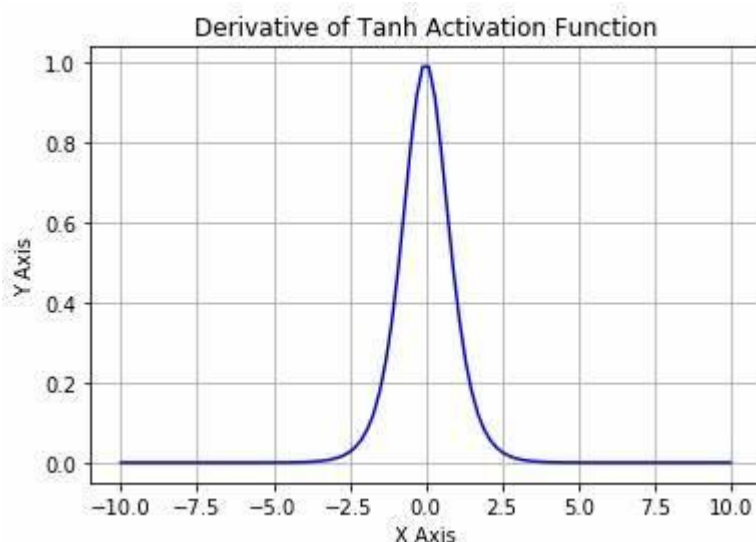


Fig.70 - Dérivée de tanh

En comparaison, nous pourrions observer la dérivée de sigmoid, une autre fonction d'activation connue (généralement utilisée pour des probas du fait de son intervalle d'arrivée). (Voir Fig.71)

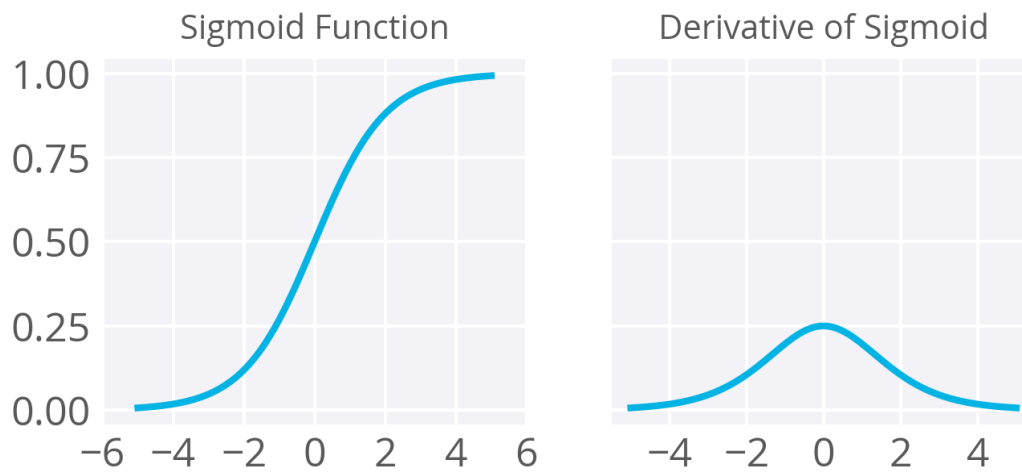


Fig.71 - Dérivée de sigmoid

L'algorithme de descente de gradient utilisé est Adam.

Le détail des paramètres (intervalles de définition etc) sont disponibles sur le Notebook.

Voici différents résultats obtenus après divers entraînements :

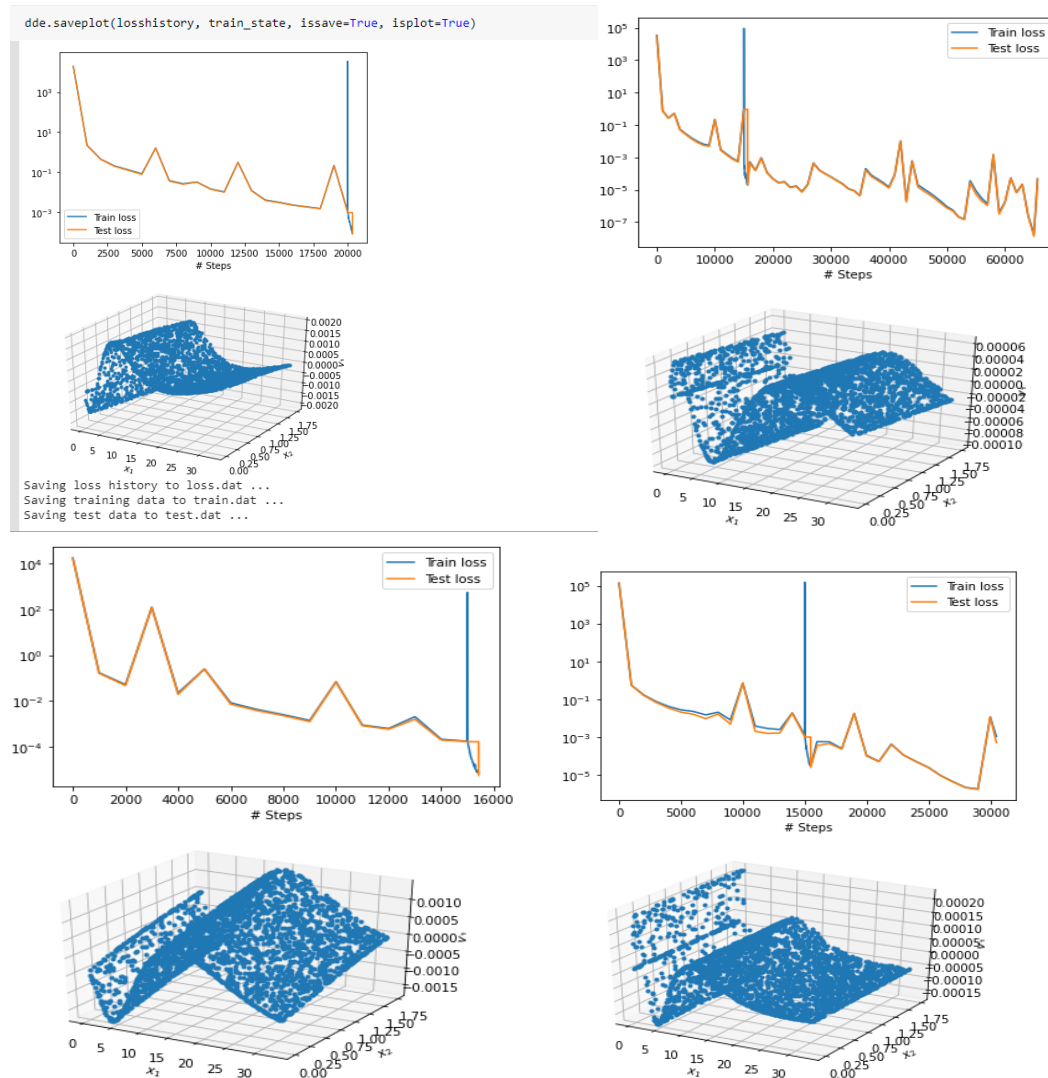


Fig.66 - Résultats de la résolution de l'équation différentielle par DeepXde

Globalement, la forme d'une ligne d'influence peut être reconnue. Cependant, selon la précision (perte de plus en plus basse), les résultats varient.

Ils sont donc difficilement utilisables à ce stade.

Ces calculs ont été faits avec les caractéristiques du pont de Senlis.

Une idée aurait été de résoudre cette équation par des méthodes classiques de résolution d'équation différentielle. Cependant, à cause des dérivées intervenantes (4^{eme} dans l'espace et 2nd dans le temps), il est difficile de trouver une bibliothèque pouvant convenir.

Le temps manquant, le sujet n'a pas pu être plus creusé.

Bilan d'expérience technique, organisationnel et humain

L'ensemble du stage a nécessité l'utilisation de diverses librairies indispensables à la pratique de la data science et de l'intelligence artificielle et m'a rompu à leur utilisation.

Plus précisément, d'un point de vue technique, j'ai pu perfectionner ma connaissance de **NumPy**, **Scipy**, **Seaborn**, **Matplotlib** ou encore **Sklearn** et plus globalement de la syntaxe **python** (utilisation de *list comprehension*, de *boucle enumerate* et création d'une bibliothèque).

J'ai également découvert une nouvelle partie de l'intelligence artificielle intéressante et complexe : le *physic informed deep learning*.

A la partie technique s'est mêlée une grande partie scientifique, basée sur le traitement du signal, l'algèbre, l'analytique, et un ensemble de notions issues des domaines mathématiques et physiques.

Ce mélange de savoir et de technique m'a permis d'élever mon niveau dans ces domaines qui font partie intégrante mon cursus d'ingénieur.

Mes tuteurs et tutrice Mme Franziska SCHMIDT, Mr Jean-François BERCHER et Mr Giovanni CHIERCHIA m'ont remarquablement bien accompagné tout au long du stage et ont pris le temps de m'expliquer des notions parfois complexes pour que ce stage se déroule au mieux. **Je les en remercie.**

A l'issue de ce stage, je me sens bien plus à l'aise en **python**, et dans l'ensemble des domaines cités précédemment.

Il faut également noter l'importance du sujet, qui, grâce à son caractère captivant, a rendu l'acquisition de ces connaissances bien plus simple et fluide.

En parallèle des côtés techniques et scientifiques, j'ai pu mettre en pratique une organisation professionnelle pour la première fois, notamment avec la mise en place de réunions hebdomadaires (voire plus fréquentes), la rédaction de comptes-rendus à l'issue de ces réunions, l'utilisation de **Git** et **GitLab** pour le partage de ressources et l'adaptabilité nécessaire pour réaliser ce stage dans de bonnes conditions malgré une situation sanitaire compliquée (Covid-19) impliquant sa réalisation en télétravail.

Je regrette de n'avoir pu travailler sur place à l'IFSTTAR pour faire de nouvelles rencontres et enrichir ma culture scientifique. Ces regrets sont largement compensés par l'ensemble des éléments cités ci-dessus.

Je profite également de cette partie pour remercier, en plus de mes tuteurs et tutrice, plus généralement l'IFSTTAR de m'avoir fait confiance et confié cette mission des plus intéressantes.

Annexes

Annexe 1 - Bibliographie

L'ensemble des documents scientifiques utilisés pour me former sont disponibles dans le dossier 'papers'.

J'ai également utilisé à de nombreuses reprises **Stackoverflow**, **Wikipédia** et les **documentations officielles** de bibliothèques.

Ressource technique utilisée	Lien vers la ressource
 python	https://www.python.org/
 NumPy	https://numpy.org/
 SciPy	https://www.scipy.org/
 scikit learn	https://scikit-learn.org/stable/
Seaborn	https://seaborn.pydata.org/
 matplotlib	https://matplotlib.org/
 git	https://git-scm.com/
 GitLab	https://git.esiee.fr/
 TensorFlow	https://www.tensorflow.org/

Annexe 2 - DataFrame des camions de calibration

	name	time	speed	signals	peaks	weights
0	2015-09-29-09-28-46-125	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.999990	[-0.0132997, -0.0133036, -0.0132623, -0.013174...	[260, 340, 493]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
1	2015-09-29-09-56-08-457	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.380992	[0.000226915, -8.02875e-05, -0.000158966, -1.4...	[261, 341, 472, 492, 514]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
2	2015-09-29-10-28-52-687	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.272699	[-0.00589252, -0.00536799, -0.00480485, -0.004...	[262, 342, 467, 494, 521]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
3	2015-09-29-11-08-45-998	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.630769	[0.00175452, 0.00153422, 0.00130749, 0.0011000...	[317, 397, 522, 549, 575]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
4	2015-09-29-11-50-31-328	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.000029	[-0.00212467, -0.00248653, -0.00277889, -0.002...	[263, 342, 471, 494, 518]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
5	2015-09-29-12-17-33-320	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	21.633769	[-0.0144981, -0.0136787, -0.0132154, -0.013189...	[263, 347, 480, 508, 536]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
6	2015-09-29-13-33-05-960	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.630779	[-0.0215423, -0.0214133, -0.0212388, -0.021041...	[265, 344, 470, 496, 522]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
7	2015-09-29-14-00-38-369	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	18.731689	[-0.0243554, -0.0245404, -0.0246975, -0.024829...	[266, 368, 528, 561, 593]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
8	2015-09-29-14-28-15-093	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	18.963019	[-0.0254103, -0.0254015, -0.0250018, -0.024329...	[266, 366, 526, 558, 590]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
9	2015-09-29-14-55-32-562	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	18.505997	[-0.020827, -0.0209844, -0.0211789, -0.0214018...	[266, 367, 527, 560, 593]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
10	2015-09-29-15-23-14-921	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	21.333342	[-0.0167387, -0.0168498, -0.0170026, -0.017174...	[265, 354, 492, 521, 550]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]
11	2015-09-29-15-52-01-990	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	18.962942	[-0.0185622, -0.0196713, -0.0207321, -0.021570...	[266, 367, 527, 559, 591]	[6.16068, 10.8499, 9.08406, 6.98472, 7.0632]

Annexe 3 - DataFrame des camions de trafic

	name	time	speed	signals	peaks	weights
0	PL10_2015-10-01-14-58-50-212	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.630779	[-0.0191894, -0.0187526, -0.0184853, -0.018449...	[276, 351, 472, 497, 522]	[6.75, 12.45, 8.0, 8.1, 8.0]
1	PL11_2015-10-01-15-00-57-826	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.999981	[-0.00928965, -0.00987199, -0.0105873, -0.0113...	[276, 355, 469, 492, 517]	[7.15, 11.4, 8.05, 8.0, 8.45]
2	PL12_2015-10-01-15-10-30-906	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.630816	[-0.0229519, -0.0233275, -0.0233024, -0.022886...	[277, 354, 478, 503, 529]	[7.4, 10.8, 8.8, 9.05, 8.75]
3	PL13_2015-10-01-15-19-48-718	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	25.180255	[-0.0206496, -0.0207517, -0.0208063, -0.020829...	[277, 350, 465, 489, 512]	[8.15, 12.2, 7.3, 8.3, 8.8]
4	PL14_2015-10-01-15-34-03-982	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.381001	[-0.0116909, -0.0108816, -0.010028, -0.009303...	[275, 358, 488, 508, 531]	[6.2, 5.05, 7.95, 7.65, 7.4, 7.65]
5	PL15_2015-10-01-15-36-13-519	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	22.260846	[-0.0136828, -0.0139356, -0.0142175, -0.014500...	[272, 399, 425, 518, 568, 622]	[5.35, 8.25, 8.1, 7.75, 7.8, 7.6]
6	PL1_2015-10-01-09-45-52-576	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	22.260904	[-0.014285, -0.0140879, -0.013944, -0.013884, ...	[275, 361, 473, 501, 529]	[7.35, 11.25, 8.2, 8.1, 8.25]
7	PL2_2015-10-01-09-55-36-371	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.000000	[-0.0100142, -0.00989681, -0.0097338, -0.00955...	[274, 353, 477, 503, 529]	[7.65, 14.1, 6.05, 6.2, 6.3]
8	PL3_2015-10-01-10-22-34-654	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.000029	[-0.014935, -0.0149689, -0.015053, -0.0152061,...	[274, 349, 470, 495, 519]	[7.4, 11.6, 8.35, 8.5, 8.5]
9	PL4_2015-10-01-10-35-43-279	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.380962	[-0.0168291, -0.0172036, -0.017665, -0.0181426...	[276, 354, 462, 487, 513]	[7.6, 12.35, 7.8, 7.7, 7.8]
10	PL5_2015-10-01-10-59-39-060	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.272746	[0.00408244, 0.00418735, 0.00422049, 0.0041573...	[274, 353, 452, 478, 505]	[7.35, 10.5, 9.3, 9.7, 9.45]
11	PL6_2015-10-01-11-05-36-548	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.000048	[-0.015224, -0.0151083, -0.0148442, -0.0144753...	[276, 354, 478, 503, 529]	[7.3, 14.3, 6.8, 6.6, 6.7]
12	PL7_2015-10-01-11-14-06-712	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	24.380893	[-0.0128941, -0.0127466, -0.012642, -0.0126112...	[275, 350, 469, 493, 518]	[7.05, 13.95, 7.25, 7.55, 7.9]
13	PL8_2015-10-01-11-42-22-222	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.630750	[-0.0150944, -0.0153917, -0.0159598, -0.016714...	[271, 344, 468, 493, 519]	[6.4, 14.05, 6.85, 6.75, 6.65]
14	PL9_2015-10-01-14-42-22-908	[0.0, 0.00195312, 0.00390625, 0.00585938, 0.00...	23.272718	[-0.0253963, -0.0255132, -0.0254464, -0.025214...	[277, 357, 474, 502, 528]	[8.2, 12.0, 8.35, 8.4, 8.35]