

DYNA, véhicule expérimental pour la Dynamique du Véhicule

Gérald Dherbomez, Thierry Monglon, Paul Crubillé, Bin Wang,
Alessandro Correa Victorino, Ali Charara

Heudiasyc UMR CNRS 7253, Université de Technologie de Compiègne, France

*gerald.dherbomez@hds.utc.fr; thierry.monglon@utc.fr; paul.crubille@hds.utc.fr;
bin.wang_bis@hds.utc.fr; alessandro.victorino@hds.utc.fr; ali.charara@hds.utc.fr*

Résumé

La connaissance de certains paramètres/variables qui décrivent la dynamique de véhicules est essentielle dans le fonctionnement de systèmes d'assistance à la conduite, et en particulier pour la stabilisation et le contrôle du comportement dynamique. Des variables/paramètres essentiels, comme les forces de contact pneu-route, l'angle de dérive et l'adhérence ne peuvent être mesurés, ou déployés en grande série, pour des raisons économiques ou techniques. Nous travaillons, au laboratoire Heudiasyc, sur le développement des observateurs d'état appliqués à l'estimation de ces variables, en temps-réel pendant le déplacement du véhicule, à partir de données fournies par des capteurs à bas coût embarqués dans le véhicule. Ces observateurs ont été validés en temps-réel à bord du véhicule démonstrateur DYNA dans différentes situations critiques de conduite.

Le laboratoire Heudiasyc développe une plateforme expérimentale PACPUS [6] composée actuellement de 6 véhicules expérimentaux instrumentés pour des applications de localisation, de perception de l'environnement, de navigation robotique, de dynamique du véhicule et d'automatisation de la conduite. Le véhicule démonstrateur DYNA, spécifiquement instrumenté pour mener des recherches en dynamique du véhicule fait partie de la plateforme PACPUS et sera décrite dans la suite de cet article. Les résultats expérimentaux des observateurs testés sur DYNA seront présentés en conclusion.

Mots-clés: *véhicule intelligent, dynamique du véhicule, observateurs d'état, capteurs logiciels, systèmes temps réel embarqués.*

1 Le véhicule expérimental DYNA

1.1 Finalité du démonstrateur DYNA

La connaissance de certains paramètres qui décrivent la dynamique du véhicule est essentielle dans le fonctionnement de systèmes d'assistance à la conduite, ou en tant qu'entrées des systèmes de stabilisation et contrôle du véhicule. Notamment, l'angle de dérive et les forces de contact pneumatiques-chaussée sont utiles pour le développement d'applications de stabilisation de la trajectoire des véhicules. Ces données sont difficilement accessibles à la mesure (coûts des capteurs très onéreux), ainsi le laboratoire a mis au point le démonstrateur DYNA pour pouvoir tester et valider des observateurs d'état capables d'estimer les grandeurs de la dynamique du véhicule. DYNA permet à la fois de récupérer en temps réel les données présentes de série sur le véhicule et d'enregistrer les informations de vérité terrain provenant des capteurs de validation.



Figure 1: Le véhicule expérimental DYNA et ses équipements embarqués.

1.2 Description des équipements embarqués

Le véhicule DYNA est une Peugeot 308 SW à carburant essence de 150CV. Le véhicule a été complètement paramétrisé (masses sur chaque roue, position du centre de gravité, raideurs de suspension, ...), ces paramètres étant ensuite utilisés par les observateurs d'état. Il est instrumenté avec différents types de capteurs listés ci-dessous.

Capteurs de vérité terrain utilisés pour la validation :

- 4 roues dynamométriques Kistler RoaDyn S625 fournissant pour chaque roue les 3 forces de contact pneumatiques-chaussée, les 3 moments de roue, la vitesse roue et l'angle roue.
- 1 capteur de vitesse 2 axes Correvit S400 fournissant la norme de vitesse et l'angle de dérive.
- 1 récepteur GPS fonctionnant en mode RTK permettant de localiser le véhicule avec une précision de quelques centimètres.

Capteurs pour la dynamique du véhicule utilisés par les observateurs d'état :

- Données accessibles sur le bus CAN natif du véhicule :
 - les vitesses des roues, le régime moteur et l'accélération longitudinale,
 - la vitesse de lacet, la pression de freinage et l'accélération latérale issues de l'ESP,
 - l'angle volant donnant une image de l'angle de braquage,
- Capteurs de hauteur de caisse Corrsys HT500 qui fournissent la hauteur de la caisse au niveau des 4 roues.

- Capteurs à fil pour la mesure des débattements de suspension arrière.
- Une centrale inertielle Crossbow VG700AB fournissant les accélérations et vitesses de rotation du véhicule dans les 3 axes.

Capteurs extéroceptifs :

- Une caméra scène avant.
- Un système Mobileye fournissant en temps réel sur le bus CAN du véhicule une liste des obstacles détectés (piétons, véhicules, ...) ainsi que la position du véhicule par rapport aux marquages au sol latéraux.
- Un télémètre laser multi-faisceaux et multi-echos Ibeo LUX 8L.

1.3 Architecture du système d'acquisition

Si une part importante des données utilisées par les estimateurs est issue de capteurs déjà intégrés au véhicule et accessible sous forme numérique sur les différents bus CAN, les données issues des capteurs de "vérité terrain", permettant de valider les estimations nécessitent une chaîne d'acquisition de qualité suffisante.

Nous avons réalisé un tel système d'acquisition capable d'acquérir 16, 32 ou 48 voies analogiques utilisant un convertisseur 24 bits. Il est constitué d'un calculateur de la société UEI animé par un processeur PowerPc à 200 Mhz exécutant un système d'exploitation temps réel Xenomai. Les cartes d'acquisition s'empilent sous la carte mère et l'ensemble se glisse dans un petit boîtier cubique. Nous utilisons des cartes A/D DNA-AI-217.

Un protocole client/serveur met à disposition sur le réseau Ethernet interne au véhicule les données brutes acquises à 4 kHz ou les données filtrées selon 2 options aux choix des clients.

1.3.1 Protocole de communication

Pour obtenir les données, les clients envoient au serveur une requête spécifiant

- l'ensemble des échantillons qu'il veut obtenir. Pour chacun, il précise le canal, le filtre et l'indice par rapport à l'échantillon courant, le serveur envoie des trames à 200 Hz et stocke 20 échantillons (acquis à 4kHz) entre 2 émissions.
- la fréquence à laquelle il souhaite recevoir des trames décrite comme un diviseur de la fréquence de base.
- et le nombre total de trames échantillons qu'il demande. Pour un fonctionnement continu, plutôt que d'indiquer un très grand nombre de trames, on renouvellera périodiquement la requête.

1.3.2 Filtre Tchébichev ordre 5

Le filtre Tchebichev d'ordre 5 [7] a une fréquence de coupure de 35 Hz et est sans perte de niveau jusqu'à 20 HZ. Dans les figures, l'axe des "x" est en échantillons.

La valeur filtrée est calculée pour chaque échantillon et pour toutes les voies par une tâche dédiée.

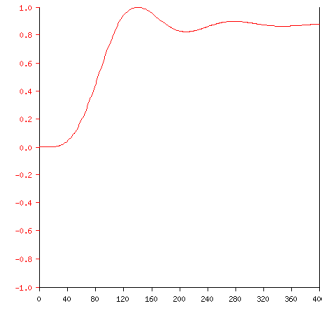
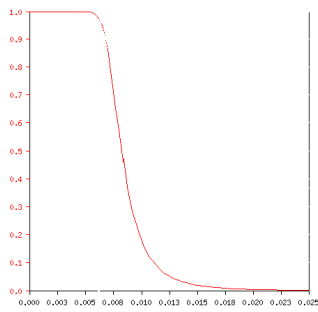


Figure 2: Réponse en fréquence du filtre Tchébichev

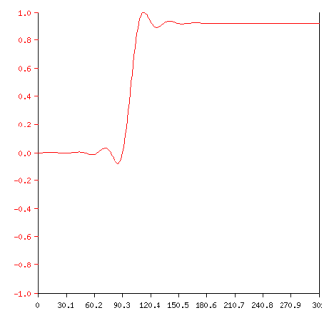
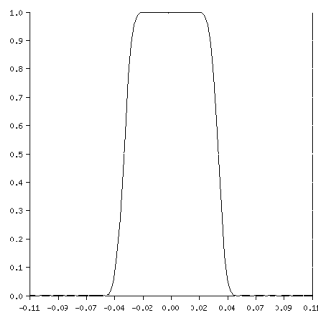


Figure 4: Réponse en fréquence du filtre FIR.

Figure 5: Réponse sur Dirac du filtre FIR

1.3.3 Filtre FIR

Le filtre FIR [7] a une fréquence de coupure de 145 Hz et est sans perte de niveau jusqu'à 100 Hz. Son calcul pour 1 échantillon prend en compte 201 échantillons et est bien trop coûteux pour qu'il soit possible de le calculer systématiquement. Il est calculé seulement pour les échantillons et les voies demandés par un client.

1.3.4 Implémentation

La stabilité temporelle des acquisitions et du logiciel du système d'acquisition est garantie par l'utilisation de Xenomai, le sous-noyau temps réel de Linux. Les 2 premières tâches sont effectivement des tâches temps réel, les 2 suivantes sont des tâches réseaux utilisant les services de communications Linux et sont donc incapables de respecter des échéances temporelles précises.

- Une tâche d'acquisition à 4 kHz de priorité maximale, périodique.
- Une tâche de filtrage qui applique sur toute les voies le filtre Tchebichev d'ordre 5.
- Une tâche d'émission des messages qui, pour chaque requête à servir, envoie le message avec les données requises. Elle calcule le filtre FIR sur les échantillons pour lesquels ce filtre est spécifié.
- Une tâche qui attend les requêtes des clients.

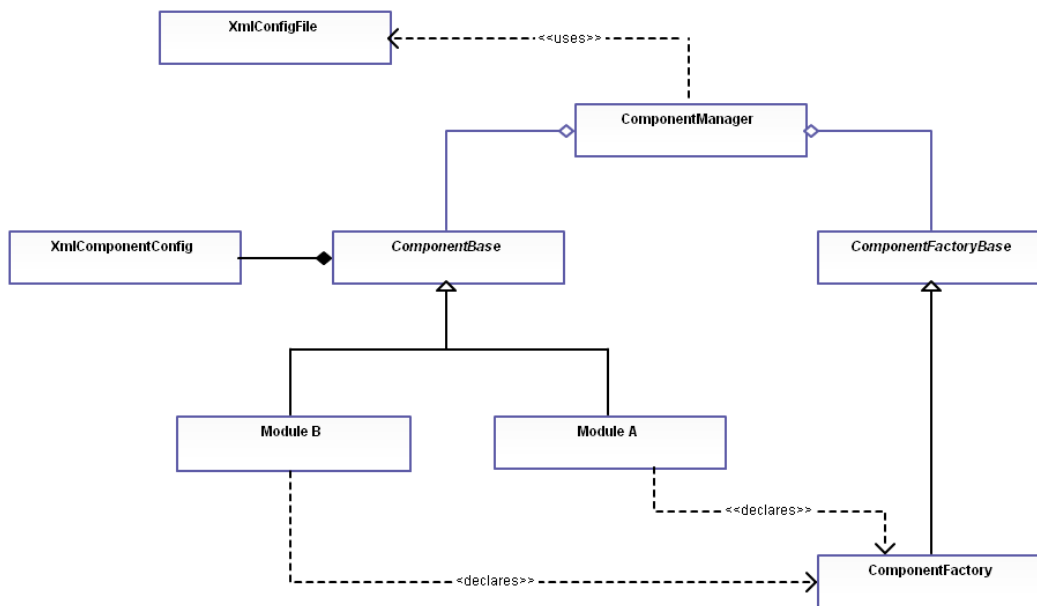


Figure 6: Diagramme de classe d'une application développée avec le framework PACPUS.

2 L'architecture logicielle

2.1 Environnement de prototypage rapide : le framework PACPUS

Le laboratoire a développé un environnement de prototypage rapide permettant de concevoir des modules logiciels et de les tester en temps réel sur des véhicules. Il s'agit du framework PACPUS disponible en open-source sous licence libre Cecill-C. Le framework PACPUS a été développé suivant le principe de conception orientée composants afin d'offrir aux utilisateurs plus de modularité dans leurs développements.

L'environnement de prototypage rapide PACPUS est développé en langage C++ afin de :

- garder une compatibilité très serrée avec le C et ainsi faciliter le développement de couches basses (drivers, interfaces capteurs, appels systèmes, ...),
- pouvoir utiliser des mécanismes de la programmation orientée objet, répondant au besoin de modularité,
- obtenir un code compilé offrant des vitesses d'exécution plus élevées permettant de se rapprocher des contraintes temps réel.

Le framework PACPUS utilise l'API Qt pour les interfaces graphiques et peut-être interfacé avec d'autres environnement de développement pour la fusion multi-capteurs [9] [8] [10]. Le diagramme de classe de la Figure 6 présente l'architecture typique d'une application utilisant le framework PACPUS avec ici deux composants Module A et Module B qui sont gérés par la classe ComponentManager.

2.2 Capteurs logiciels : Application à l'observation d'état pour la dynamique du véhicule

Les observateurs d'état permettent de mesurer des grandeurs difficilement accessibles et notamment les données de la dynamique du véhicule comme l'angle de dérive ou les forces de contact

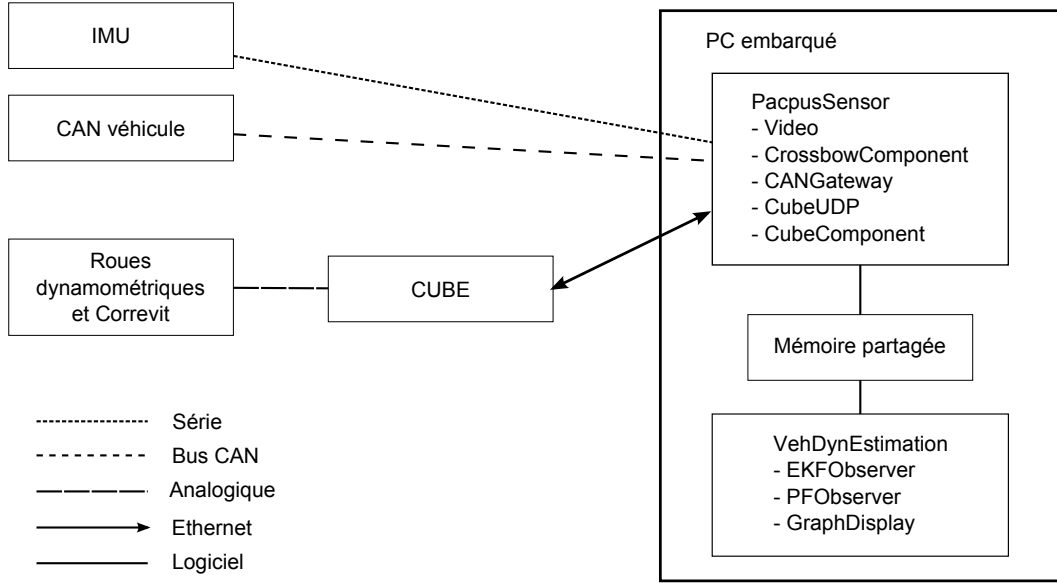


Figure 7: Architecture logicielle de l'application finale d'observation d'état.

pneumatique/chaussée grâce à un modèle numérique du système et des données issues de capteurs physiques. Ces observateurs, également appelés capteurs logiciels, ont été implémentés dans l'environnement PACPUS sur le véhicule DYNA.

3 Représentation de l'observateur non linéaire

3.1 Description de l'observateur

Dans le but d'embarquer les observateurs, nous avons opté pour deux observateurs non linéaires: le filtre de Kalman étendu (EKF) et le filtre particulaire (PF). Les observateurs sont dédiés à estimer respectivement les forces latérales et l'angle de dérive pendant le déplacement du véhicule en temps-réel. Nous présenterons par la suite le principe de ces deux technologies d'observation. Considérons un modèle non linéaire général :

$$\begin{aligned} x_t &= f(x_{t-1}) + u_t, \\ y_t &= h(x_t) + v_t, \end{aligned} \quad (1)$$

où u_t et v_t sont respectivement le bruit d'état et le bruit d'observation. Dans la formulation (1), le vecteur d'état x_t est composé par : la vitesse du véhicule, la vitesse de lacet, l'angle de dérive, la force latérale au niveau des quatre roues ainsi que la somme des forces longitudinales au niveau de l'essieu avant F_x :

$$X = [V_g, \psi, \beta, F_{y11}, F_{y12}, F_{y21}, F_{y22}, F_{x1}]. \quad (2)$$

Les variables de mesure et d'entrée sont données comme suit:

$$\begin{aligned} Y &= [V_g, \psi, a_x, a_y], \\ U &= [\delta, F_{z11}, F_{z12}, F_{z21}, F_{z22}]. \end{aligned} \quad (3)$$

Dans notre étude, la représentation discrète du modèle est faite par la méthode d'Euler, ensuite les variables d'état de la dynamique du véhicule peuvent être estimées par les techniques d'observation EKF et PF.

L'EKF est toujours utilisé pour sa simplicité, dans [1] et [2]. La méthode de EKF utilise un développement de Taylor au premier ordre afin de calculer l'approximation de l'équation de mesure. Après ce processus de linéarisation, le filtre de Kalman classique peut être utilisé pour estimer les états dans le processus de l'EKF.

Algorithme de filtre de Kalman étendu (EKF):

- Équations de mise à jour de l'état:

$$\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}), \quad (4)$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t' + Q, \quad (5)$$

- Équations de mise à jour de mesure:

$$K_t = P_{t|t-1} H_t' (H_t P_{t|t-1} H_t' + R)^{-1}, \quad (6)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t (y_t - h(\hat{x}_{t|t-1})), \quad (7)$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1}. \quad (8)$$

où F_t et H_t sont les matrices jacobiennes d'état et de mesure à l'étape t :

$$F_t = \left. \frac{\partial f(x_{t-1})}{\partial x_{t-1}} \right|_{x_{t-1} = \hat{x}_{t-1|t-1}}, \quad (9)$$

$$H_t = \left. \frac{\partial h(x_t)}{\partial x_t} \right|_{x_t = \hat{x}_{t|t-1}}.$$

Le PF peut être considéré comme une approximation d'un filtre bayésien récursif. Le principe est de mettre en place un filtre bayésien récursif par la simulation de Monte Carlo. Les densités de probabilité impliquées dans le filtre bayésien sont représentées par un ensemble d'échantillons aléatoires avec leurs pondérations associées. L'avantage du PF réside dans sa capacité de traitement du modèle non linéaire avec une précision satisfaisante [3]. Nous avons choisi un algorithme de PF basé sur la technique de ré-échantillonnage (SIR). En considérant la qualité de ré-échantillonnage et la complexité des calculs, plusieurs comparaisons sont faites entre des algorithmes de ré-échantillonnage fréquemment utilisées dans le filtrage particulaire dans [4]. Par conséquent, nous utilisons le ré-échantillonnage résiduel afin de réduire la complexité de calcul et effectivement éviter le problème de dégénération.

Algorithme de filtre Particulaire (PF) :

- Initialisation, $t = 0$: for $i = 1, \dots, N$,
 - Échantillon x_0^i d'une densité à priori $p(x_0)$.
 - Réglage des poids $\omega_0^i = \frac{1}{N}$.
- Échantillons des distributions : au moment $t \geq 1$, pour $i = 1, \dots, N$,
 - Échantillon $x_t^i \sim q(x_t | x_{t-1}^i, y_t)$.
 - Calcul des poids $\tilde{\omega}_t^i = \tilde{\omega}_{t-1}^i \frac{p(y_t | x_t^i) p(x_t^i | x_{t-1}^i)}{q(x_t^i | x_{t-1}^i, y_t)}$.
 - Normalisation des poids $\omega_t^i = \frac{\tilde{\omega}_t^i}{\sum_{i=1}^N \tilde{\omega}_t^i}$.

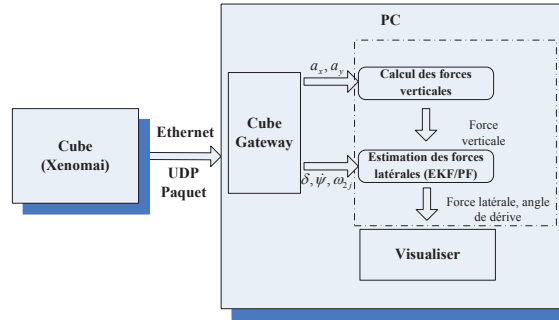


Figure 8: Le système embarqué sur le véhicule expérimental DYNA : la partie en pointillée représente le processus d’observation embarqué.

- Ré-échantillons résiduels : au moment $t \geq 1$, ré-échantillonnage des particules \tilde{x}_t^i à partir de x_t^i selon les poids ω_t^i .
 - Conserver $k_t^i = \lfloor N\omega_t^i \rfloor$ copies de particules x_t^i à la nouvelle distribution.
 - Ré-échantillonner $m_t = N - \sum k_t^i$ particules dans $\{x_t^i\}$ en faisant \tilde{m}_t copie de particules x_t^i , la probabilité de sélection de \tilde{x}_t^i est proportionnelle à $\tilde{\omega}_t^i = N\omega_t^i - k_t^i$ ce qui est décidée par la distribution multinomiale résiduelles dans la méthodologie de ré-échantillonnage multinomial.

Les résultats expérimentaux, montrés dans la section XX, obtenus à partir des données acquises avec le démonstrateur DYNA, permettront de valider la performance de ces deux différentes techniques d’observation.

3.2 Processus d’estimation

Dans cette section, nous présentons le processus d’estimation qui est embarqué dans notre véhicule (Figure 1). Le diagramme de blocs représenté dans la Figure 8 montre les différentes étapes du processus d’observation embarqué, ainsi que le flux de données. Le bloc ”cube-Xenomai” représente un PC industriel sur lequel sont branchés les capteurs embarqués. Le cube envoie les paquets UDP (User Datagram Protocol) par l’Ethernet sur un autre ordinateur embarqué, où les modules d’observation sont exécutés. Le composant ”cube-gateway” a pour rôle de recevoir les paquets UDP, les décoder et ensuite d’envoyer les structures de données au module applicatif, qui inclut les algorithmes d’observation écrits en C++.

Dans la partie pointillée, le premier bloc calcule la force verticale à partir des accélérations latérale et longitudinale issues du bus CAN. Ces forces seront utilisées comme des entrées dans le deuxième bloc, qui estime la force latérale et l’angle de dérive. Les processus d’estimation EKF et PF sont développés en respectant les contraintes temps-réel de mise à disposition du résultat, le système d’acquisition est aussi développé comme un module du système embarqué en C/C++ avec le framework PACPUS. Ce module est entièrement développé avec l’aide du ”Service Plateformes Technologiques” du laboratoire Heudiasyc.

4 Résultats expérimentaux

Cette étude présente un essai slalom sur une route mouillée pour l’évaluation de la performance de l’observateur dans [5]. Les figures 10, 11 et 12 montrent les résultats en temps réel des observateurs EKF et PF comparés aux capteurs de vérité terrain.

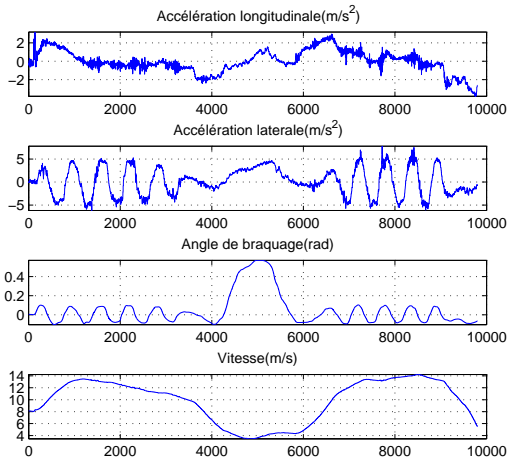


Figure 9: Accélération, vitesse et angle de braquage pour l'essai sur sol mouillé

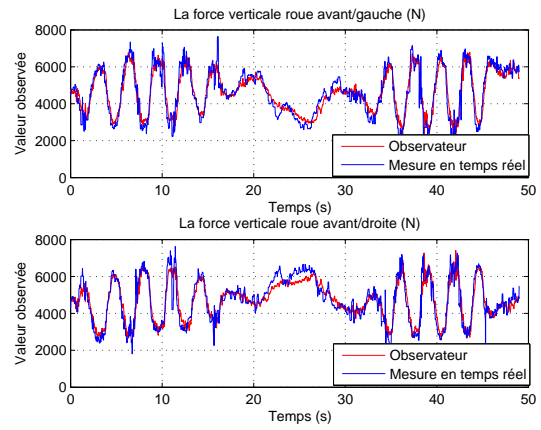


Figure 10: Forces verticales aux pneus avant

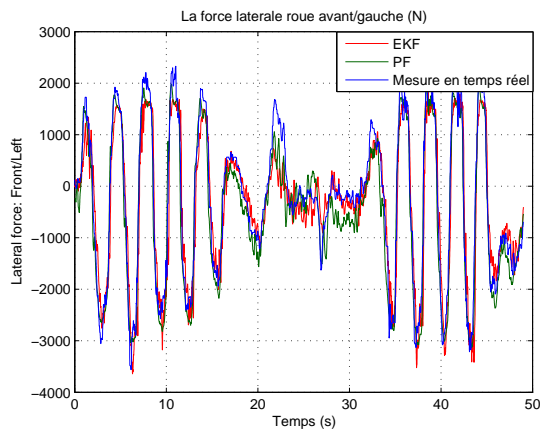


Figure 11: Force latérale au pneu avant/gauche

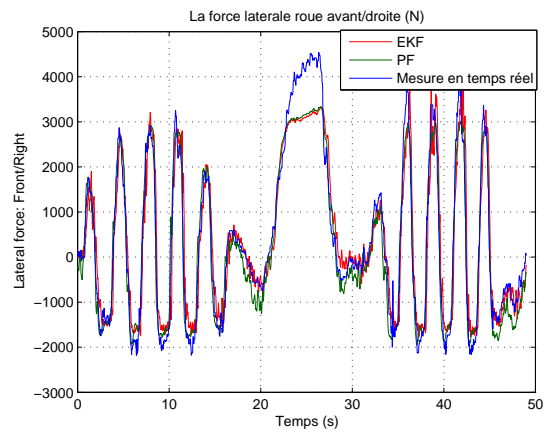


Figure 12: Force latérale au pneu avant/droite

5 Conclusion

Le véhicule DYNA est complètement opérationnel pour mener à bien des expérimentations dans le domaine des véhicules intelligents, de l'aide à la conduite et plus particulièrement pour des essais autour de la dynamique du véhicule. Ce démonstrateur est actuellement utilisé pour la validation des développements réalisés dans le cadre du projet franco-allemand VERVE et a permis de réaliser les expérimentations du projet régional SEDVAC et du projet ANR JCJC Percoive.

6 Remerciements

Le démonstrateur DYNA a été financé par le Conseil Régional de Picardie, le FEDER et le CNRS dans le cadre du CPER (Contrat de Projets Etat-Région).

References

- [1] J. Kim, "Identification of lateral tyre force dynamics using an extended kalman filter from experimental road test data", *Control Engineering Practice*, vol. 17, no. 3, pp. 357-367, 2009.
- [2] G. Baffet, A. Charara, D. Lechner and D. Thomas, "Experimental evaluation of observers for tire-road forces, sideslip angle and wheel cornering stiffness", *Vehicle System Dynamics*, vol. 46, no. 6, pp. 501-520, 2008.
- [3] T. A. Wenzel, K. J. Burnham, M. V. Blundell and R. A. Williams, "Dual extended Kalman filter for vehicle state and parameter estimation", *Vehicle System Dynamics*, Vol. 44, No. 2 153-171, 2006.
- [4] J. D. Hol, T. B. Schon and F. Gustafsson, "On Resampling Algorithms for Particle Filters", *IEEE Nonlinear Statistical Signal Processing Workshop*, pp. 79-82, 2006.
- [5] B. Wang, A. Victorino and A. Charara, "Estimation des variables de la dynamique latérale sur chaussée mouillée: validation expérimentale sur véhicule démonstrateur", 5èmes Journées Doctorales / Journées Nationales MACS, 2013.
- [6] Site web de la plateformes Véhicules Intelligents PACPUS du laboratoire Heudiasyc : <http://www.hds.utc.fr/pacpus>.
- [7] T. Fisher, "Interactive Digital Filter Design", <http://www-users.cs.york.ac.uk/~fisher/>, 2000.
- [8] C. Tessier, C. Cariou, C. Debain, F. Chausse, R. Chapuis, and C. Rousset, "A real-time, multi-sensor architecture for fusion of delayed observations: Application to vehicle localisation," in 9th International IEEE Conference on Intelligent Transportation Systems, Toronto, Canada, September 17-20 2006, pp. 1316-1321.
- [9] B. Steux, P. Coulombeau, C. Laugeau (2000, October), "RTmaps: a framework for prototyping automotive multi-sensor applications", *Proc. Intelligent Vehicles Symposium*.
- [10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, A. Y. Ng, (2009, May), "ROS: an open-source Robot Operating System", *ICRA workshop on open source software* (Vol. 3, No. 3.2).