

Fiabilité et sécurité fonctionnelle des ADAS ADAS Reliability and Safety

Frédérique VALLÉE, Laurent RAFFAELLI,
ALL4TEC
Immeuble Odyssée – Bât. E,
2-12 rue du Chemin des Femmes, 91300 Massy

Guy FAYOLLE et al.¹
INRIA - Rocquencourt
Domaine de Voluceau B.P.105
78153 Le Chesnay

Résumé

La validation de la sécurité fonctionnelle et de la fiabilité des systèmes d'aide à la conduite (aussi appelés ADAS) est un problème complexe. Il est lié à la difficulté de construire à moindre coût des campagnes de test suffisamment complètes pour valider les objectifs de sécurité fonctionnelle et de fiabilité des ADAS. Or la présence de ces systèmes tend à se généraliser dans les véhicules automobiles modernes, ce qui fait de la problématique de leur validation un enjeu important. En cas d'accident, la responsabilité juridique du constructeur peut être directement engagée. Dans ce cadre, le projet COVADEC a démarré au milieu de l'année 2013. Il a pour objectif de développer des méthodes et outils directement industrialisable pour faire face à cette problématique, tout en évaluant l'applicabilité de normes de sécurité fonctionnelle telle que l'ISO 26262 à la conception et à la validation des ADAS. Ce document présente un résumé des travaux réalisés dans le cadre de ce projet, avec les points durs rencontrés et les solutions mises en œuvre pour les dénouer.

Summary

Safety and reliability validation of Advanced Driver Assistance Systems (ADAS) is not a trivial issue. Build at a low cost level a sufficiently detailed campaign in order to validate safety related and reliability related goals for an ADAS is very difficult, since this kind of systems can encounter an almost infinite number of situations. Now, this kind of systems tends to be more and more integrated in modern cars, what makes of the problem of ADAS validation an important stake. In the event of an accident, the manufacturer can be held legally responsible. In this context, the COVADEC project has started in the mid-2013. The COVADEC project aims to provide methods and techniques for automotive OEMs and suppliers who face these problems, while enhancing the global knowledge of automotive safety standard likes ISO 26262 applicability to design and validation of ADAS sensors, and shed light on its limitations, in order to propose solutions. This document presents a summary of the works realized within this project, with the met hard points and the solutions implemented to solve them.

1 Avant-propos

La validation des systèmes d'aide à la conduite (ou ADAS en anglais, pour Advanced Driver Assistance Systems) est un problème complexe et stratégique, alors que ces systèmes deviennent incontournables dans le domaine automobile. Les ADAS apportent du confort aux conducteurs, et peuvent même être un argument de vente. Mais leurs fonctions, bien qu'utiles, ne doivent pas affecter la sécurité générale du véhicule, qui est de la responsabilité du constructeur.

Les ADAS actuels sont, pour la plupart, basés sur des capteurs caméra, fournissant des entrées aux applications telles que la détection d'obstacle ou la détection de piétons qui sont elles-mêmes des composants essentiels de fonctions comme le système de freinage d'urgence automatique. Ces systèmes qui protègent les utilisateurs de la route prennent encore plus d'importance avec les nouveaux protocoles d'évaluation de la sécurité des véhicules, qu'ils soient mis au point en Europe ou en Amérique du Nord. Pour ces raisons, la robustesse et la fiabilité des fonctions ADAS ne peuvent pas être négligées et les constructeurs automobiles doivent se doter d'outils pour s'assurer que les fonctions ADAS intégrées à leurs véhicules fonctionnent avec le niveau de sécurité requis.

La complexité de ces systèmes combinée avec le nombre quasiment infini de combinaisons de paramètres liées aux profils d'usage de fonction basées sur des capteurs caméra nécessitent de développer de nouvelles méthodes d'optimisation de test et de nouveaux outils pour concevoir et valider les systèmes ADAS. Les ressources nécessaires en coût et en délai, si l'on souhaite utiliser les techniques actuelles de test, rendent ces techniques de moins en moins adaptées aux nouvelles fonctionnalités de sécurité, qui induisent de très fortes exigences en terme de validation. Aujourd'hui, pour valider la sécurité des ADAS basés sur des caméras, des véhicules de test sont équipés avec ces systèmes et doivent rouler un très grand nombre de kilomètres, parfois pendant des années. Il y a donc un besoin évident d'améliorer les techniques utilisées pour la validation des ADAS, tout en respectant les normes de sécurité fonctionnelle automobile telle que l'ISO 26262. L'objet de ce document est de présenter les techniques développées au cours du projet COVADEC pour répondre à cette problématique.

2 Le projet COVADEC

Le projet de recherche et développement COVADEC (*), débuté vers le milieu de l'année 2013, a pour objectif de fournir des méthodes et techniques aux équipementiers et fournisseurs automobiles confronté à ces problèmes.

(*) COVADEC signifie « COnception et VALIDation Des systèmes Embarqués d'aide à la Conduite ».

Les principaux objectifs de COVADEC sont les suivants :

- Optimiser les scénarios de tests et ainsi réduire les kilomètres d'essais nécessaires à la validation des systèmes ADAS.
- Optimiser les coûts et délais de validation des systèmes ADAS.
- Répondre efficacement aux exigences de fiabilité et de sécurité conformément aux normes (ISO 26262, etc...).
- Prendre en compte les exigences de sûreté de fonctionnement en amont du développement des algorithmes de traitement d'images.
- Standardiser les méthodes et outils nécessaires pour la validation d'exigences de sécurité fonctionnelles et opérationnelles.

- Améliorer le développement des fonctions ADAS en anticipant et en implémentant en priorité les situations critiques qui peuvent mettre en défaut les systèmes d'aide à la conduite.
- Assurer l'interopérabilité entre les plateformes de test, de simulation et les autres plateformes de développement.

Le domaine des ADAS n'étant pour le moment pas pleinement couvert par les exigences de sécurité fonctionnelle de la norme ISO 26262, un des objectifs du projet est de proposer de nouvelles solutions qui pourraient être intégrées dans le processus de validation mis en place pour n'importe quel système automobile de manière à combler les manques de cette norme sur le sujet.

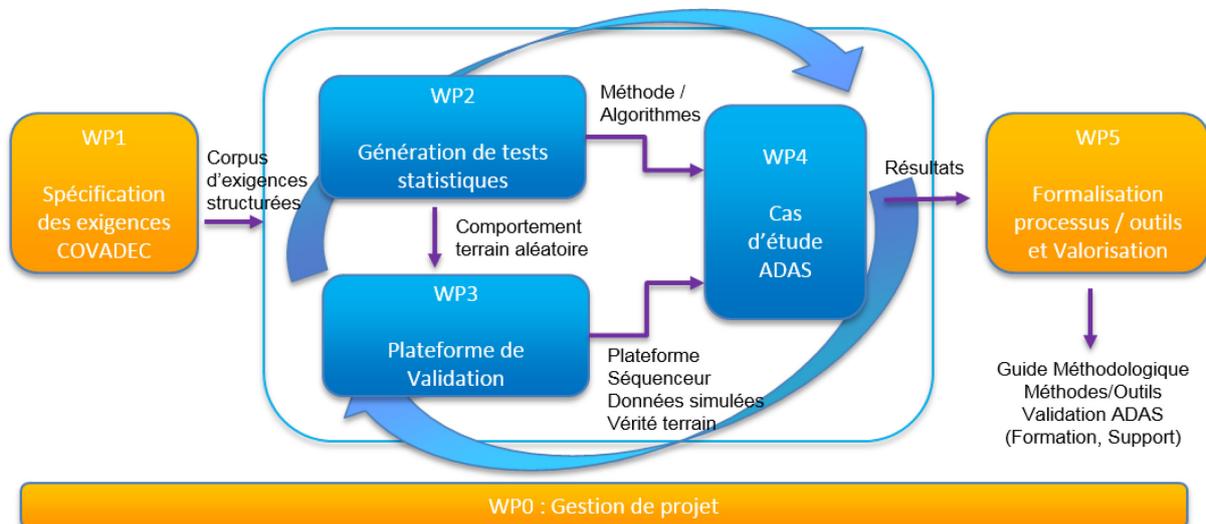


Figure 1. Structuration du projet en Work-Packages

COVADEC est divisé en 5 Work-Packages :

- WP0 : Conduite du projet
- WP1 : Spécifications des exigences COVADEC

Les principaux objectifs du WP1 sont :

- Collecter et faire une synthèse des besoins des parties prenantes dans l'industrie automobile en termes de certification et de validation, plus particulièrement sur des systèmes d'aide à la conduite basés sur la perception de leur environnement.
- Identifier les pratiques communes de gestion de la validation et exprimer les attentes de l'industrie pour une amélioration du processus.
- Spécifier l'architecture des solutions proposées en réponse à ces attentes.

- WP2 : Génération de tests aléatoires :

Le WP2 a pour objectif de développer une méthodologie et les outils associés pour générer les tests statistiques couvrant les exigences du projet COVADEC.

- WP3 : Plateforme de validation :

Dans le WP3 sont développés les outils logiciels utilisés pour l'exécution des cas de test générés avec les outils du WP2 et la génération des rapports de tests.

- WP4 : Cas d'usage ADAS :

Deux cas d'usage ont été choisis pour démontrer la validité de la méthode et de la plateforme COVADEC :

- Une fonction LDW (Lane Departure Warning), fonction qui informe le conducteur en cas de franchissement de ligne non désiré.
 - Une fonction AEB (Autonomous Emergency Braking), fonction destinée à réduire le risque de collision frontale.
- WP5 : Définition du processus et valorisation des outils :

Les objectifs du WP5 sont d'étendre les résultats des études des cas d'usage réalisées dans COVADEC à toutes les autres fonctions ADAS ; de mettre à disposition de la communauté ADAS les résultats du projet COVADEC et d'expliquer comment les résultats seront réutilisés et mis en valeur par les partenaires du projet.

3 Positionnement par rapport à l'état de l'art et à la norme ISO 26262

Le rôle de la sécurité fonctionnelle, pour garantir la sécurité des biens et des personnes, est d'évaluer les risques intrinsèques du système et ainsi apporter des solutions pour réduire l'occurrence des dangers. Le sujet a été largement couvert et différentes normes ont vu le jour, dont en 2011, la norme ISO 26262 qui est applicable à l'électronique programmable des ADAS pour les systèmes automobiles.

Du point de vue des ADAS, les risques de sécurité principaux concernent une analyse erronée de la situation de conduite, qui peut remonter une information incorrecte au conducteur, voire pire, provoquer une réponse automatique inappropriée du véhicule. Par exemple, pour une fonction de suivi du marquage, si l'ADAS est couplé au contrôle latéral du véhicule, le système peut déclencher à tort un brusque écart du véhicule, ou dans le cas d'une fonction de prévention de collision frontale, provoquer un freinage intempestif.

Dans les faits, la version actuelle de la norme ISO 26262 exclut explicitement les dangers qui sont inhérents au comportement nominal du système, c'est-à-dire lorsque les dangers potentiels sont causés par des limitations intrinsèques des performances des capteurs, et ne sont pas le résultat d'une défaillance du système.

De fait, les décisions erronées potentiellement prises par un ADAS peuvent être considérées comme un comportement normal du système et par conséquent ne sont pas couvertes par la norme. Par conséquent, les risques résiduels les plus critiques ne sont plus les défaillances logicielles ou matérielles, qui sont déjà couvertes par la norme ISO 26262 existante, mais les cas où le comportement de l'ADAS diverge du comportement nominal.

Une décision erronée d'ADAS concerne deux sortes d'événements redoutés :

- Le système ne détecte pas la situation dangereuse qui aurait dû être détectée comme telle.
- Le système détecte une situation dangereuse alors que ça n'est pas le cas dans la réalité.

Dans le cas d'une fonction de freinage d'urgence autonome, le premier cas est moins critique en terme de sécurité fonctionnelle, car le conducteur peut toujours prendre la bonne décision et l'ADAS ne déclenche aucune action. Dans ce cas, la qualité de service est diminuée, mais il n'y a pas plus de risque qu'avec un véhicule non équipé de ce système. En revanche, le second cas est critique, car l'ADAS peut générer une mauvaise information et déclencher une mauvaise réaction du véhicule. Dans ce cas, utiliser un véhicule équipé de cet ADAS est plus risqué qu'utiliser un véhicule non-équipé.

De manière à valider que les ADAS sont robustes face aux événements redoutés, différentes approches ont été étudiées par le projet COVADEC :

- La première approche, basée sur la preuve formelle, a été écartée car elle est non exploitable dans le cas de systèmes très complexes (explosion des algorithmes de preuves) et l'expression formelle peut être inappropriée dans le cas de logiciel couche basse.
- La seconde approche basée sur la simulation est limitée par la quantité de cas de test qui doivent être générés pour couvrir tous les cas possibles.
- La troisième approche consiste à tester le système dans des conditions réelles. Cependant, atteindre l'objectif de validation en termes de tests (des centaines de millions de kilomètres) est fastidieux voir même infaisable vu les cycles de vie de ces systèmes. De plus, la définition et la réalisation de campagnes de test de roulages sont notoirement connues comme étant peu représentatives d'un usage réel.

Il existe actuellement des tentatives pour résoudre ces problèmes, par exemple en essayant de construire une chaîne d'outils basée sur Matlab / Simulink et / ou StateMate et / ou SCADE + DesignVerifier et / ou Prover et / ou MaTeLo et / ou Teststand... Cependant il n'y a à l'heure actuelle aucune solution intégrée qui réponde à tous les points durs soulevés par COVADEC, notamment, l'intégration d'une analyse détaillée des scénarios et des composants de détection.

La nouvelle approche développée par le projet COVADEC propose différentes manières d'évaluer les fonctions ADAS et d'établir des objectifs de sécurité pour les systèmes basés sur des caméras.

4 Méthodologie et points durs

4.1 Utilisation d'une approche statistique

Les tests statistiques tels que proposés actuellement par l'outil MaTeLo doivent être réexaminés dans le contexte des ADAS. Ils doivent être combinés avec les capacités des bancs de test et des simulateurs et adaptés à l'analyse de la sûreté de fonctionnement automobile.

Pour mener les étapes de conception et de validation d'un système de détection d'objets basés sur des caméras, de très nombreuses heures de roulages sont nécessaires. Certains cas de test, identifiés comme critiques, peuvent ne jamais se produire durant la campagne de roulage. La simulation peut alors être utilisée pour couvrir ces cas. Elle devra produire des données d'images de synthèse aussi proches que possible de la réalité de telle manière que l'algorithme ADAS étudié puisse avoir un comportement identique à celui qu'il aurait en conditions de roulage réel similaires.

L'approche statistique permet alors d'évaluer la sensibilité de la réponse du système en cas de petits changements de l'environnement du véhicule. Il doit être souligné cependant que beaucoup de paramètres qui caractérisent l'environnement ne sont pas indépendants. Comme conséquence directe, les valeurs des paramètres variables qui sont exploités pour la génération des tests sur une base statistique ne peuvent pas être tirées complètement indépendamment. Par exemple, le nombre d'autres véhicules (la densité du trafic) et le comportement des conducteurs ne sont pas indépendants du type de route. Si le paramètre n'est pas indépendant, la sélection aléatoire peut conduire à générer des situations qui n'existent pas dans la réalité, voire fausser la probabilité d'apparition de certaines situations.

L'approche statistique pour générer les tests présente également deux autres difficultés :

- La première difficulté est d'ordre pratique : la matrice d'incompatibilité des paramètres doit être totalement connue. Si une situation de conduite peut être caractérisée par des douzaines voir des centaines de paramètres, connaître cette matrice n'est pas trivial.
- La seconde difficulté est d'ordre théorique : la méthode de Monte Carlo suppose que les paramètres sont indépendants, et que la distribution de probabilité souhaitée correspond précisément à l'usage. Il est donc nécessaire de voir comment corriger la méthode de Monte Carlo pour prendre en compte les dépendances des paramètres avant de sélectionner les cas de test pertinents.

4.2 Exécuter les cas de test dans un environnement ad hoc

Une fois que les cas de test ont été identifiés et générés, il faut pouvoir les exécuter de manière automatique pour pouvoir gérer un nombre important de cas de test. Il faut également pouvoir réduire le temps nécessaire à leur exécution en utilisant des moyens informatique de haute performance et parallélisés. Aujourd'hui il existe de tels outils, en particulier pour la gestion des bancs de test en HIL (Hardware In the Loop), mais aucun n'incorpore les outils dédiés aux architectures ADAS.

L'enjeu du projet COVADEC est de fournir un outil à la fois facilement accessible en termes d'interface utilisateur et de configuration (import de cas de test, accessibilité aux rapports d'exécution), de modularité (capable de gérer des cibles

d'exécution comme RTMaps et Pro-SiVIC, mais extensible à d'autres outils comme Simulink) et de performance (exécution distribuée sur des machines multiples, sans duplication des ressources logicielles pour traiter les données de capteurs ou les enregistrements 3D qui peuvent être particulièrement volumineux).

4.3 Tracabilité des exigences

Dans COVADEC, la méthodologie de test traite la vérification et la validation de l'ADAS considéré en termes de disponibilité, de fiabilité et de sécurité fonctionnelle. Certaines exigences donnent des objectifs cibles sous formes de taux de disponibilité ou de fiabilité de détection. Le respect de ces exigences doit être évalué en prenant en compte l'intégralité de la campagne de test et la tracabilité de ces exigences n'a pas vocation à être gérée de manière fine.

D'autres exigences concernent des règles de comportement de l'ADAS quand celui-ci est confronté à des perturbations connues de l'environnement (par exemple des règles d'inhibition). La tracabilité liant ce type d'exigences avec la séquence de test peut donc être exploitée pour fournir des métriques de couverture, utilisables pour l'analyse globale de la campagne de test.

5 Génération automatique des cas de test

5.1 Problématique

La validation des ADAS est un problème complexe, particulièrement pour des systèmes basés sur des caméras, car ces fonctions peuvent être confrontées à un ensemble de situations quasiment infini. Dans cet ensemble, certaines situations auront plus d'influence que d'autres sur la réponse de la fonction ADAS et certaines se présenteront plus souvent. Aussi, bien que toutes les situations ne puissent pas être couvertes par le test, il est possible de réduire l'espace à tester à une zone suffisamment petite pour pouvoir tester la fonction en ne choisissant que les situations les plus influentes et les plus représentatives qu'un ADAS puisse rencontrer.

Quelle que soit la nature des données utilisées pour la validation, réelles ou simulées, l'approche du test par les modèles (MBT, Model-Based Testing) peut être utilisée pour construire automatiquement une base de données complète de tests qui remplissent cette exigence de taille limitée tout en couvrant la plupart des situations ayant le plus d'influence sur la fonction ADAS à tester.

5.2 L'approche de test par les modèles (le MBT)

L'outil utilisé pour l'approche MBT est MaTeLo (Markov Test Logic). MaTeLo est un outil MBT, qui permet de construire un modèle du comportement attendu du système sous test et de générer ensuite, à partir de ce modèle, un ensemble de cas de test adapté à des besoins particuliers (par exemple, tester uniquement les fonctions les plus utilisées du système, ou avoir une couverture de 100% des exigences du système). MaTeLo est basé sur les chaînes de Markov. Pour la génération non-déterministe des cas de test, MaTeLo utilise les méthodes de Monte Carlo, associées à des stratégies de génération adaptées aux besoins de l'utilisateur. Pour contrer l'explosion combinatoire, le graphe généré par MaTeLo est couplé à un échantillonneur de Gibbs, qui converge à une vitesse géométrique vers la distribution cible de cas de test, comme expliqué plus tard dans ce document. Grâce à ces techniques d'accélération de test, MaTeLo permet d'obtenir une couverture maximale de la validation du système en utilisant un nombre minimum de cas de test. En d'autres termes, le nombre de kilomètres de roulage nécessaire est fortement réduit.

5.3 Résumé de la génération des cas de test

La figure suivante donne un résumé de la génération de cas de test :

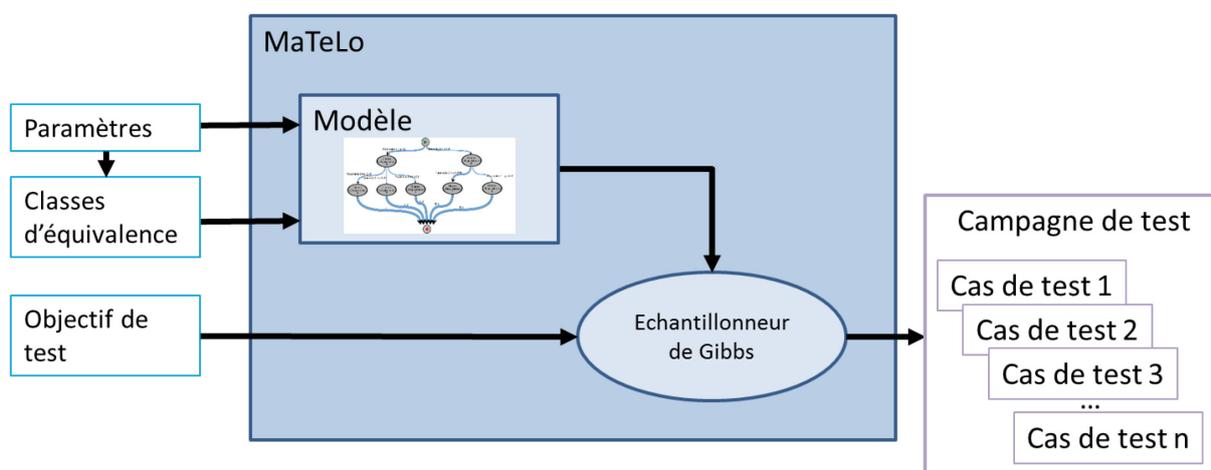


Figure 2. Schéma de principe de la génération des cas de test

Des détails supplémentaires sur l'algorithme de génération sont donnés dans les paragraphes suivants.

5.4 Méthode générale

Comme dit précédemment, la génération des cas de test pour les systèmes ADAS se heurte à la question intrinsèque d'explosion combinatoire. Typiquement, il s'agit de produire des échantillons de grands vecteurs aléatoires, dont les composantes sont possiblement dépendantes et prennent un nombre fini de valeurs, selon des probabilités données. Une contrainte importante est de générer presque toutes les situations de la façon la plus économique. En général, cette tâche peut être abordée selon deux points de vue : déterministe (via les arbres de recherche binaire) ou stochastique, via la méthode de Monte Carlo (MC) qui revient à simuler une chaîne de Markov ad hoc. Dans le projet COVADEC, nous avons choisi l'approche probabiliste, précisément basée sur l'implémentation d'un échantillonneur de Gibbs, brièvement décrit par la suite.

Dans une première étape, partant du graphe de simulation généré par MaTeLo, la méthode consiste à construire un champ aléatoire Markovien (cf. Section 5.5.1). Lorsque les paramètres (ou composantes) sont localement dépendants, ceci peut s'effectuer de façon assez simple au moyen des classiques formules de Bayes. Ensuite, les cas de test sont obtenus à l'aide de l'échantillonneur de Gibbs. En particulier, nous nous sommes efforcés d'optimiser la vitesse de convergence vers l'équilibre, laquelle est exponentielle d'après la théorie générale.

5.5 Échantillonneurs de Gibbs et champs aléatoires

Pour simuler des systèmes ayant un grand espace d'états et des distributions multidimensionnelles données - tels ceux rencontrés en physique statistique - et pour étudier certaines propriétés de l'état stationnaire, de puissantes méthodes ont été proposées dès les années 1950, en particulier, les algorithmes de Metropolis-Hastings [MET], [HAST]. Dans le contexte du traitement d'image, où une image numérisée peut être considérée comme la réalisation d'un certain champ Markovien, il faut citer un travail précurseur [GEM] qui analyse un modèle probabiliste de l'image à l'aide d'un échantillonneur de Gibbs.

5.5.1 Champ aléatoire Markovien (CAM)

Pour une introduction aux propriétés des objets mathématiques présentés ci-après, le lecteur pourra consulter par exemple [GRI] et [BRE].

Notons V le nombre de paramètres significatifs du système. On veut simuler le vecteur aléatoire $X = (X_1, X_2, \dots, X_V)$, où chaque composante X_i prend ses valeurs dans un espace fini A_i , habituellement dénommé espace des phases, avec $|A_i| = C_i$. Typiquement, $0 < C_i \approx 10$, et $V \approx 10^2$. Les variables X_i sont en général dépendantes. Une configuration $x = (x_1, x_2, \dots, x_V)$, écrite en lettres minuscules, appartient ainsi à l'espace $\Lambda = \prod_{i=1}^V A_i$.

On s'intéressera particulièrement au CAM possédant des propriétés d'interactions locales. Cette notion classique s'appuie principalement sur la notion d'espérance conditionnelle, après avoir défini une topologie convenable sur l'ensemble des indices $S = \{1, 2, \dots, V\}$ des composantes de X , qui désormais sera plutôt appelé l'ensemble des sites. Il est alors possible de définir un système de voisinages sur S (donc une topologie), qui est une famille $F = \{N_s \in S\}$ telle que, $\forall s \in S$,

$$s \notin N_s \text{ et } t \in N_s \Rightarrow s \in N_t \quad \{1\}$$

Le sous-ensemble N_s est le voisinage du site s . Dans un contexte général de graphes,

S est l'ensemble des sommets et F définit les arcs : s et t sont reliés par un arc si et seulement si ils sont voisins, i.e. $t \in N_s$.

Définition 1. Le champ aléatoire X est appelé champ Markovien par rapport au système de voisinages F si pour tout site $s \in S$ les variables aléatoires X_s et $(X_i, i \notin N_s)$, sont indépendantes conditionnellement aux $(X_i, i \in N_s)$.

Soit $\pi(\cdot)$ la mesure de probabilité du vecteur X , de sorte que $\pi(x) \stackrel{\text{def}}{=} P(X = x)$. Alors π est une *distribution de Gibbs* relative au graphe $\{S, F\}$ si elle est de la forme :

$$\pi(x) = \frac{1}{Z_T} e^{-\frac{U(x)}{T}}, \quad \{2\}$$

où $T > 0$ est la température, $U(x)$ est l'énergie de la configuration x , qui dérive d'un certain potentiel, et Z_T est la constante de normalisation. Sous une condition dite de *positivité* (Lemme de Brook, en particulier satisfait lorsque $\pi(x) > 0, \forall x \in \Lambda$), un important théorème dû à Hammersley et Clifford montre l'équivalence entre les distributions de Gibbs et les CAM, qui sont essentiellement des objets de même nature.

5.5.2 Échantillonneur de Gibbs (EG)

L'échantillonnage de Gibbs a de nombreuses applications et est devenu une méthode de simulation efficace dans l'univers de Monte Carlo. Il s'applique à toute distribution multivariée de la forme $\pi(x_1, x_2, \dots, x_V)$. Les EG peuvent être classés selon deux familles principales : Les EG à *balayage aléatoire* (EGA) et ceux à *balayage périodique* (EGP)

Échantillonneur de Gibbs à balayage aléatoire (EGA)

Le principe en est simple : à chaque étape, on choisit au hasard un site (composante) $s \in S$, puis on calcule la nouvelle valeur y_s du site correspondant selon la probabilité conditionnelle $\pi(y_s | x_j \neq s) = \pi(y_s | x_j, j \in N_s)$.

Soit α_s la probabilité of visiter le site s , avec $0 < \alpha_s < 1$ and $\sum_1^V \alpha_s = 1$.

L'algorithme construit une chaîne de Markov qui évolue de la façon suivante :

- (a) Sélectionner un vecteur initial $X(0)$ et un vecteur de probabilité $(\alpha_1, \alpha_2, \dots, \alpha_V)$.
- (b) À la j -ième itération, choisir un indice s avec probabilité α_s ;

- (c) Générer $X_s(t)$ avec la probabilité $\pi(X_s | X_j(t-1), j \in N_s)$;
 (d) Répéter l'étape (b) jusqu'à atteindre l'équilibre.

On peut montrer que la chaîne de Markov $X(t)$ est réversible, et que sa mesure invariante est précisément la distribution π du vecteur X .

Échantillonneur de Gibbs à balayage périodique (EGP)

Ici les sites sont choisis dans un ordre prédéterminé fixé, soit (s_1, s_2, \dots, s_V) , qui correspond à une permutation of $(1, 2, \dots, V)$. On génère une chaîne de Markov $Z(t)$ selon l'algorithme suivant. D'abord, on sélectionne X_{s_1} , conditionnellement à l'état courant des autres sites, puis de la même façon on choisit X_{s_2} , etc., jusqu'à X_{s_V} . Après ce balayage, on considère que la chaîne de Markov a effectué exactement une transition et il n'est pas difficile de montrer que π est sa mesure stationnaire.

Vitesse de convergence

Il est connu, d'après les résultats standards sur les chaînes de Markov, que la vitesse de convergence des EG vers l'équilibre est géométrique. Ceci signifie qu'on a l'inégalité (cf. Par exemple [BRE])

$$|X(0)\mathbb{P}^n - \pi| \leq \frac{1}{2} |X(0) - \pi| \delta(\mathbb{P})^n, \quad \{3\}$$

où \mathbb{P} désigne la matrice de transition de la chaîne de Markov obtenue à partir d'un EG, and $0 \leq \delta(\mathbb{P}) \leq 1$ est le coefficient ergodique of \mathbb{P} , dit coefficient de Dobrushin,

$$\delta(\mathbb{P}) = 1 - \inf_{i,j \in \Lambda} \sum_{k \in \Lambda} p_{ik} \wedge p_{kj}, \quad \{4\}$$

les p_{ik} désignant les éléments of \mathbb{P} .

Le calcul de bornes explicites satisfaisantes pour $\delta(\mathbb{P})$ reste un problème difficile (ouvert en général), qui varie évidemment selon le type d'EG considéré. Le lecteur pourra consulter quelques résultats globaux sur ce sujet dans [LIU] and [LEV].

De fait, la vitesse de convergence dépend fortement de la structure du CAM sous-jacent décrivant le système. Dans le projet COVADEC, nous implémenterons un EGA. Ensuite, en utilisant les propriétés spécifiques du graphe produit par le logiciel MaTeLo, nous étudierons la vitesse de convergence vue comme une fonction du vecteur de probabilité $(\alpha_1, \alpha_2, \dots, \alpha_V)$ introduit plus haut et qui est un paramètre libre.

5.6 Paramètres du modèle MaTeLo

De manière à couvrir toutes les situations auxquelles un système ADAS peut faire face, il est nécessaire de fournir un modèle de l'environnement et du contexte de conduite. L'objectif est de fournir un méta-modèle des séquences de test, prenant en compte les paramètres influents et qui exprime la diversité de situations que le système peut rencontrer. La construction de ce type de modèle implique de prendre en compte des paramètres de natures hétérogènes, ayant un impact très différent sur la scène perçue par le système.

Le modèle doit capturer les informations à propos de l'environnement dans lequel évolue l'ADAS (paysage, type de route, courbure de la route, infrastructure, etc.), de la situation de conduite (comportement du véhicule équipé et des véhicules proches), conditions climatiques (soleil, pluie, brouillard) et des éléments perturbateurs connus.

La modélisation de l'environnement doit être aussi compréhensible que possible. En fait, le modèle est supposé représenter n'importe quelles circonstances que le véhicule peut rencontrer. Aussi, si un paramètre effectivement influent est oublié, il ne sera pas pris en compte lors de la création des cas de test et donc le simulateur pourrait ne jamais générer la situation correspondant aux valeurs perturbatrices de ce paramètre. Cependant, ces situations sont potentiellement présentes dans les bases de données de vidéo réelles, même si le paramètre influent en question n'est pas explicité.

Dans le cadre du projet COVADEC, plusieurs catégories de paramètres influents ont été identifiées, elles sont présentées ci-après. Ces catégories incluent les paramètres en fonction de leur nature et permettent une transposition à n'importe quelle autre fonction.

5.6.1 Conditions climatiques

Les conditions climatiques ont un impact sur la manière dont l'ADAS percevra la scène. Cela ne comprend pas seulement la météo, mais aussi les perturbations induites par des conditions comme l'exposition de la caméra à la lumière dans la scène.

5.6.2 Structure de la route et de l'environnement

Cette catégorie inclue les caractéristiques intrinsèques de la route, c'est-à-dire les paramètres pour décrire de manière précise sa structure (courbure, topologie, nombre de voies, etc.) tout autant que son apparence (type de surface, marquage, etc.).

5.6.3 Comportement du véhicule équipé

Cette catégorie est utilisée pour exprimer le comportement du véhicule équipé dans une séquence de test, que ce soit en termes de vitesse ou de trajectoire. De plus, cette catégorie inclue les actions du conducteur qui peuvent avoir un impact sur la fonction sans impliquer un changement de trajectoire ou de vitesse (par exemple mise en route des essuie-glaces, etc.).

5.6.4 Comportement des autres véhicules

La présence d'autres véhicules peut influencer la perception qu'un ADAS a de la scène, soit sous la forme d'un véhicule cible, soit comme un élément masquant ce que l'ADAS devrait détecter. Le comportement des autres véhicules est décrit par un lot de paramètres en partie identiques à ceux utilisés pour le comportement du véhicule équipé, mais auxquels sont ajoutés des paramètres relatifs à leur position et aux changements de trajectoire qu'ils peuvent faire.

5.6.5 Piétons

Cette catégorie de paramètres décrit comment les piétons évoluent dans la scène (nombre, trajectoire, traversant la route, etc.).

5.6.6 Obstacles et perturbations

Cette catégorie inclue tous les obstacles et autres perturbations connues comme ayant un impact sur la manière dont l'ADAS percevra une situation de conduite. Les obstacles sont groupés en plusieurs sous-catégories, telles que :

- Cibles fixes posées sur le passage : cela inclue les cônes de travaux, les véhicules arrêtés, un chargement tombé sur la route ou tout autre objet pouvant être sur la route.
- Obstacles à la limite de la trajectoire du véhicule équipé : cela comprend les panneaux routiers, les glissières de sécurité, les véhicules stationnés sur le côté de la route.
- Les piétons, vélos, etc... dans des situations particulières.

5.6.7 Classes d'équivalence

L'intervalle de valeurs possibles pour chaque paramètre est découpé en plusieurs classes d'équivalence, pour deux raisons principales :

- Pour sélectionner des ensembles de valeurs ayant un impact réel sur la fonction ADAS. Cela correspond à la notion d' « intervalle », toutes les situations étant considérées comme équivalentes dans cet intervalle. Par exemple, 129 km/h et 130 km/h sont considérés comme équivalents pour l'ADAS, mais 20 km/h appartient à une classe d'équivalence différente.
- Pour gérer les dépendances entre paramètres. En effet, certaines valeurs d'un paramètres influent X pourraient ne pas être possibles ou avoir une probabilité d'occurrence différente si un autre paramètre a une valeur Y (par exemple, « nuit » et « ensoleillé » sont incompatibles ; « vitesse > 130 km/h » et « environnement urbain » ne devrait pas se produire).

Lors de la construction des campagnes de test, c'est-à-dire des ensembles de cas de test auxquels sera soumise la fonction ADAS, si un cas de test a toutes ses valeurs de paramètres dans les mêmes classes d'équivalence qu'un autre cas de test, il sera considéré comme un doublon et éliminé de la campagne. Cependant, son poids (c'est-à-dire le nombre de fois où un cas de test unique est présent dans la campagne) sera conservé pour l'évaluation de la performance totale de la fonction ADAS. Une première série de test a permis de démontrer que la réduction de l'effort de test requis pour valider une fonction ADAS en utilisant cette méthode pouvait atteindre 90%, en comparaison des méthodes actuelles à disposition. Ce résultat devra cependant être confirmé lors de la démonstration grandeur nature (construction et passage d'une campagne de test complète sur les cas d'usage) qui devrait avoir lieu en septembre 2016.

5.7 Structure du modèle MaTeLo

La structure du modèle MaTeLo pour générer les cas de test est basée sur les paramètres influents. En particulier, les dépendances entre paramètres sont modélisées comme une série de transitions dépendantes dans le modèle MaTeLo. En effet, si tous les paramètres étaient indépendants entre eux, la manière la plus naturelle pour construire le modèle MaTeLo serait de créer une simple chaîne comme suit :

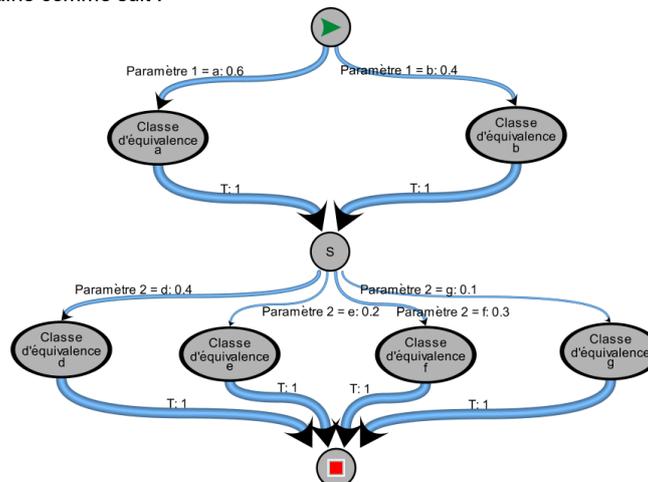


Figure 3. Modèle MaTeLo avec paramètres sans dépendance

Ce modèle en cas de paramètres dépendants n'est plus recevable, puisqu'un tel modèle générerait des cas de test qui ne peuvent se produire dans la réalité, faussant la représentativité des campagnes de test générées.

Le graphe suivant est le graphe des cas possibles du modèle MaTeLo ci-dessus. Supposons que les cas identifiés par des rectangles rouges soient des cas impossibles.

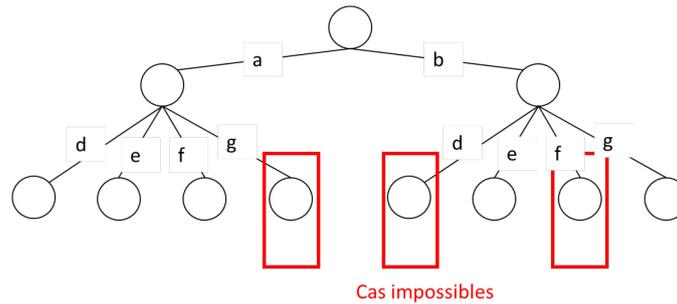


Figure 4. Graphe des possibles du modèle précédent et identification des cas impossibles

La chaîne MaTeLo telle qu'illustrée dans l'exemple suivant sera alors construite :

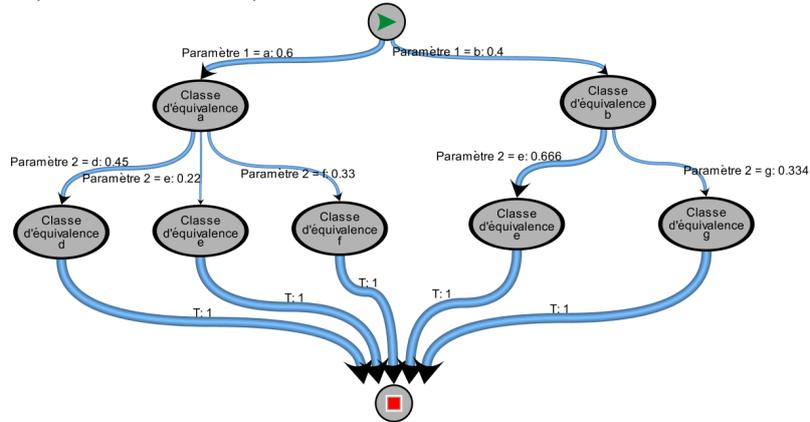


Figure 5. Modèle MaTeLo modifié pour ne pas générer les cas impossible

Les dépendances modélisées peuvent être vues de deux manières différentes :

- Soit en termes de classes d'équivalence atteignables.
- Soit en termes de probabilité de choisir une classe d'équivalence connaissant la classe d'équivalence choisie pour le paramètre précédent.

Un modèle MaTeLo est un graphe orienté, il y a donc une notion d'ordre dans le tirage des paramètres. Par conséquent, les dépendances contraignent la manière de construire le modèle, en incluant l'ordre dans lequel les paramètres apparaissent.

Quand le niveau de dépendances entre paramètres est faible (interdépendances jusqu'à 3 paramètres), les modèles MaTeLo peuvent être simplifiés en créant des macro-paramètres, ce qui peut transformer une chaîne de dépendances en chaîne indépendante. Par exemple, le moment de la journée (jour ou nuit) est lié à la luminosité ambiante de la scène, puisque la luminosité est plus sombre durant la nuit que durant le jour. Mais la luminosité pourrait être seule considérée et deviendrait un macro-paramètre, avec un plus grand nombre de valeurs possibles en considérant séparément la luminosité du jour et la luminosité de la nuit. Pour faire cela, il est nécessaire de calculer les probabilités des transitions correspondantes, qui peuvent être obtenues à partir des probabilités conditionnelles du modèle initial. Des études dans le projet ont démontré que les probabilités conditionnelles liées aux paramètres dépendants pouvaient être calculées à partir d'un modèle MaTeLo avec un algorithme très simple, en utilisant les formules de Bayes.

En conclusion, les solutions déjà obtenues dans le cadre du projet permettront de résoudre totalement le problème des dépendances entre paramètres.

5.8 Interprétation des paramètres par le simulateur

Les paramètres influents n'ont pas nécessairement une traduction directe dans le simulateur. Une coopération entre ALL4TEC et CIVITEC (ESI Group) a permis d'assurer que les paramètres fournis par MaTeLo pouvaient être correctement interprétés dans le simulateur. Ils peuvent être exprimés de manière différente dans le simulateur :

- En faisant appel à des composants de simulation existants (modèles d'objets).
- Par l'intermédiaire de ressources statiques appelées par ces composants (fichiers de ressource).
- Via la configuration de ces composants (commandes de scripts).

Pour des aspects pratiques de performance (liés aux temps de chargement et à une définition simplifiée des scénarios dans le simulateur), il est préférable de minimiser le nombre de créations de composants et de chargement de ressources. Aussi, il est possible de créer certains composants qui masquent la création de composants ou le chargement de ressources, en donnant une correspondance plus directe entre les paramètres influents et les scripts de commandes utilisés par le simulateur. L'inconvénient de cette approche est qu'elle nécessite de créer de nombreux composants très spécifiques à des sous-ensembles de scénarios.

6 Utilisation conjointe de données réelles et de simulation

Dans ce contexte, la principale innovation est de gérer des collections de données, en utilisant un modèle statistique basé sur l'outil MaTeLo. Plutôt que de conduire des millions de kilomètres dans le but de rencontrer toutes les situations qu'une fonction ADAS peut croiser durant son cycle de vie, la solution est de réduire le nombre de kilomètres en ciblant uniquement les situations les plus influentes sur le système.

Un premier travail est de collecter suffisamment de données réelles pour valider en profondeur les deux fonctions ADAS choisies comme cas d'usage. Cependant, dans le cas de certains tests de validation générés par MaTeLo, il est possible que la base de données ne couvre pas tous ces tests (en prenant en compte les différents paramètres / variables du test et la difficulté de tester des situations dangereuses en conditions réelles). L'outil de simulation doit venir combler les manques de la base de données en générant des données capteurs réalistes pour ces cas de test. Idéalement, il serait souhaitable que la plateforme de simulation génère des scènes, scénarios et données capteurs directement à partir de la définition de ces tests. En pratique, certaines étapes préparatoires doivent être faites avant l'exécution des tests, en particulier pour construire des éléments spécifiques de l'environnement. L'optimisation de ce processus et les opportunités de générer ces éléments tout en exécutant les tests est actuellement étudiée dans le cadre du projet COVADEC.

7 Chaîne d'outils de test utilisée pour la simulation

Une fois que les cas de test ont été définis, il est nécessaire de les injecter dans une chaîne d'outils de test pour les soumettre à l'ADAS en cours de validation. Cette chaîne d'outils est composée de :

- Un simulateur de scénario, d'environnements et de séquences vidéo (Pro-SiVIC – CIVITEC / ESI Group)
- Un enregistreur de données ADAS haute performance (Intempora dataloggers)
- Un framework pour l'exécution en temps réel ou simulé des algorithmes ADAS (RTMaps / Intempora)
- Un simulateur d'architecture matérielle ADAS (Rabbits / TIMA)
- Un serveur d'exécution automatique des tests (I-DEEP / Intempora)

Pro-SiVIC est un logiciel de simulation d'environnement spécialisé dans le rendu avancé des capteurs ADAS (caméras, lidars, radars, GPS, systèmes de communication). Il offre des modèles de capteurs complexes ainsi que d'environnements en prenant en compte de nombreuses caractéristiques physiques et électroniques (par exemple, pour une caméra, modélisation de la distorsion, bruit, conditions atmosphériques et météorologiques, conditions de lumières, etc...). L'aspect clé pour le processus de validation est la capacité de contrôler les caractéristiques des conditions environnementales. Pro-SiVIC permet également d'intégrer des modèles de dynamiques véhicules, de programmer des situations complexes de conduite dans un environnement routier complet, d'intégrer des animations d'objets (tels que des piétons par exemple). Pro-SiVIC peut fonctionner en temps réel ou en temps simulé, ce qui permet de réaliser le test et la validation de fonctions ADAS avec ou sans humain ou ECU dans la boucle.

RTMaps est un framework logiciel modulaire (basé sur des composants), permettant le développement rapide et l'exécution optimisée d'applications temps réelles qui doivent gérer, traiter et fusionner de nombreux flux de données capteurs à haute bande passante, asynchrones et hétérogènes, tels que des caméras, lidars, radars, bus CAN, GPS, IMUs, communications V2V and V2I, etc... RTMaps offre également des possibilités d'enregistrement de données en provenance de n'importe quel type de capteurs ADAS, puis de les rejouer de manière synchronisée, en temps réel et en temps simulé, afin de permettre le développement d'applications de perception, de fusion de données, de communication, de prise de décisions et de contrôle-commande (développements, tests, validation et analyse comparative). RTMaps peut également être connecté à des outils de simulation et/ou de contrôle commande tels que Pro-SiVIC.

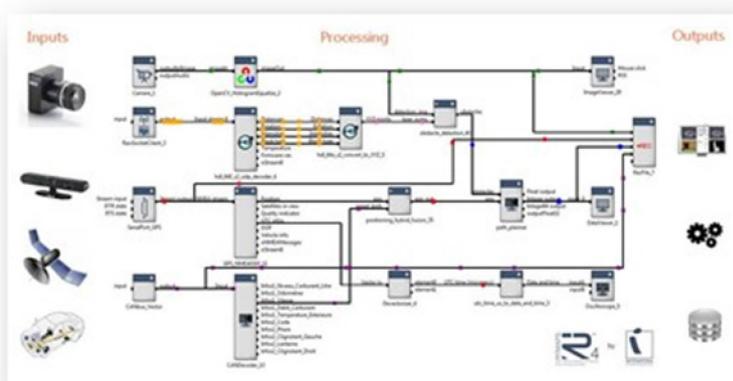


Figure 6. Exemple de diagramme RTMaps

I-DEEP est un serveur d'exécution automatique de test dédié à la validation de fonctions de perception et de décision pour les ADAS, particulièrement les fonctions basées sur la vision :

- I-DEEP peut stocker des données capteurs enregistrées et leurs données de vérité terrain associées et/ou des ressources de scénarios de simulation, et peut également accueillir des algorithmes de traitement d'images /

traitement de signal / fusion de données à tester (intégrés dans RTMaps comme plugins), et permet ensuite de définir et exécuter automatiquement les nombreux cas de test sur plusieurs clusters de calculateurs.

- I-DEEP offre également une approche double pour la validation de fonction ADAS en utilisant soit la simulation, soit le jeu de données préalablement enregistrées.

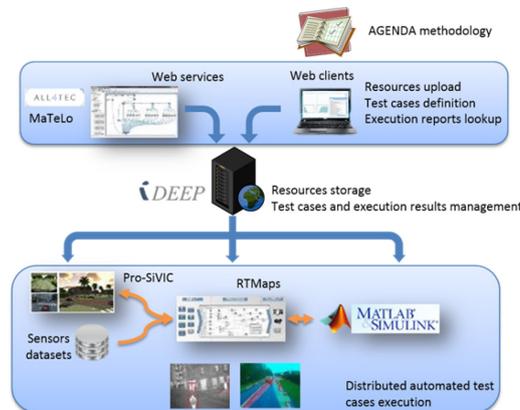


Figure 7. Schéma de la plateforme de validation

Rabbits est un simulateur matériel/logiciel rapide capable de co-simuler des systèmes multiprocesseurs sur une puce. Il tire parti de QEMU pour la modélisation du processeur, et SystemC TLM pour la modélisation des composants matériels. Rabbits peut gérer plusieurs paramètres, comme un nombre variable de processeurs, de taille mémoire, de disponibilité de cache et de taille de cache, il supporte des instructions spécifiques, par exemple SIMD ou à virgule flottante, etc... Bien qu'assez abstraite, la technologie de simulation permet d'évaluer le temps d'exécution sur les systèmes matériels/logiciels. Dans ce projet, Rabbits est utilisé comme le simulateur de l'architecture matérielle des ADAS dans le but d'explorer l'espace de conception de la plateforme matérielle et de l'implémentation logicielle. Rabbits a été intégré dans l'ensemble du flux de conception comme un composant RTMaps afin de cibler la validation d'une architecture optimisée. RTMaps fait un pré-traitement des images générées par Pro-SiVIC et les envoie, à travers une interface d'I-DEEP, à Rabbits. Rabbits exécute le logiciel cross-compilé de la fonction ADAS, qui lit les images par le biais d'une caméra virtuelle lié à l'interface d'I-DEEP, effectue des calculs sur ces images et renvoie à RTMaps, à travers une interface série virtuelle, les informations sur l'environnement renvoyées par la fonction de perception de l'ADAS simulée. Ensuite RTMaps se charge de fournir au reste de la chaîne de traitement ces informations, de telle manière que la décision appropriée puisse être prise par le système.

8 Conclusion

Plusieurs bénéfices sont attendus du projet COVADEC :

- Améliorer la connaissance globale de l'applicabilité de la norme ISO 26262 pour la conception et la validation des capteurs ADAS, et faire la lumière sur ses limites afin de proposer des solutions.
- Réduire le nombre de kilomètres de roulages nécessaires pour la validation des ADAS, en utilisant un modèle statistique et en optimisant les plans de tests en utilisant la notion de classes d'équivalence.
- Construire une plateforme de validation ADAS (Model in the Loop et Software in the Loop), combinant des données d'environnement réelles et simulées.

A ce jour, la méthodologie a été entièrement développée et testée sur des cas d'étude réduits. A l'heure de l'écriture de ce document, la plateforme de validation est en phase d'intégration, notamment pour la validation avec des données réelles. Le simulateur Pro-SiVIC est en cours d'amélioration afin de fournir des environnements routiers modulables et conformes à ceux demandés par les campagnes de test. L'implémentation de l'algorithme de l'échantillonneur de Gibbs dans l'outil de génération des cas de test (MaTeLo) est également en cours.

Le projet COVADEC est dans sa dernière phase, la campagne de démonstration grandeur nature devrait avoir lieu en septembre 2016. La communication que nous présenterons à Lambda Mu sera alors enrichie des derniers résultats obtenus.

Bibliographie

- [BRE] P. Brémaud (1995), *Markov Chains*, Springer.
 [GEM] S. & D. Geman (1984), *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6, 721-741.
 [GRI] G. Grimmett (2010), *Probability on Graphs*, CUP.
 [HAST] Hastings (1970), *Biometrika*, 57 (1), 97-109.
 [LEV] R.A. Levine & al. (2006), *Journal of Multivariate Analysis*, 97, 2071-2100.
 [LIU] J.S. Liu & al. (1995), *Journal of the Royal Stat. Society, Series B*, 57 (1), 157-169.
 [MET] Metropolis & al., 1953, *The Journal of Chemical Physics* 21, 1087-1092.

ⁱ Autres auteurs contributeurs

P. DE SOUZA (CIVITEC (ESI Group)), X. ROUAH (INTEMPORA), M. PFEIFFER (MAGILLEM), S. GÉRONIMI (PSA), F. PÉTROTTOT et A. TEMANI (TIMA – INPG), S. AHIAH (VALEO)