

## Optimisation des algorithmes d'apprentissage automatique dans le contexte de véhicules intelligents

Ramarolahivonjitiana F.J.<sup>1</sup>, Randriamitantsoa P.A.<sup>2</sup>, Randriamitantsoa A.A.<sup>3</sup>

Laboratoire de Recherche Télécommunications, Automatique, Signal et Images (L.R-T.A.S.I)

Ecole Doctorale en Science et Technique de l'Ingénierie et de l'Innovation (ED-STII)

Ecole Supérieure Polytechnique Antananarivo

Université d'Antananarivo

BP 1500, Ankatso – Antananarivo 101 – Madagascar

<sup>1</sup>vonjilahitiana@yahoo.fr, <sup>2</sup>rpauguste@gmail.com, <sup>3</sup>andriau23@gmail.com

### Résumé

*Le contrôle et la maîtrise des objets connectés et autonomes deviennent des préoccupations sociétales et économiques majeures. Cela a incité les milieux universitaires et l'industrie à une intense activité dans le domaine de recherche de l'apprentissage automatique. Avec un véhicule intelligent, les algorithmes d'apprentissage automatique sont implémentés dans les sous-systèmes de la détection, de la perception et de la décision pour assurer la localisation, la reconnaissance d'objets, la planification de trajectoire, la prédiction d'action jusqu'à l'évitement d'obstacle. Dans le but ultime d'optimiser ces algorithmes, nous avons implémenté des fonctions de coût pour un modèle de prédiction d'une trajectoire souhaitée et la planification de mouvement. En plus, nous avons pu déterminer une fonction de coût d'une cyberattaque. Aussi, nous avons utilisé la technique de réduction de la dimensionnalité pour l'optimisation de la planification de mouvement afin de dresser une carte de collision.*

**Mots clés :** *Apprentissage automatique, fonction de coût, fonction de perte, réduction de la dimensionnalité, véhicules intelligents*

### Abstract

*Controlling and mastering connected and autonomous objects are becoming major societal and economic concerns. This has sparked intense activity in academia and industry in the area of machine learning research. With an intelligent vehicle, machine learning algorithms are implemented in the detection, perception and decision subsystems to ensure localization, object recognition, trajectory planning, action prediction until obstacle avoidance. With the ultimate goal of optimizing these algorithms, we implemented cost functions for a predictive model of a desired trajectory and motion planning. In addition, we were able to determine a cost function of a cyber-attack. Also, we used the dimensionality reduction technique for motion planning optimization to map the collision.*

**Keywords :** *Machine learning, Cost function, Lost function, Dimensionality reduction, Intelligent vehicles*

## 1. Introduction

L'intelligence des objets connectés a connu un grand progrès grâce à une combinaison de la demande du marché et du progrès technologique. L'exploitation des données disponibles est en train de devenir une préoccupation majeure pour la société et l'économie. Cette préoccupation a incité les milieux universitaires et l'industrie à se lancer dans une intense activité dans le domaine de l'intelligence artificielle. L'apprentissage automatique serait donc en pole position pour l'intelligibilité de ces données massives. Dans notre quotidien, avec l'avènement des objets connectés, l'apprentissage automatique est la base des systèmes de recommandation et des systèmes de prédiction. Ces algorithmes d'apprentissage automatique sont évalués selon leur capacité à classer ou prédire de manière correcte pour atteindre des objectifs spécifiques. Pour l'optimisation de ces systèmes, on parlera de la fonction de coût également appelée fonction de perte. Dans ce présent article, nous allons mettre en évidence la minimisation des pertes et les critères pour la sélection de modèle avec une application dans le contexte de véhicules intelligents et connectés. En premier lieu, nous allons définir les différentes fonctions de coût et démontrer les techniques telles que la réduction de la dimensionnalité, le changement de covariable et l'approximation des fonctions. Par la suite, nous allons voir l'architecture global d'un véhicule connecté et autonome et l'implémentation de l'apprentissage automatique.

## 2. Optimisation de l'apprentissage automatique

### 2.1. Fonction de coût

La *fonction de coût*, également appelée *fonction de perte*, est la fonction utilisée pour déterminer la perte. Le *coût* ou la *perte*

d'une prédiction  $y'$ , lorsque la valeur correcte est  $y$ , est une mesure de l'utilité relative de cette prédiction étant donné cette valeur correcte. Une fonction de perte courante utilisée avec l'apprentissage de la classification est la perte 0-1. La perte 0-1 attribue 0 à la perte pour une classification correcte et 1 pour une classification incorrecte. La classification sensible aux coûts attribue différents coûts à différentes formes de classification erronée. Une fonction de perte courante utilisée avec la régression est l'erreur quadratique. C'est le carré de la différence entre les valeurs prévues et réelles. [1]

Pour la mesure de succès, l'*erreur d'un classificateur* est définie comme étant la probabilité qu'il ne prédit pas l'étiquette correcte sur un point de données aléatoire généré par la distribution sous-jacente susmentionnée. Autrement dit, l'erreur de  $h$  est la probabilité de tirer une instance aléatoire  $x$ , selon la distribution  $D$ , telle que  $h(x)$  ne soit pas égale à  $f(x)$ . [2]

Formellement, étant donné un sous-ensemble de domaine  $A \subset X$ , la distribution de probabilité  $D$  attribue un nombre  $D(A)$  qui détermine la probabilité d'observer un point  $x \in A$ . Dans de nombreux cas, nous nous référons à  $A$  comme un événement et l'exprimons en utilisant une fonction  $\pi: X \rightarrow \{0,1\}$ , à savoir  $A = \{x \in X : \pi(x) = 1\}$ . Dans ce cas, nous utilisons également la notation  $\mathbb{P}_{x \sim D}[\pi(x)]$  pour exprimer  $D(A)$ .

On définit l'erreur d'une règle de prédiction,  $h: X \rightarrow Y$ , comme étant : [2]

$$\begin{aligned} \text{perte}_{D,f}(h) &\stackrel{\text{def}}{=} \mathbb{P}_{x \sim D}[h(x) \neq f(x)] & (1) \\ \text{perte}_{D,f}(h) &\stackrel{\text{def}}{=} D(\{x: h(x) \neq f(x)\}) \end{aligned}$$

### 2.2. Réduction de la dimensionnalité

La réduction de la dimensionnalité est un type d'apprentissage non supervisé. Le but est également de révéler une structure particulière des données, qui est de nature

différente de celle des regroupements. Par exemple, bien que les données puissent être représentées dans un espace de grande dimension, elles peuvent se trouver autour d'un sous-espace de dimension inférieure ou d'un collecteur. Ces méthodes sont très importantes dans l'apprentissage automatique pour des représentations compressées ou pour des raisons de réduction de calcul. L'apprentissage de la structure de dimension inférieure associée à un ensemble donné de données gagne en importance dans le contexte du traitement et de l'analyse du Big Data.

### 2.2.1. Utilisation d'une transformation linéaire

Soient  $x_1, \dots, x_m$   $m$  vecteurs dans  $\mathbb{R}^d$ . Nous aimerions réduire la dimensionnalité de ces vecteurs en utilisant une transformation linéaire. Une matrice  $W \in \mathbb{R}^{n,d}$ , où  $n < d$ , induit une application  $x \mapsto Wx$  où  $Wx \in \mathbb{R}^n$  est la représentation de dimensionnalité inférieure de  $x$ . Ensuite, une seconde matrice  $U \in \mathbb{R}^{d,n}$  peut être utilisée pour (approximativement) récupérer chaque vecteur original  $x$  à partir de sa version compressée. Autrement dit, pour un vecteur compressé  $y = Wx$ , où  $y$  est dans l'espace de faible dimension  $\mathbb{R}^n$ , nous pouvons construire  $\tilde{x} = Uy$ , de sorte que  $\tilde{x}$  est la version récupérée de  $x$  et réside dans l'espace de grande dimension d'origine  $\mathbb{R}^d$ .

### 2.2.2. Analyse en composantes principales ACP

En ACP, nous trouvons la matrice de compression  $W$  et la matrice de récupération  $U$  de sorte que la distance quadratique totale entre les vecteurs d'origine et récupérés soit minimale; à savoir, nous visons à résoudre le problème : [2][3]

$$\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^m \|x_i - UWx_i\|_2^2 \quad (2)$$

Pour résoudre ce problème, nous montrons d'abord que la solution optimale prend une forme spécifique.

Soit  $(U, W)$  une solution de l'équation (2). Alors les colonnes de  $U$  sont orthonormées (à savoir,  $U^T U$  est la matrice d'identité de  $\mathbb{R}^n$ ) et  $W = U^T$ .

Nous avons le théorème suivant : [2]

Soient  $x_1, \dots, x_m$  des vecteurs arbitraires dans  $\mathbb{R}^d$ , soit  $A = \sum_{i=1}^m x_i x_i^T$ , et soit  $u_1, \dots, u_n$   $n$  vecteurs propres de la matrice  $A$  correspondant aux  $n$  plus grandes valeurs propres de  $A$ . Ensuite, la solution au problème d'optimisation ACP donné dans l'équation  $\underset{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}}{\operatorname{argmin}} \sum_{i=1}^m \|x_i - UWx_i\|_2^2$  est de définir  $U$  comme étant la matrice dont les colonnes sont  $u_1, \dots, u_n$  et pour définir  $W = U^T$ .

### 2.2.3. Projections aléatoires

Cette section, nous montre que la réduction de la dimensionnalité en utilisant une transformation linéaire aléatoire conduit à un schéma de compression simple avec une distorsion étonnamment faible. La transformation  $x \mapsto Wx$ , lorsque  $W$  est une matrice aléatoire, est souvent appelée projection aléatoire. En particulier, nous proposons une variante d'un célèbre lemme dû à Johnson et Lindenstrauss, montrant que les projections aléatoires ne déforment pas trop les distances euclidiennes. [2].

Soient  $x_1, x_2$  deux vecteurs dans  $\mathbb{R}^d$ . Une matrice  $W$  ne déforme pas trop la distance entre  $x_1$  et  $x_2$  si le rapport :

$$\frac{\|Wx_1 - Wx_2\|}{\|x_1 - x_2\|} \text{ est proche de } 1 \quad (3)$$

En d'autres termes, les distances entre  $x_1$  et  $x_2$  avant et après la transformation sont presque les mêmes. Pour montrer que  $\|Wx_1 - Wx_2\|$  n'est pas trop éloigné de  $\|x_1 - x_2\|$ , il suffit de montrer que  $W$  ne déforme pas la norme du vecteur de

différence  $x = x_1 - x_2$ . Par conséquent, à partir de maintenant, nous nous concentrons sur le rapport  $\frac{\|W_x\|}{\|x\|}$ .

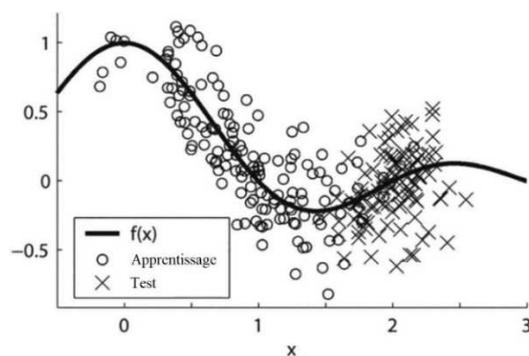
### 2.3. Changement de covariable

S'il n'y a pas de connexion entre les données d'apprentissage et les données de test, on n'apprendra rien sur les données de test à partir des échantillons d'apprentissage. Cela signifie qu'une hypothèse raisonnable est nécessaire pour relier les échantillons d'apprentissage aux données de test. Le changement de covariable est l'une de ces hypothèses [4].

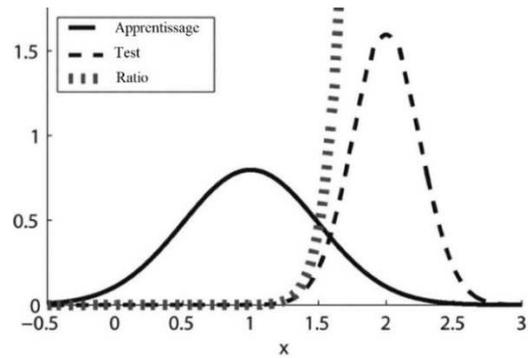
#### 2.3.1. Adaptation du changement de covariable

Considérons un problème de régression d'apprentissage d'une fonction  $f(x)$  à partir de ses échantillons  $\{(x_i^a, y_i^a)\}_{i=1}^{n_a}$ . Une fois qu'une bonne fonction d'approximation  $\hat{f}(x)$  est obtenue, nous pouvons prédire la valeur de sortie  $y^t$  à un point d'entrée de test invisible  $x^t$  par  $\hat{f}(x^t)$  [4].

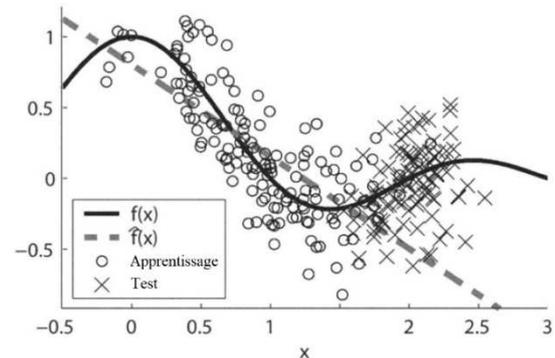
Dans l'exemple de régression illustré sur la **Figure 01 (a)**, les échantillons d'apprentissage sont situés sur le côté gauche du graphique et les échantillons de test sont répartis sur le côté droit. Il s'agit donc d'un problème d'extrapolation où les échantillons de test sont situés en dehors de la région d'apprentissage. Les densités de probabilité des points d'entrée d'apprentissage et de test,  $p_a(x)$  et  $p_t(x)$ , sont représentées sur la **Figure 01 (b)**.



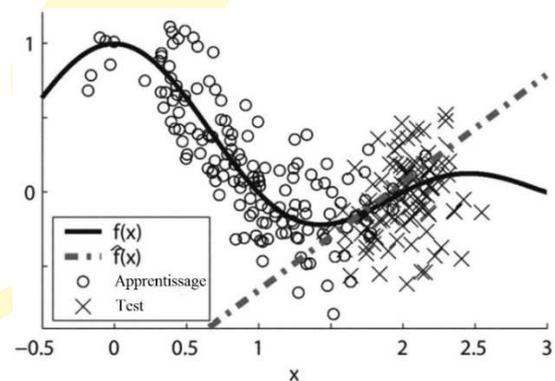
(a) Données d'apprentissage et de test



(b) Densités des données d'entrée



(c) Fonction apprise par les moindres carrés ordinaires



(d) Fonction apprise par les moindres carrés pondérés

**Figure 01** : Un exemple de régression avec changement de covariable

Considérons l'ajustement de fonction linéaire par la méthode des moindres carrés :

$$\min_{\theta_1, \theta_2} \left[ \sum_{i=1}^{n_a} (\hat{f}(x_i^a) - y_i^a)^2 \right] \quad (4)$$

où

$$\hat{f}(x) = \theta_1 x + \theta_2$$

Ces moindres carrés ordinaires donnent une fonction qui parcourt bien les échantillons d'apprentissage, comme illustré sur la **Figure 01 (c)**. Cependant, la fonction apprise par les moindres carrés n'est pas utile pour prédire les valeurs de sortie des échantillons de test situés dans la partie droite du graphique.

Intuitivement, les échantillons d'apprentissage qui sont loin de la région de test (par exemple, les échantillons d'apprentissage avec  $x < 1$  sur la **Figure 01 (a)**) sont moins informatifs pour prédire les valeurs de sortie des échantillons de test situés dans la partie droite du graphique. Cela donne l'idée qu'il est plus prometteur d'ignorer ces échantillons d'apprentissage moins informatifs et d'apprendre uniquement à partir des échantillons d'apprentissage proches de la région de test (par exemple, les échantillons d'apprentissage avec  $x > 1,2$  sur la **Figure 01 (a)**). L'idée clé de l'adaptation de changement de covariable est de choisir (doucement) des échantillons d'apprentissage informatifs de manière systématique, en tenant compte de l'importance de chaque échantillon d'apprentissage dans la prédiction des valeurs de sortie de test. Plus précisément, nous utilisons le rapport des densités d'entrée d'apprentissage et de test (voir **Figure 01 (b)**),

$$\frac{p_t(x_i^a)}{p_a(x_i^a)}$$

comme poids pour le  $i$ -ème échantillon d'apprentissage dans l'ajustement des moindres carrés :

$$\min_{\theta_1, \theta_2} \left[ \sum_{i=1}^{n_a} \frac{p_t(x_i^a)}{p_a(x_i^a)} (\hat{f}(x_i^a) - y_i^a)^2 \right] \quad (5)$$

Ensuite, nous pouvons obtenir une fonction qui extrapole bien les échantillons de test (voir **Figure 01 (d)**). Notez que les échantillons de test ne sont pas utilisés pour

obtenir cette fonction. Dans cet exemple, les échantillons d'apprentissage situés dans la partie gauche du graphique (pour  $x < 1,2$ ) ont une importance presque nulle (voir **Figure 01 (b)**). Ainsi, ces échantillons sont essentiellement ignorés dans la méthode des moindres carrés pondérés en importance ci-dessus, et les échantillons informatifs au milieu du graphique sont automatiquement sélectionnés par pondération d'importance.

Comme illustré ci-dessus, les pondérations d'importance jouent un rôle essentiel dans l'adaptation des changements de covariables. Ci-dessous, le problème de l'adaptation du changement des covariables est formulé de manière plus formelle.

### 2.3.2. Apprentissage des fonctions à partir d'exemples

Considérons le problème d'apprentissage supervisé de l'estimation d'une dépendance d'entrée-sortie inconnue à partir d'échantillons d'apprentissage. Posons :

$$\{(x_i^a, y_i^a)\}_{i=1}^{n_a}$$

les échantillons d'apprentissage, où le point d'entrée de l'apprentissage :

$$x_i^a \in \mathcal{X} \subset \mathbb{R}^d, i = 1, 2, \dots, n_a$$

est un échantillon indépendant et identiquement distribué (i.i.d.) suivant une distribution de probabilité  $P_a(x)$  de densité  $p_a(x)$  :

$$\{x_i^a\}_{i=1}^{n_a} \stackrel{i.i.d.}{\sim} P_a(x) \quad (6)$$

La valeur de sortie de l'apprentissage :

$$y_i^a \in \mathcal{Y} \subset \mathbb{R}, i = 1, 2, \dots, n_a$$

suit une distribution de probabilité conditionnelle  $P(y|x)$  avec une densité conditionnelle  $p(y|x)$ .

$$y_i^a \sim P(y|x = x_i^a) \quad (7)$$

$P(y|x)$  peut être considérée comme la superposition de la sortie réelle  $f(x)$  et du bruit  $\epsilon$  :

$$y = f(x) + \epsilon \tag{8}$$

Nous supposons que le bruit  $\epsilon$  a une valeur nulle et une variance  $\sigma^2$ . Alors la fonction  $f(x)$  coïncide avec la valeur conditionnelle de  $y$ , étant donné  $x$ .

### 2.3.3. Fonctions de perte

Soit  $perte(x, y, \hat{y})$  la fonction de perte qui mesure l'écart entre la valeur de sortie réelle  $y$  en un point d'entrée  $x$  et son estimation  $\hat{y}$ . Dans les scénarios de régression où  $Y$  est continu, la perte quadratique est souvent utilisée [4].

$$perte(x, y, \hat{y}) = (\hat{y} - y)^2 \tag{9}$$

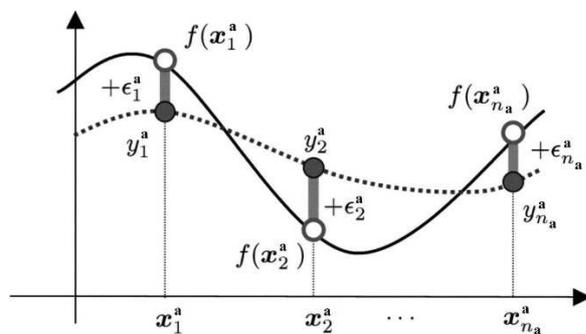
En revanche, dans les scénarios de classification binaire où  $Y = \{+1, -1\}$ , la perte 0-1 suivante est un choix typique car elle correspond au taux d'erreurs de classification.

$$perte(x, y, \hat{y}) = \begin{cases} 0 & \text{si } \text{sgn}(\hat{y}) = y, \\ 1 & \text{sinon} \end{cases} \tag{10}$$

où  $\text{sgn}(\hat{y})$  désigne le signe de  $\hat{y}$  :

$$\text{sgn}(\hat{y}) \begin{cases} +1 & \text{si } \hat{y} > 0 \\ 0 & \text{si } \hat{y} = 0 \\ -1 & \text{si } \hat{y} < 0 \end{cases}$$

Bien que les fonctions de perte ci-dessus soient indépendantes de  $x$ , la perte peut généralement dépendre de  $x$ .



**Figure 02 :** Modèle d'apprentissage supervisé.

### 2.4. Approximation des fonctions

Nous allons introduire des méthodes d'apprentissage capables de faire face au changement de covariable. Soit la fonction paramétrée  $\hat{f}(x; \theta)$  pour approximer une fonction cible  $f(x)$  à partir d'échantillons d'apprentissage  $\{x_i^a, y_i^a\}_{i=1}^{n_a}$ .

Une méthode standard pour apprendre le paramètre  $\theta$  serait la minimisation du risque empirique (ERM) : [4]

$$\hat{\theta}_{ERM} := \underset{\theta}{\operatorname{argmin}} \left[ \frac{1}{n_a} \sum_{i=1}^{n_a} perte(x_i^a, y_i^a, \hat{f}(x_i^a; \theta)) \right] \tag{11}$$

où  $perte(x, y, \hat{y})$  est une fonction de perte. Si  $P_a(x) = P_t(x)$ ,  $\hat{\theta}_{ERM}$  est connu pour être cohérent. Cependant, avec le changement de covariable où  $P_a(x) \neq P_t(x)$ , la situation est différente : ERM donne toujours un estimateur cohérent si le modèle est correctement spécifié, mais il n'est plus cohérent si le modèle est mal spécifié :

$$\operatorname{plim}_{n_a \rightarrow \infty} [\hat{\theta}_{ERM}] \neq \theta^* \tag{12}$$

où "plim" désigne la convergence des probabilités, et  $\theta^*$  est le paramètre optimal dans le modèle :

$$\theta^* := \underset{\theta}{\operatorname{argmin}} [Gen] \tag{13}$$

Gen est l'erreur de généralisation définie par :

$$Gen = \mathbb{E}_{x^t} \mathbb{E}_{y^t} [perte(x^t, y^t, \hat{f}(x^t; \theta))] \tag{14}$$

#### 2.4.1. ERM pondérée par l'importance

L'échec de la méthode ERM vient du fait que la distribution des entrées d'apprentissage est différente de la distribution des entrées de test. L'échantillonnage préférentiel est une technique standard pour compenser la différence des distributions. L'identité

suivante montre l'idée essentielle de l'échantillonnage préférentiel.

Pour une fonction  $g$ ,

$$\mathbb{E}_{x^t}[g(x^t)] = \int g(x)p_t(x)dx \quad (15)$$

$$\mathbb{E}_{x^t}[g(x^t)] = \int g(x)\frac{p_t(x)}{p_a(x)}p_a(x)dx \quad (16)$$

$$\mathbb{E}_{x^t}[g(x^t)] = \mathbb{E}_{x^a}\left[g(x^a)\frac{p_t(x^a)}{p_a(x^a)}\right] \quad (17)$$

où  $\mathbb{E}_{x^a}$  et  $\mathbb{E}_{x^t}$  désignent l'espérance sur  $x$  tirée de  $p_a(x)$  et  $p_t(x)$  respectivement.

La quantité :

$$\frac{p_t(x)}{p_a(x)}$$

s'appelle l'importance. L'identité ci-dessus montre que l'espérance d'une fonction  $g$  sur  $x^t$  peut être calculée par l'espérance pondérée préférentielle de la fonction sur  $x^a$ . Ainsi, la différence des distributions peut être systématiquement ajustée par pondération d'importance.

Avec le changement de covariable, ERM pondérée en importance (IWERM),

$\hat{\theta}_{IWERM}$

$$:= \operatorname{argmin}_{\theta} \left[ \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{p_t(x_i^a)}{p_a(x_i^a)} \operatorname{perte}(x_i^a, y_i^a, \hat{f}(x_i^a; \theta)) \right]$$

se révèle cohérente même pour des modèles mal spécifiés, c'est-à-dire qu'elle satisfait :

$$\operatorname{plim}_{n_a \rightarrow \infty} [\hat{\theta}_{IWERM}] = \theta^*$$

#### 2.4.2. IWERM adaptative

Comme indiqué ci-dessus, IWERM donne un estimateur cohérent. Cependant, elle peut également produire un estimateur instable, et donc IWERM peut ne pas être la meilleure méthode possible pour les échantillons finis. Dans la pratique, une variante légèrement stabilisée de IWERM serait préférable, c'est-à-dire obtenue en

"aplatissant" légèrement le poids d'importance dans IWERM. Nous appelons cette variante IWERM adaptative (AIWERM) :

$$\hat{\theta}_{\gamma} \quad (18)$$

$$:= \operatorname{argmin}_{\theta} \left[ \frac{1}{n_a} \sum_{i=1}^{n_a} \left( \frac{p_t(x_i^a)}{p_a(x_i^a)} \right)^{\gamma} \operatorname{perte}(x_i^a, y_i^a, \hat{f}(x_i^a; \theta)) \right]$$

où  $\gamma(0 \leq \gamma \leq 1)$  est appelé le paramètre d'aplatissement.

#### 2.4.3. IWERM régularisée

Au lieu d'aplatir la pondération d'importance, nous pouvons ajouter une régularisation au terme de risque empirique. Nous appelons cela IWERM régularisée :

$$\hat{\theta}_{\lambda} \quad (19)$$

$$:= \operatorname{argmin}_{\theta} \left[ \frac{1}{n_a} \sum_{i=1}^{n_a} \frac{p_t(x_i^a)}{p_a(x_i^a)} \operatorname{perte}(x_i^a, y_i^a, \hat{f}(x_i^a; \theta)) + \lambda R(\theta) \right]$$

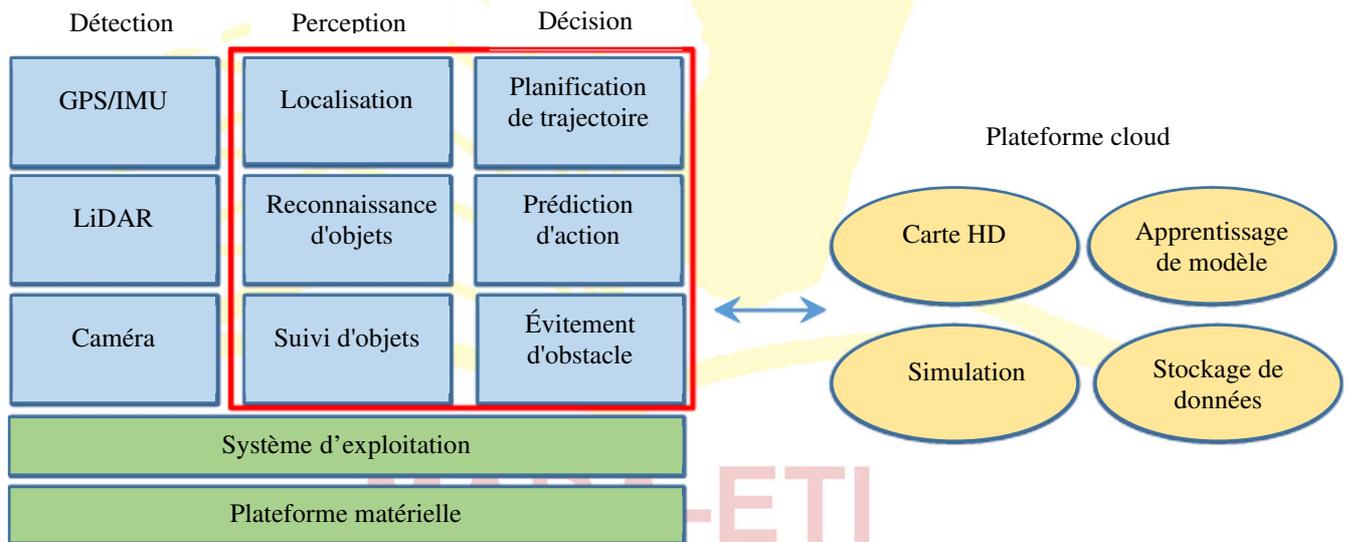
où  $R(\theta)$  est une fonction de régularisation et  $\lambda(\geq 0)$  est le paramètre de régularisation qui contrôle la force de la régularisation.

### 3. Apprentissage automatique et véhicules intelligents

#### 3.1. Architecture globale d'un système de conduite autonome

La conduite autonome n'est pas une technologie unique, mais plutôt un système extrêmement complexe composé de nombreux sous-systèmes. Décomposons-le en trois composantes principales : les algorithmes, y compris la détection, la perception et la décision (qui nécessite un raisonnement pour les cas complexes) ; les systèmes clients, y compris le système d'exploitation et la plate-forme matérielle ; et la plate-forme cloud, y compris la carte haute définition (HD), la formation aux modèles d'apprentissage en profondeur, la simulation et le stockage de données. [5][6]

Le sous-système d'algorithme extrait des informations significatives des données brutes du capteur pour comprendre son environnement et prendre des décisions sur ses actions futures. Les systèmes clients intègrent ces algorithmes pour répondre aux exigences de temps réel et de fiabilité. Par exemple, si la caméra génère des données à 60 Hz, les systèmes clients doivent s'assurer que l'étape la plus longue du pipeline de traitement prend moins de 16 ms. La plateforme cloud fournit des capacités de calcul et de stockage hors ligne pour les voitures autonomes. Avec la plateforme cloud, nous sommes en mesure de tester de nouveaux algorithmes, de mettre à jour la carte HD et de former de meilleurs modèles de reconnaissance, de suivi et de décision.



**Figure 03** : Architecture d'un système de conduite autonome

#### 3.2. Algorithme de conduite autonome

Les algorithmes consistent en la détection, c'est-à-dire l'extraction d'informations significatives à partir des données brutes des capteurs ; la perception, qui consiste à localiser le véhicule et à comprendre l'environnement courant ; et la décision, c'est-à-dire prendre des mesures pour atteindre de manière fiable et sûre les destinations cibles.

##### 3.2.1 La détection

Normalement, une voiture autonome se compose de plusieurs capteurs majeurs. En effet, chaque type de capteur présentant des avantages et des inconvénients, dans les véhicules autonomes, les données de plusieurs capteurs doivent être combinées pour une fiabilité et une sécurité accrues. Ils peuvent inclure les éléments suivants : [6]

- **GPS/IMU** : Le système GPS/IMU aide le véhicule autonome à se localiser en signalant à la fois des mises à jour inertielles et une estimation de la position globale à un taux élevé, par exemple 200 Hz. Le GPS est un capteur de localisation assez précis, mais son taux de mise à jour est lent, à environ 10 Hz seulement. Il est donc incapable de fournir des mises à jour en temps réel. Cependant, les erreurs IMU s'accumulent avec le temps, conduisant à une dégradation correspondante des estimations de position. Néanmoins, une IMU peut fournir des mises à jour plus fréquemment, à ou au-dessus de 200 Hz. Cela devrait satisfaire l'exigence de temps réel. En combinant à la fois le GPS et l'IMU, nous pouvons fournir des mises à jour précises et en temps réel pour la localisation des véhicules.
- **LiDAR** : Le LiDAR est utilisé pour la cartographie, la localisation et l'évitement d'obstacles. Il fonctionne en faisant rebondir un faisceau sur des surfaces et mesure le temps de réflexion pour déterminer la distance. En raison de sa grande précision, le LiDAR peut être utilisé pour produire des cartes HD, pour localiser un véhicule en mouvement par rapport à des cartes HD, pour détecter un obstacle devant, etc. Normalement, une unité LiDAR, telle que le laser Velodyne à 64 faisceaux, tourne à 10 Hz et prend environ 1,3 million de lectures par seconde.
- **Caméras** : Les caméras sont principalement utilisées pour la reconnaissance d'objets et les tâches de suivi d'objets telles que la détection de voie, la détection des feux de signalisation et la détection des piétons, etc. Pour améliorer la sécurité des véhicules autonomes, les implémentations existantes

montent généralement huit caméras 1080p ou plus autour de la voiture, de sorte que nous pouvons les utiliser pour détecter, reconnaître et suivre les objets devant, derrière et des deux côtés du véhicule. Ces caméras fonctionnent généralement à 60 Hz et, lorsqu'elles sont combinées, génèrent environ 1,8 Go de données brutes par seconde.

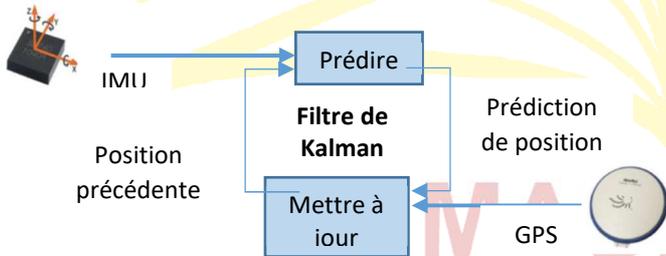
- **Radar et Sonar** : Le système radar et sonar est principalement utilisé pour la dernière ligne de défense pour éviter les obstacles. Les données générées par le radar et le sonar indiquent la distance ainsi que la vitesse de l'objet le plus proche sur la trajectoire du véhicule. Une fois que nous détectons qu'un objet n'est pas loin devant, il peut y avoir un risque de collision, alors le véhicule autonome doit utiliser les freins ou tourner pour éviter l'obstacle. Par conséquent, les données générées par le radar et le sonar ne nécessitent pas beaucoup de traitement et sont généralement transmises directement au processeur de contrôle, et donc pas à travers le pipeline de calcul principal pour mettre en œuvre des fonctions « urgentes » telles que dévier, utiliser les freins ou prétendre les ceintures de sécurité.

### 3.2.2 La perception

Les données des capteurs sont ensuite introduites dans la phase de perception pour fournir une compréhension de l'environnement du véhicule. Les trois tâches principales de la perception de la conduite autonome sont la localisation, la détection d'objets et le suivi d'objets.

Le GPS/IMU peut être utilisé pour la localisation et, comme mentionné ci-dessus, le GPS fournit des résultats de localisation assez précis mais avec un taux de mise à jour comparativement faible, tandis qu'une IMU fournit des mises à jour très rapides au

prix de résultats moins précis. Nous pouvons ainsi utiliser les techniques de filtre de Kalman pour combiner les avantages des deux et fournir des mises à jour de position précises et en temps réel. Comme le montre la **Figure 4**, cela fonctionne comme suit : l'IMU met à jour la position du véhicule toutes les 5 ms, mais l'erreur s'accumule avec le temps. Heureusement, toutes les 100 ms, une mise à jour GPS est reçue, ce qui permet de corriger l'erreur IMU. En exécutant ce modèle de propagation et de mise à jour, la combinaison GPS/IMU peut générer des résultats de localisation rapides et précis. Néanmoins, nous ne pouvons pas nous fier uniquement à cette combinaison pour la localisation pour trois raisons : (1) la précision n'est que d'environ un mètre ; (2) le signal GPS a des problèmes de trajets multiples, ce qui signifie que le signal peut rebondir sur les bâtiments, introduisant plus de bruit; et (3) le GPS nécessite une vue dégagée du ciel et ne fonctionnerait donc pas dans des environnements tels que les tunnels. [6]



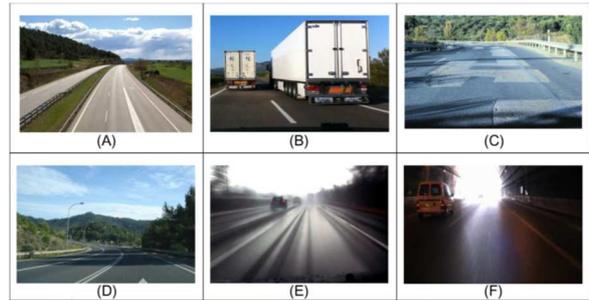
**Figure 4** : Localisation GPS/IMU

### 3.2.3. La variabilité environnementale

La détection des voies et la reconnaissance des panneaux de signalisation dépendent d'un certain nombre de paramètres, tels que l'évolution rapide de la morphologie de la route et des conditions d'éclairage, ce qui en fait un problème non négligeable. La conception de systèmes efficaces de détection et de reconnaissance des voies et des panneaux de signalisation nécessite le développement d'algorithmes complexes et

spécifiques prenant en charge de nombreux paramètres et problèmes, avec dans de nombreux cas des ressources de calcul limitées.

La Figure 5 montre certaines situations rendant la détection de voie difficile à gérer, tandis que la Figure 6 montre certains scénarios de reconnaissance de panneaux de signalisation difficiles à gérer.



**Figure 5** : Exemples de situations extrêmes à gérer par les systèmes de détection de voie: (A) Marquage de voie différent; (B) occlusion de voie; (C) Changement de texture de la chaussée; (D) voies avec ombre; (e) Route pluvieuse; (F) Image saturée à la sortie du tunnel.



**Figure 6** : Exemples de situations difficiles à gérer par les systèmes de reconnaissance des panneaux de signalisation (A)Occlusion (B)étalement (C)Saturation (D)Dégradation (E)Ombres (F)Rétroéclairage.

## 4. Optimisation des fonctions de coût dans le contexte de véhicules intelligents

### 4.1. Modèle de prédiction et fonction de coût

Considérons le contrôle de mouvement latéral. Le problème du contrôle latéral est complexe en raison de la dynamique

couplée longitudinale et latérale ainsi que du comportement du pneu.

Les différents algorithmes de contrôle de prédiction diffèrent dans les modèles utilisés pour décrire le système et dans la fonction de coût à minimiser [7].

Les expressions mathématiques pour la prédiction et la fonction de coût sont :

$$\hat{y} = Gu + f \quad (20)$$

avec  $u$ , la fonction de coût, est proportionnelle au couple latéral :

$$J = \sum_{j=1}^p [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^m \lambda [\Delta u(t+j) - 1]^2 \quad (21)$$

Où

- $\hat{y}$  est un vecteur de dimension égale à l'horizon de prédiction contenant les sorties prédites jusqu'à l'horizon de prédiction  $p$  ;
- $w$  est la future valeur souhaitée de sortie ;
- $u$  est un vecteur de dimension égale à l'horizon de contrôle  $m$  contenant les actions de contrôle futures ;
- $G$  est le contrôle matriciel dynamique du système localement linéarisé ;
- et  $f$  est le vecteur de réponse libre, de dimension égale à l'horizon de prédiction.

La réponse libre est la prédiction du comportement du système si la commande reste constante et égale à la dernière commande calculée. Le paramètre  $\lambda$  permet d'effectuer la pesée des erreurs de suivi de trajectoire et des efforts de contrôle séparément, de telle manière que l'on pourrait concevoir un contrôleur pour essayer de s'ajuster à la trajectoire souhaitée, quelle que soit la commande

d'utilisation, ou d'autre part le contrôleur pourrait être plus permissif avec les erreurs de suivi de chemin et a une utilisation plus douce de la commande, économisant de l'énergie dans la commande.

### Interprétations :

- Le modèle du processus est utilisé pour prédire les sorties futures en utilisant les informations des signaux d'entrée passés, des commandes de contrôle passées, ainsi que les actions de contrôle futures calculées par un optimiseur.
- Pour calculer les futurs signaux de commande, l'optimiseur utilise la fonction de coût mentionnée précédemment. En gardant à l'esprit cette explication, le processus modèle est fondamental pour le bon fonctionnement du système.

Notons que si, dans le processus d'optimisation, on n'inclut pas les restrictions du modèle physique, il est possible d'obtenir la minimisation de la fonction de coût suivante de manière analytique :

$$J = ee^T + \lambda uu^T \quad (22)$$

où  $e$  est le vecteur des erreurs prédites jusqu'à l'horizon de prédiction et  $u$  est le vecteur des incréments de commande de signal futurs jusqu'à l'horizon de commande.

L'expression mathématique pour calculer les futures commandes est obtenue en prenant la dérivée de  $J$  et en l'égalisant à zéro :

$$u = (G^T G + \lambda I)^{-1} G^T (w - f) \quad (23)$$

L'optimiseur pourra calculer l'angle de braquage de manière à minimiser les différences entre la réponse libre et la trajectoire souhaitée. En d'autres termes, l'optimiseur calculera l'angle de braquage afin de produire le meilleur suivi de trajectoire.

## Interprétations :

- L'application lit les capteurs et définit les valeurs des états internes du système à chaque itération. Ces états sont la position, l'orientation et la vitesse du véhicule. Avec ces valeurs, la réponse échelonnée du système est calculée. Les paramètres de la réponse échelonnée forment la matrice dynamique G.
- La prédiction du comportement du véhicule est calculée à partir des valeurs lues des capteurs au début de l'itération. La prédiction du mouvement du véhicule est comparée à la trajectoire souhaitée à partir du point le plus proche du prototype.
- Le vecteur des erreurs futures est le résultat de la comparaison précédente. Les futures commandes sont l'équation, mais seul le premier terme du futur vecteur de commandes est appliqué, en gardant à l'esprit le concept d'horizon glissant. Enfin, les variables de l'algorithme sont mises à jour.

### 4.2. Fonction de coût de la planification de mouvement

La tâche de la planification de mouvement est de générer une trajectoire et de l'envoyer au contrôle de rétroaction pour l'exécution de la commande physique du véhicule. La trajectoire planifiée est généralement spécifiée et représentée comme une séquence de points de trajectoire planifiés. Chacun de ces points contient des attributs tels que l'emplacement, l'heure, la vitesse, la courbure, etc.

#### 4.2.1 La planification de mouvement avec planification longitudinale et planification latérale

Au lieu de faire de la planification de la trajectoire puis de la planification de la vitesse, considérons l'idée de faire une planification longitudinale et latérale. Une

trajectoire de planification de mouvement souhaitée doit être lisse, ce qui est généralement représenté mathématiquement en continuité de pose avec ses dérivés dans une dimension (par exemple, position, vitesse et accélération). La facilité et le confort peuvent être mieux mesurés par la secousse, qui est le taux de changement d'accélération.

Considérons la trajectoire optimale de secousse reliant un état de départ  $P_0 = [p_0, \dot{p}_0, \ddot{p}_0]$  et un état final  $P_1 = [p_1, \dot{p}_1, \ddot{p}_1]$ , où la pose P pourrait être soit la s-pose longitudinale ou l-pose latérale, dans un laps de temps de  $T := t_1 - t_0$ . Notons l'intégrale de la secousse quadratique :  $S_t(p_t) := \int^{t_1} \ddot{p}^2(\tau) d\tau$ . Une proposition importante que nous avons est que, pour toute fonction de coût avec la forme [6] :

$$J = K_j S_t + K_t g(T) + K_p h(p_1) \quad (24)$$

où g et h sont des fonctions arbitraires et  $K_j, K_t, K_p > 0$ , la solution optimale pour minimiser la fonction de coût ci-dessus est un polynôme quintique. Intuitivement, la fonction de coût pénalise l'intégrale de secousse élevée  $S_t$  le long de la trajectoire, et considère également le temps T et la pose finale  $h(p_1)$ .

#### 4.2.2. La planification latérale

Nous commençons la planification avec la dimension L latérale. L'état de départ  $P_0$  devient  $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$ , et nous définissons cet état de départ en fonction de l'état de fin réel du cadre de planification précédent, de sorte que les continuités soient maintenues. Les états finaux  $P_1$  sont choisis parmi un ensemble de décalages latéraux possibles avec  $\dot{d}_1 = \ddot{d}_1 = 0$  puisque nous préférons nous déplacer parallèlement à la direction de la ligne de référence centrale (direction s). Les fonctions g et h sont choisies avec  $g(T)=T$

et  $h(d_1) = d_1^2$ . On voit que la convergence lente est pénalisée ainsi que toute différence latérale avec  $d = 0$  à l'état final. Avec la solution optimale prenant la forme d'un polynôme quintique :

$$l_{optimal}(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \quad (25)$$

qui minimise la fonction de coût de :

$$J_l = K_j S_t(l(t)) + K_t T + K_l d_1^2 \quad (26)$$

Les coefficients  $a_5, a_4, a_3, a_2, a_1, a_0$  pourraient être calculés avec des conditions aux limites à  $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$  et  $D_1 = [d_1, \dot{d}_1, \ddot{d}_1]$ . Nous pourrions choisir un ensemble de  $d_i$  et calculer un ensemble de meilleures trajectoires unidimensionnelles comme  $Solution_{i,j}$  à partir de  $[d_1, \dot{d}_1, \ddot{d}_1, T]_{i,j} = [d_1, 0, 0, T_j]$ .

#### Interprétations :

- Chaque solution candidate aura un coût. Pour chaque candidat, nous vérifions si cette solution sera cohérente avec les sorties de décision comportementale en amont et rendons cette solution valide si des violations/collisions sont trouvées. Les trajectoires valides restantes forment un ensemble candidat de dimension latérale et seront utilisées dans le calcul de la trajectoire planifiée sur la dimension 2D.
- La méthode décrite ci-dessus fonctionne bien pour les trajectoires à grande vitesse où le déplacement longitudinal et le déplacement latéral pourraient être choisis indépendamment.

#### 4.3. Fonction de coût d'une cyberattaque

La croissance significative des systèmes autonomes implique un large éventail de systèmes cyber-physiques, et expose ainsi à la menace de cyberattaques en raison de leur intégration étroite des ressources

informatiques, des processus physiques et des capacités de communication.

Considérons un modèle linéaire en temps discret soumis à des cyberattaques a priori inconnues :

$$x_a(k+1) = Ax_a(k) + Bu(k) + B_a a(k), \quad (27)$$

$$x_a(0) = x_0$$

Où

- l'indice « a » désigne le système attaqué ;
- $x_a(k) \in \mathbb{R}^n$  et  $u(k) \in \mathbb{R}^p$  sont les états et entrées du système
- A et B sont les matrices système de dimensions appropriées ;
- $k \in \mathbb{N}$  désigne l'indice de temps discret, prenant des valeurs de l'horizon temporel (éventuellement infini)  $\mathbb{N} = \{0, 1, 2, \dots, N\}$

Différentes séquences d'attaque  $a(k) \in \mathbb{R}^s$  peuvent être injectées dans le système avec les matrices d'attaque  $B_a$  de dimension compatible. On suppose que les paires de matrices système (A, B) satisfont à la condition de contrôlabilité. En pratique, les attaques injectées sont réalisées à travers la structure physique des composants de l'actionneur. Ainsi, la matrice d'attaque  $B_a$  doit satisfaire [8] :

$$portée\{B_a\} \subseteq portée\{B\} \quad (28)$$

Où « portée » désigne l'espace des colonnes de la matrice.

Pour simplifier, nous allons utiliser la notation suivante :

$$x_{[\tau_1, \tau_2]} := [x^T(\tau_1) x^T(\tau_1 + 1) \dots x^T(\tau_2)]^T, \\ 0 \leq \tau_1 \leq \tau_2 \leq N$$

Avec des mesures d'état parfait, l'historique des états du système,  $x_{a[0,k]}$ , et les attaques passées,  $a_{[0,k-1]}$ , sont disponibles pour chaque pas de temps k, bien que les signaux de futures attaques  $a_{[k,N]}$  soient inconnus.

Pour mesurer l'impact des cyberattaques sur le système, nous introduisons une fonction de coût  $J(u_{[\tau_1, \tau_2]}, a_{[\tau_1, \tau_2]})$ . La fonction de coût pendant l'intervalle de temps  $[\tau_1, \tau_2]$  est définie par la forme quadratique suivante [8] :

$$\begin{aligned} J(u_{[\tau_1, \tau_2]}, a_{[\tau_1, \tau_2]}) & \quad (29) \\ & := \sum_{k=\tau_1}^{\tau_2} (x_a^T(k)Qx_a(k) \\ & \quad + u^T(k)Ru(k)) \\ & \quad + x_a^T(\tau_2 + 1)Qx_a(\tau_2 + 1) \end{aligned}$$

Où  $Q \geq 0$  et  $R > 0$  sont des paramètres de conception comprenant les informations sur les facteurs d'échelle, les pondérations de sécurité, les efforts de contrôle, etc.

#### Interprétations :

- Notons que la fonction de coût  $J$  peut être établie différemment selon l'objectif opérationnel du système cyber-physique, et donc motiver différentes stratégies de contrôle. On peut voir que  $J$  évalue simultanément les performances de contrôle et la sévérité de l'attaque sur l'intervalle de temps  $[\tau_1, \tau_2]$ .

#### 4.4. Réduction de la dimensionnalité

Les dispositifs informatiques omniprésents ont modifié l'acquisition des données de mobilité : le taux de pénétration élevé et la capacité à capturer et partager des informations en continu. Cela s'applique aux informations de géolocalisation, à la position GPS, les données opérationnelles de téléphonie mobile, fréquemment échangées avec le réseau, ainsi que les informations de crowdsourcing sur les réseaux sociaux. De plus, sous l'égide de la tendance de l'IoT, le déploiement du concept de véhicule connecté, soutenu par des communications V2X avancées, fournit un volume de données massif. [7]

La planification de mouvement, d'abord pour les robots mobiles, puis pour les véhicules autonomes, a été largement étudiée au cours des dernières décennies. Les stratégies résultantes ont été conçues pour répondre, sous différentes hypothèses, à une variété de contraintes cinématiques, dynamiques et environnementales. La notion de costmap peut être considérée à ce stade.

Un costmap est un concept fondamental en robotique autonome. Il représente le coût (difficulté) de traverser différentes zones de la carte. Le costmap conserve les informations sur les zones occupées/libres sur la carte sous la forme d'une grille d'occupation. Il utilise les données du capteur et les informations de la carte statique pour stocker et mettre à jour les informations sur les obstacles dans le monde, qui sont marqués sur la carte (ou effacés, s'ils ne sont plus là). [7]

Le calcul de costmap est pris en charge sur un costmap en couches qui sera utilisé pour l'intégration des différentes sources d'informations dans un costmap monolithique unique. Au niveau de chaque couche, les informations sur les zones occupées/libres dans les environs du véhicule sont conservées sous la forme d'une grille d'occupation en utilisant les différentes sources d'observation comme entrée. En utilisant ces informations, les obstacles dynamiques et statiques sont indiqués sur la carte. Par exemple, supposons que chaque cellule de la carte puisse avoir 255 valeurs de coût différentes. Ensuite, à chaque couche, le costmap est représenté comme suit :

- Une valeur de 255 signifie qu'aucune information n'est disponible sur une cellule spécifique de la carte ;
- 254 signifie qu'un capteur a marqué cette cellule spécifique comme étant

occupée. Ceci est considéré comme une cellule mortelle, donc le véhicule ne doit jamais y entrer ;

- Le reste des cellules est considéré comme libre, mais avec des niveaux de coûts différents selon une méthode d'accroissement par rapport à la taille du véhicule et à sa distance par rapport à l'obstacle.

Les valeurs de coût diminuent avec la distance à la cellule occupée la plus proche à l'aide de l'expression suivante :

$$C(i, j) = \exp(-1 \cdot \alpha \cdot (\|c_{ij} - \vec{d}\| - \rho_{\text{inscrit}})) \cdot 253$$

Dans cette expression,  $\alpha$  est un facteur d'échelle qui permet d'augmenter ou de diminuer le taux de décroissance du coût de l'obstacle.  $\|c_{ij} - \vec{d}\|$  est la distance entre la cellule  $c_{ij} \in \mathcal{C}$  (où  $\mathcal{C}$  est l'ensemble des cellules du costmap) et l'obstacle. Enfin,  $\rho_{\text{inscrit}}$  est le rayon inscrit, qui est le cercle intérieur des limites de la voiture.

## 5. Conclusion

L'optimisation a été divisée en deux parties. La première partie a montré des résultats de l'application d'une fonction de coût sur un contrôleur prédictif. Le modèle peut ainsi assurer le bon fonctionnement du système pour prédire les sorties futures en utilisant les informations des signaux d'entrée passés, des commandes de contrôle passées, ainsi que les actions de contrôle futures calculées par un optimiseur. On a pu voir le résultat d'un optimiseur bayésien qui utilise les informations de performances pour mettre à jour de manière itérative le modèle dynamique afin d'améliorer la performance. Une fonction de coût peut aussi être appliquée à la planification du mouvement, que ce soit longitudinal ou latéral. Elle permet de voir la cohérence de la solution avec les sorties de décision comportementale en amont et ainsi valider les trajectoires si des violations ou collisions sont trouvées.

Avec l'importance de la connectivité pour un véhicule autonome et connecté, nous avons pu définir la fonction de coût d'une cyberattaque. Le résultat est l'évaluation simultanée des performances de contrôle et de la sévérité de l'attaque sur un intervalle de temps.

La deuxième partie travaille sur la réduction de la dimensionnalité vu le volume de données manipulées et traitées. On a pu utiliser le concept de costmap à partir duquel on a pu élaborer la carte de collision. Les différents niveaux de danger seront ainsi cartographiés.

## BIBLIOGRAPHIE

- [1] C. Sammut, G. I. Webb, « *Encyclopedia of Machine Learning and Data Mining* », Springer, 2017
- [2] S. Shalev-Shwartz, S. Ben-David, « *Understanding machine learning: from theory to algorithms* », Cambridge University Press, 2014
- [3] S Theodoridis, « *Machine Learning, A Bayesian and Optimization Perspective* », Academic Press, 2020
- [4] M Su.giyama, M. Kawanabe, « *Machine Learning in Non-Stationary Environments – Introduction to Covariate Shift Adaptation* », MIT PRESS, 2012
- [5] X. Zhang, M. Mansoor Khan, « *Principles of Intelligent Automobiles* », Springer, 2019
- [6] S. Liu, L. Li, J. Tang, S. Wu, J.-L. Gaudiot, « *Creating Autonomous Vehicle Systems* », Morgan & Claypool Publishers, 2018
- [7] F Jiménez, « *Intelligent Vehicles* », Elsevier, 2018
- [8] H. Yu, X. Li, R. M. Murray, S. Ramesh, C. J. Tomlin, « *Safe, Autonomous and Intelligent Vehicles* », Springer, 2019