Calibration de modèle pour le véhicule autonome

Clara CARLIER

15 septembre 2020

Plan

Introduction

Présentation du sujet

- **Expansion** et **évolution** du monte de l'automobile : véhicules autonomes
  - grande quantité de capteurs embarqués
- **Fiabilité** et **validation** des véhicules : réglementations multiples et strictes
  - réalisation d'un grand nombre de tests
- Logiciel développé par Renault pour **simuler les données** nécessaires à la validation

$\Longrightarrow$ CRÉATION DE BASES DE DONNÉES SIMULÉES GIGANTESQUES

Encore faut-il prouver que les simulations retranscrivent bien la réalité...

Les données

Un scénario et un contexte spécifique

- Véhicule lancé face à un obstacle à une vitesse donnée :

  10, 20, 30, 40 ou 50 km/h
- Intérêt : vérifier le comportement de la voiture face à un arrêt d'urgence

Les données :

- 10 expériences **réelles** : 2 par vitesse initiale
- 500 expériences **simulées** : similaires aux réelles, 100 par vitesse initiale
- 2 **paramètres** pour chaque expérience simulée : `EgoSpeed` et `Overlap`
  - *inconnus pour les expériences réelles*

Paramètre `EgoSpeed`

Vitesse initiale de l'expérience : répartition clairement uniforme

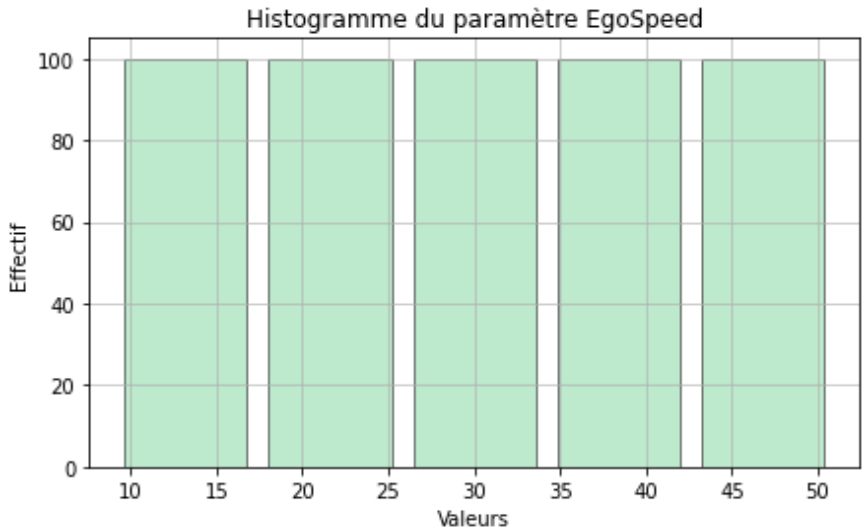


FIGURE – Histogramme du paramètre `EgoSpeed`

Paramètre `Overlap`

Décalage de la voiture selon l'axe du milieu :

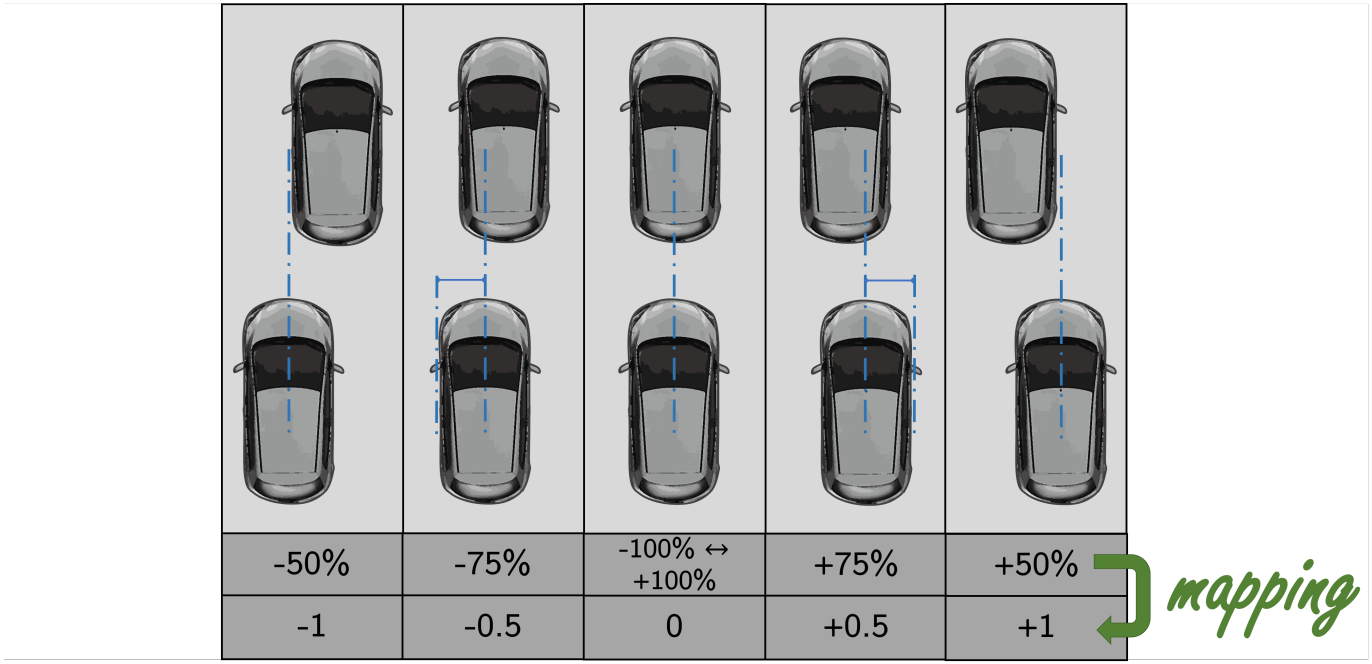
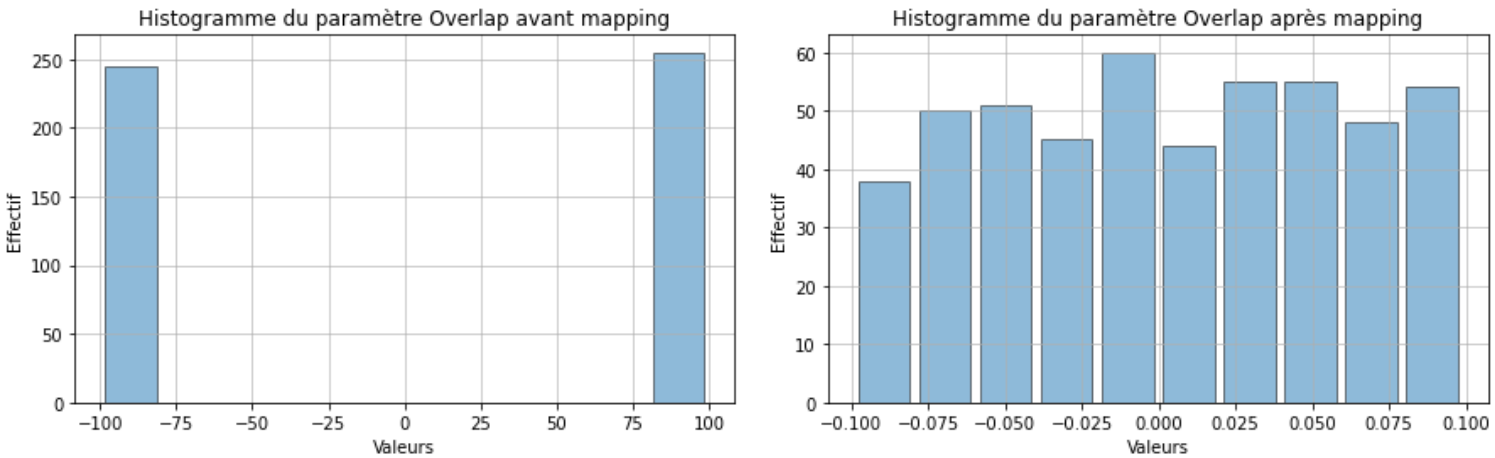


FIGURE – Présentation du paramètre latéral `Overlap`

Paramètre `Overlap`

Répartition à peu près uniforme



(a) Avant (b) Après

FIGURE – Histogrammes représentant le paramètre `Overlap` avant et après mapping

Données brutes

- la position (non représentée ici), la vitesse et l'accélération du véhicule
- la distance restante avec l'obstacle (Bumper Distance) et le temps avant collision (TTC)
- **Mauvaise synchronisation**, accélérations trop **bruitées** et valeurs **aberrantes** pour le TTC

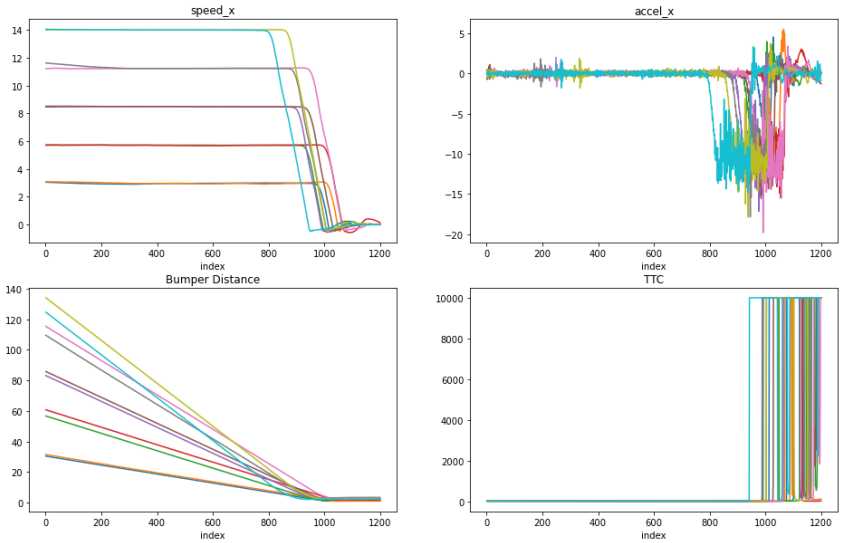


FIGURE – Représentation graphique des variables d'intérêt des données réelles avant modification et synchronisation

Données finales

- Synchronisation des données selon l'accélération
- Filtrage de l'accélération avec le filtre de Butterworth [1]
- Découpage du TTC [2] sur la zone qui nous intéresse
- Découpage de la fin des expériences

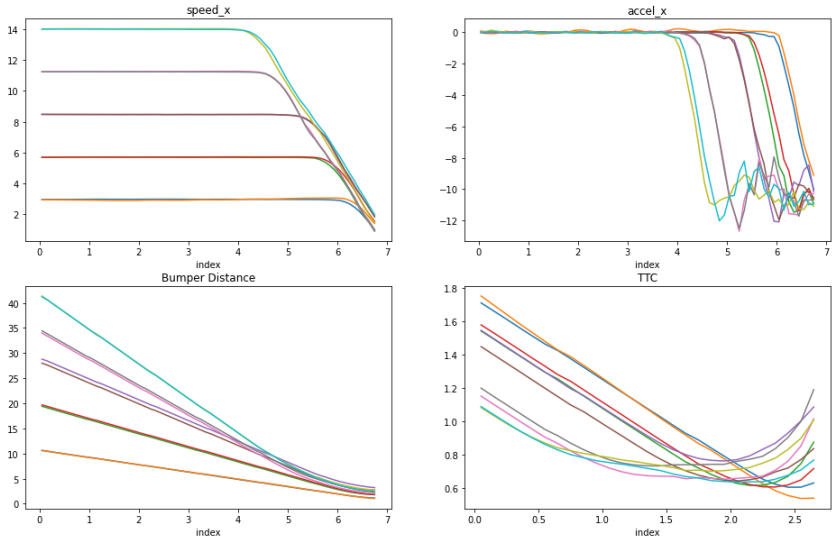


FIGURE – Représentation graphique des variables d'intérêt des données réelles après modification et synchronisation

1. utilisé par les organismes de certification
2. TTC = Bumper Distance / speed_x

Données brutes

- Courbes **non similaires** à celles des données réelles
  $\rightarrow$ conserver uniquement la fin des simulations
- La synchronisation semble correcte
- Accélération pas trop bruitée
- Valeurs aberrantes pour le TTC également

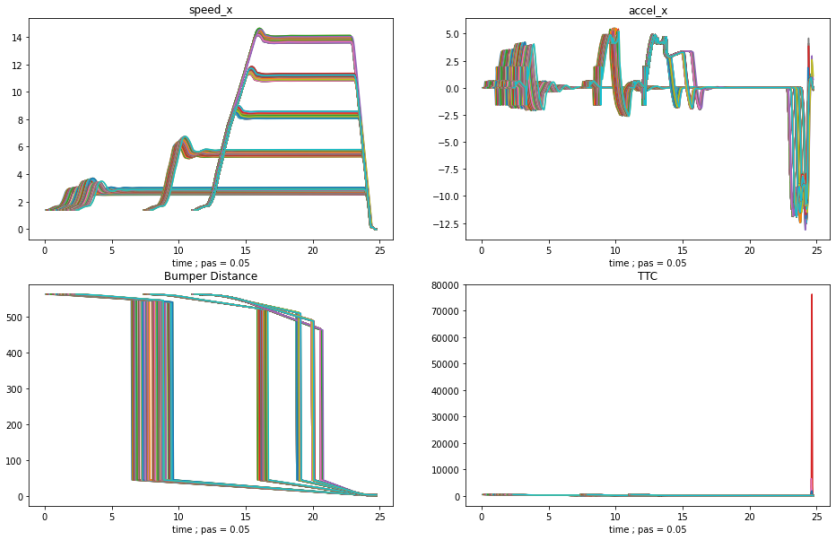


FIGURE – Représentation graphique des variables d'intérêt des données simulées avant modification et synchronisation

Données finales

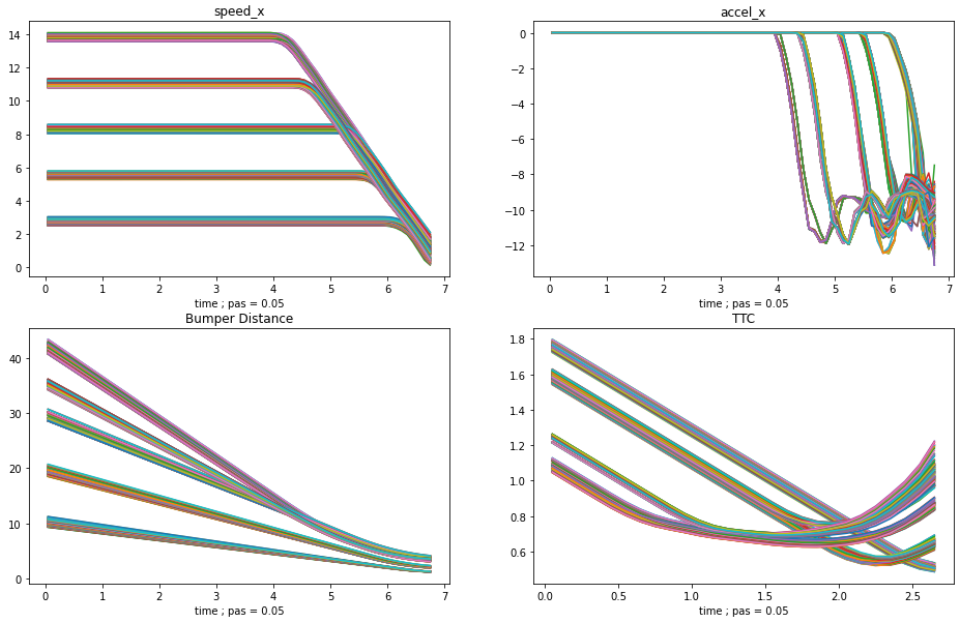


FIGURE – Représentation graphique des variables d'intérêt des données simulées après modification et synchronisation

- Les courbes sont maintenant **très similaires** à celles des données réelles.

Superposition des données réelles et simulées

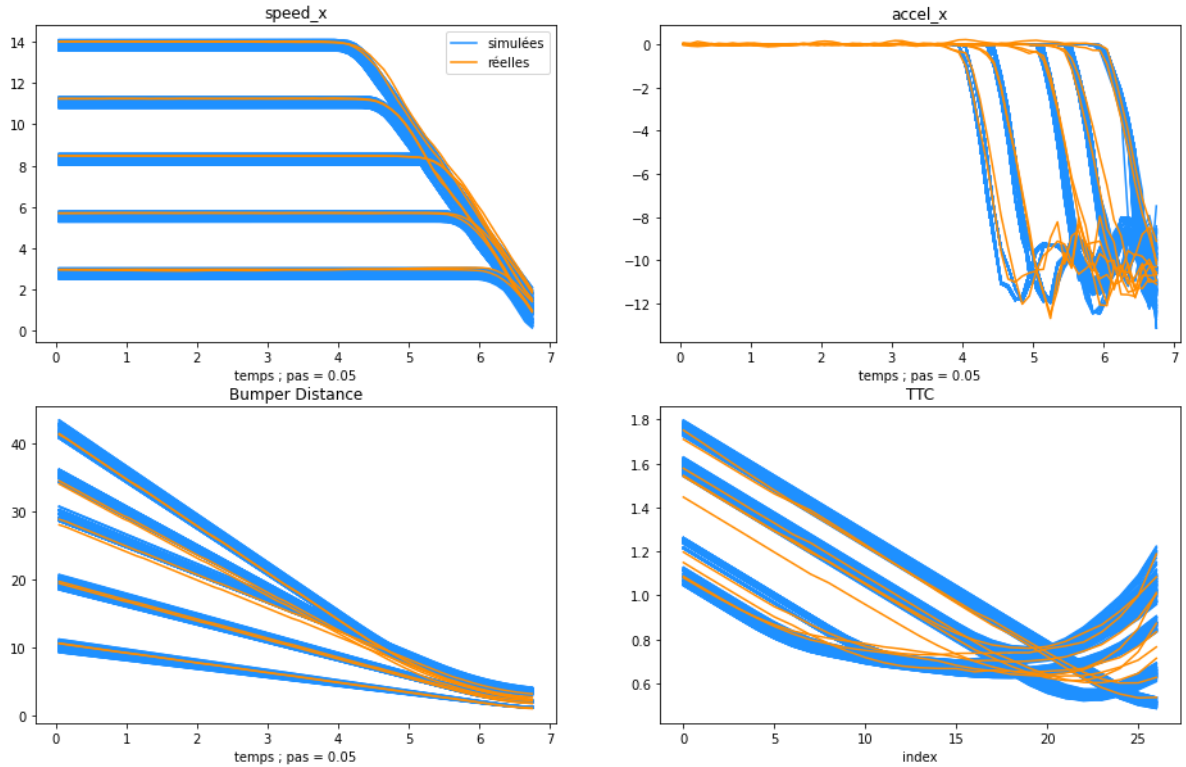


FIGURE – Représentation graphique des données réelles et simulées après modification et synchronisation

Modèle de substitution

Objectif

- Permet d'estimer la **vraisemblance** du modèle lors de l'étape d'inférence
- Construire un modèle de substitution **à partir des simulations**
- Prédire la position, la vitesse, l'accélération, la bumper distance et le ttc à partir d'`EgoSpeed` et d'`Overlap`

  - Utilisation de forêts aléatoires puis de réseaux de neurones

Quantités d'intérêt

On souhaite **minimiser** :

- Erreur quadratique moyenne globale
- Erreur quadratique moyenne de la position au dernier pas de temps
- La quantité Q définie par :

$$Q = 0.5 \times \|s\|_{q_1} + \|a\|_{q_2} \tag{1}$$

avec : s le vecteur vitesse et a le vecteur accélération

$$\|x\|_{q_1} = \|x\|_1 + \|x\|_2 + 0.5 \times \|x\|_{+\infty}$$
$$\|y\|_{q_2} = \|y\|_1 + \|y\|_2 + \|y\|_{+\infty}$$

Forêts aléatoires : choix du paramètre B

- On a fait varier **le nombre d'arbres** B dans la forêt :

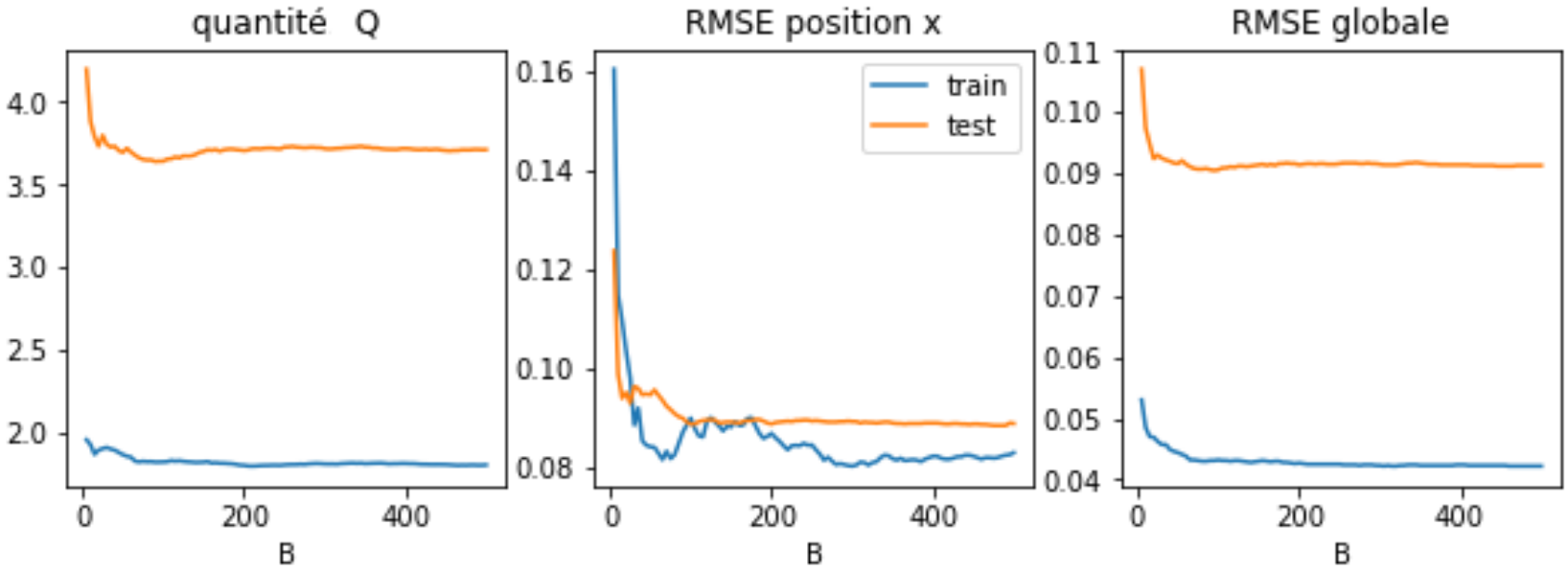


FIGURE – Évolution des quantités d'intérêt pour différents paramètres B

Forêts aléatoires : résultats obtenus

- Résultats obtenus pour $B = 91$:

données	RMSE position x	quantité Q	RMSE globale
train	0.0874	1.8151	0.0431
test	0.0893	3.6361	0.0903

TABLE – Quantités d'intérêts obtenues à l'aide d'une forêt aléatoire

Comparaison de deux réseaux de neurones

- Résultats obtenus avec un **premier réseau** de 300 neurones par couche :

nb. epochs	données	RMSE pos. x	Q	RMSE glob.
300	*train*	1.2274	**31.5462**	0.6803
	test	1.2394	**34.3152**	0.7056
400	*train*	**0.7068**	35.0264	**0.5097**
	test	**0.7062**	37.1387	**0.5394**

- Résultats obtenus avec un **deuxième réseau** avec 500 epochs :

nb. epochs	données	RMSE pos. x	Q	RMSE glob.
500 epochs	*train*	**0.6958**	**14.5039**	**0.3796**
	test	**0.7196**	**14.7249**	**0.3981**

Conclusion sur le modèle de substitution

- Résultats **peu satisfaisants** avec les réseaux de neurones
- Avec les **forêts aléatoires** : meilleures valeurs d'un facteur 10

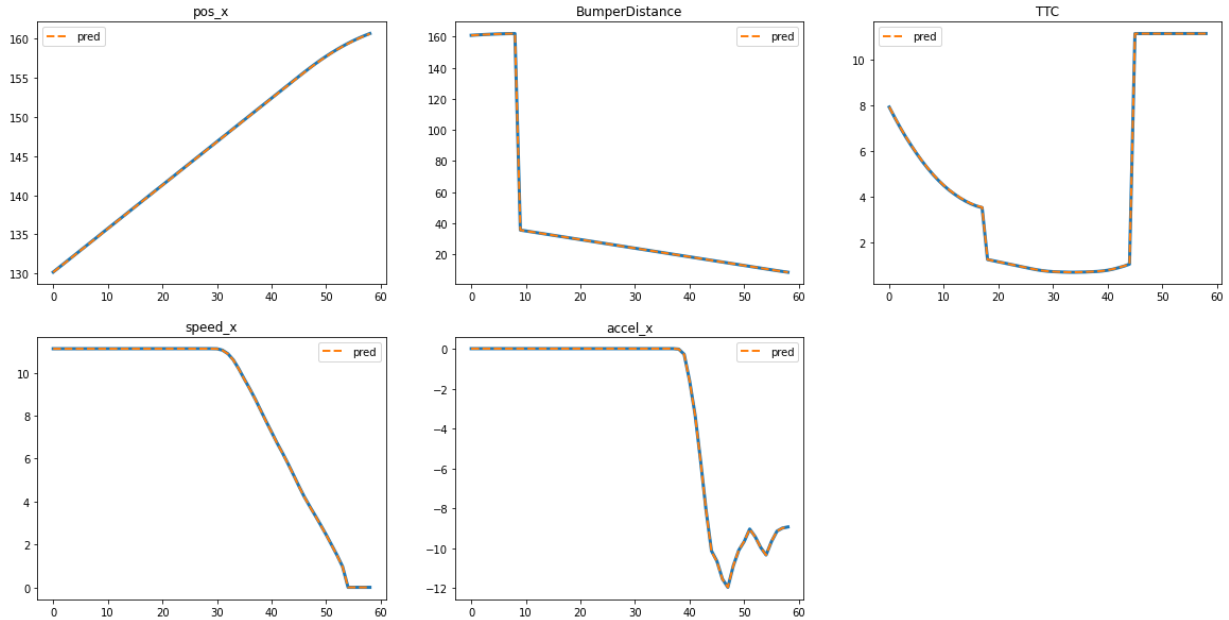


FIGURE – Illustration des prédictions réalisées avec le modèle de forêt aléatoire

Inférence des paramètres

Objectif, mise en œuvre et prior (1/2)

- Construire un modèle à partir des **données réelles** (position, vitesse, accélération, bumper distance et ttc) afin d'inférer les paramètres `EgoSpeed` et `Overlap` qui sont **inconnus**

- Par le théorème de Bayes, pour une réalisation y, le **posterior** est donné par :

$$P(\theta|Y = y) \propto P(Y = y|\theta)P(\theta) \tag{2}$$

  - $\theta = (\texttt{EgoSpeed}, \texttt{Overlap})^T$ et Y les données expérimentales
  - $P(Y = y|\theta)$ la vraisemblance obtenue par le modèle de substitution
  - $P(\theta)$ le prior, on prend des lois normales

- Appliquer la méthode expérience par expérience en inférant les paramètres deux à deux :
  Prior : $\texttt{EgoSpeed} \sim \mathcal{N}(30, \sigma_{es})$ et $\texttt{Overlap} \sim \mathcal{N}(0, \sigma_o)$
- Appliquer la méthode de manière globale en inférant les 20 paramètres :
  Prior : $\texttt{EgoSpeed} \sim \mathcal{N}\big(\{10, 20, 30, 40, 50\}, \sigma_{es}\big)$
  $\texttt{Overlap} \sim \mathcal{N}\big(0, \sigma_o\big)$

Évaluation de l'inférence et valeurs naïves

- La vraie valeur des paramètres **inconnue** : comment vérifier que l'inférence est correcte ou non ?
  $\Longrightarrow$ Chercher la simulation **la plus ressemblante** à chaque expérience réelle puis comparer les paramètres inférés à ceux sélectionnés

- Choix de la simulation la plus proche en **minimisant** :

$$\min_{k=1,\dots,500} \left(\frac{1}{n} \sum_{i=1}^{n} \left((a_{x,k}^{simu})_i - (a_{x,l}^{real})_i \right)^2 \right)^{1/2} \tag{3}$$

  - $a_{x,k}^{simu}$ vecteur correspondant à l'accélération selon le premier axe de la $k^{\text{ième}}$ expérience simulée
  - de même pour $a_{x,l}^{real}$ mais de la $l^{\text{ième}}$ expérience réelle

- Avec **valeurs naïves** $(10, 10, 20, 20, 30, 30, 40, 40, 50, 50)$ pour `EgoSpeed` et $(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ pour `Overlap` : **0.6116** et **0.0661**

Méthode ABC - Monte Carlo séquentiel : théorie

▶ **Avantage** : permet d'inférer le posterior bien que la vraisemblance soit difficile ou coûteuse à évaluer

Algorithme :

1. Échantillonner un paramètre θ^* selon le prior $P(\theta)$
2. Simuler une base de données y^* à l'aide d'une fonction qui associe à θ des données de la même dimension que les données observées y_0 (avec le modèle de substitution)
3. Comparer les données simulées y^* avec les données expérimentales y_0 en utilisant une distance d et un seuil de tolérance ε

▶ **Méthode ABC** : transforme itérativement le prior en posterior en propageant les paramètres échantillonnés à travers une série de distributions

Résultats obtenus avec le Monte Carlo séquentiel

- Résultats obtenus en inférant **expérience par expérience** :

EgoSpeed			Overlap			tps. d'exéc.
σ prior	RMSE	σ post.	σ prior	RMSE	σ post.	
0.5	**14.1176**	0.3446	0.03	0.0661	0.0302	37 min
0.05	14.2417	0.0499	0.01	**0.0660**	0.0100	29 min

- Résultats obtenus en inférant de **manière globale** :

EgoSpeed			Overlap			tps. d'exéc.
σ prior	RMSE	σ post.	σ prior	RMSE	σ post.	
0.05	**0.6116**	0.0499	0.05	**0.0654**	0.0499	3min 20sec

Conclusion

Conclusion

- Étape de **calibrage des données** : importante et longue
- Pour le modèle de substitution : très satisfaits des **forêts aléatoires**, réseaux de neurones un peu décevant mais...
- Pour l'inférence : **résultats plutôt positifs**, petit bémol sur les méthodes ABC

Perspectives et **améliorations** :

- Tester différents calibrages des données
- Passer plus de temps sur les réseaux de neurones pour améliorer la précision du modèle de substitution
- Inférence meilleure si le modèle de substitution est plus précis ?
- Inférence faite sans ajout d'erreur : permettrait de modéliser l'erreur du simulateur et ainsi réduire l'écart-type des posterior

Bilan : ce que j'ai fait

- Modification et synchronisation des données réelles et simulées
- Construction d'un modèle de substitution à l'aide de différentes méthodes déjà implémentées :
  - Forêts aléatoires (`scikit.learn`)
  - Réseau de neurones simple (`keras`)
  - Réseau de neurones convolutionnel un peu plus complexe (`keras`)
- Inférence des paramètres à l'aide de plusieurs méthodes déjà implémentées :
  - Étape de Metropolis-Hastings adaptatif (`PyMC3`)
  - Algorithme de Monte Carlo séquentiel (`PyMC3`)