Open SD2S

1. DESCRIPTION

OpenSD2S (Open Scalable Driving Simulation Software) a été une initiative de l'ENSAM en collaboration avec Renault (Centre Technique de Simulation) pour développer un logiciel de simulation de conduite sous licence LGPL pour le monde académique et de manière plus générale pour les environnements de simulation de conduite. Le projet est sur SourceForge : http://opensd2s.sourceforge.net.

2. HISTORIQUE

Une première version a été développée en 2010 (Filliard, N., et al., 2010) en collaboration entre Arts & Métiers ParisTech (Institut Image) et Renault (Centre Technique de Simulation). La même année, une première version a été mise en ligne sur la plateforme de diffusion de logiciels Open source Sourceforge et un communiqué de presse a eu lieu dans le cadre de Driving Simulation Conderence Europe 2010. En 2012, la collaboration autour du projet s'est étendue à Theoris, une nouvelle version a été mise en ligne et une démonstration à Driving Simulation Conference Europe 2012 a eu lieu.

3. ARCHITECTURE

Les principaux principes d'OpenSD2S sont la modularité et la capacité de mise à l'échelle.

3.1. MODULARITE

La modularité est réalisée par un design orienté composant qui permet de rendre le logiciel plus facilement maintenable par l'isolation des fonctionnalités indépendantes. Cette approche est particulièrement adaptée pour intégrer des modules hétérogènes réalisant des tâches parallèles. Chaque librairie spécifique est encapsulée dans un module exposant une interface standardisée, permettant de disposer d'une couche d'abstraction masquant son fonctionnement. Ce couplage lâche est plus flexible et favorise la testabilité du système du fait que les évolutions du code soient localisées, ce qui est une nécessité critique dans les phases de conception de telles applications composites. L'addition de nouvelles fonctionnalités est réalisée très simplement, par l'ajout de modules facilitant ainsi les évolutions futures. Cela permet également de proposer des alternatives pour chaque module (ex : différents moteurs 3D, modèles de dynamique véhicule). Notamment, les modules basés sur des librairies Open Source peuvent être remplacés de manière transparente par des développements internes ou des librairies propriétaires, pourvu qu'un kit développement soit disponible.

Dans la version actuelle d'OpenSD2S, les modules sont caractérisés par des processus individuels du système d'exploitation, coordonnés en utilisant MPICH (Gropp W.D., 1995), une interface de passage de messages compatible MPI (Clark L. et al., 1994). L'utilisation d'une couche de communication inter-modules basée sur MPI rend possible l'exécution optimisée et transparente du simulateur sur une variété d'architecture matérielles, de la grappe de PC reliés en réseaux aux PC multi-cœurs.

3.2. CAPACITE DE MISE A L'ECHELLE

La capacité de mise à l'échelle est nécessaire pour le logiciel puisse s'adapter à une grande variété de simulateur, de la configuration low-cost utilisant des périphériques de jeux vidéo, au simulateur professionnel haute performances requérant une puissance de calcul accrue. A cet effet, les différents modules sont organisés selon une architecture en étoile (Figure 1 : Architecture d'OpenSD2S) : Chaque module est connecté au noyau central, qui gère la base de donnée contenant toutes les données de la simulation (ex : la position de la voiture, le point de vue, l'angle du volant). Les modules peuvent mettre à jour ces données ou les consulter sur requêtes. Cette synchronisation lâche permettant de découpler les modules rend possible la cohabitation de différentes fréquences d'exécution (ex : Typiquement, le module visuel s'exécutera à 60Hz lorsque le moteur de dynamique de véhicule sera à 1000Hz). Une telle gestion centralisée des données simplifiera l'implémentation d'un outil de supervision des échanges de messages et de l'accès aux données du serveur, ce qui est utile pour déboguer.

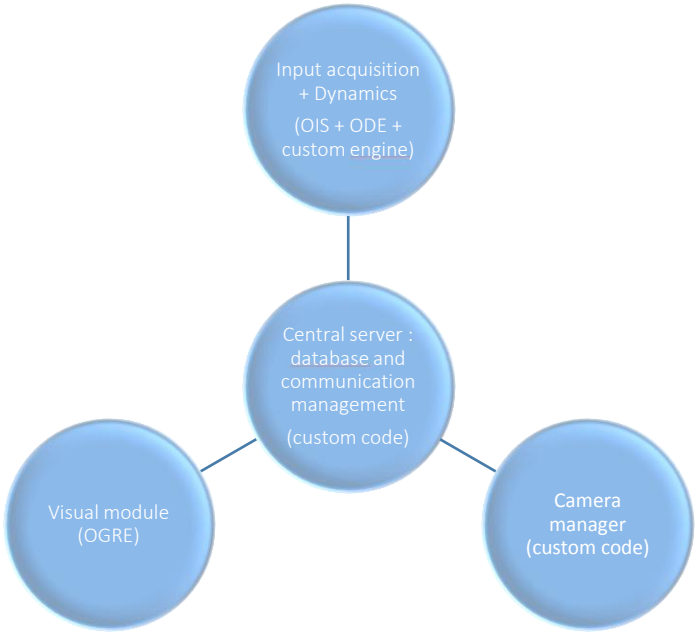

Figure 1 : Architecture d'OpenSD2S

4. MODULES

4.1. RENDU VISUEL

Le module visuel est utilisé pour afficher une vue de l'environnement virtuel du point de vue du conducteur. Avant le rendu de chaque image, le module visuel envoie un message au serveur central pour lui demander la dernière position et orientation du point de vue. Le visuel est généré avec OGRE (Streeting, S et al., 2000), un moteur de rendu 3D orienté objet open source écrit en C++, personnalisable et extensible grâce aux nombreux plugins disponibles gratuitement, permettant par exemple le support d'OpenGL et DirectX, ou des liaisons avec de multiples moteurs physiques. Enfin, OGRE bénéficie d'une communauté active

Dans le cas le plus simple, le module s'exécute sur un unique ordinateur et diffuse des images de la route sur un écran situé devant le siège du conducteur. Le module graphique est également capable de gérer plusieurs configurations d'écrans plus complexes car il peut s'exécuter plusieurs fois sur chaque ordinateur. Cela permet un clustering basique mais tout de même puissant.

Pour améliorer l'immersion et les sensations de conduite, certains mécanismes ont été implémentés au niveau du visuel. Les configurations ou des écrans entourent l'utilisateur sont supportées. Une pyramide de vision asymétrique est calculée pour chaque écran pour s'assurer d'avoir une vue simulée correcte par rapport à la taille de la tête. Des fichiers de configurations sont utilisés pour décrire la configuration spatiale des écrans ainsi que la position de la tête, ce qui est assez flexible pour permettre de s'adapter à tous types de simulateurs.

La plupart du temps, une approximation de la position de la tête obtenue à partir de la position du siège suffit pour retranscrire de bonnes sensations de conduite. Cependant, des données plus précises peuvent être requises pour rendre possible la parallaxe de mouvement, améliorant ainsi la perception de profondeur. Le module a été pensé pour des évolutions futures, ouvrant la voie à un futur module de tracking de tête. L'implémentation d'un tel module serait aisée car il existe déjà des outils open source pour gérer ces aspects, tel que VRPN. Il suffirait juste de diffuser des messages MPI contenant la position et l'orientation de la tête.

La stéréoscopie active est aussi supportée pour les configurations disposant de cartes graphiques NVIDIA Quadro grâce à la technologie Quad Buffer pourvus que les dispositifs d'affichages soient compatibles ce qui signifient qu'ils doivent être capables d'atteindre les 120Hz de fréquence d'affichage. Aussi la stéréoscopie passe est nativement gérée par OGRE.

L'un des principaux problèmes de la génération de graphismes 3D sur une grappe d'ordinateurs est la synchronisation de tous les nœuds. Sans elle, les images peuvent sembler discontinues à la jonction des écrans ou entre les deux yeux lors de l'utilisation de la 3D stéréoscopique. Plusieurs couches de synchronisations sont implémentées dans le module visuel. Le Framelock est la synchronisation de la scène 3D sur chaque nœud de la grappe. Le Swaplock permet de s'assurer que chaque image est affichée sur chaque écran en même temps. Tous ces mécanismes de

synchronisation sont implémentés logiciellement en utilisant les mécanismes de synchronisation de MPI.

Figure 2 : Vue virtuelle de Guyancourt, rendue dans OpenSD2S

Le choix d'un moteur de rendu 3D particulier implique aussi des contraintes sur les types de fichiers qu'il est possible de lire. OGRE ne gère que son propre format binaire, mais de nombreux plugins de conversion et exporteurs existent pour la plupart des logiciels d'édition 3D, incluant des logiciels open source tel que Blender. Dans le cadre de nos tests, une base de données visuelle modélisée avec Autodesk® Maya® et utilisée dans des scénarios typiques chez Renault a été exportée en utilisant un de ces plugins.

4.2. DYNAMIQUE DE VEHICULE

Le modèle de dynamique de véhicule joue un rôle capital dans un logiciel de simulation de conduite, car il récupère les commandes utilisateur en entrée et calcule la position du véhicule, son orientation et d'autres informations telles que la vitesse du moteur. Actuellement, il n'existe aucun modèle de dynamique de véhicule open source disponible sous forme de librairie. La plupart des simulations automobiles est tournée vers les jeux-vidéo et se concentre sur l'expérience ludique et la jouabilité plutôt que l'exactitude du comportement du véhicule. C'est pourquoi un module de dynamique de véhicule a été créé de zéro en utilisant ODE (Open Dynamics Engine) (Smith R., 2001) et MATLAB Simulink, un puissant outil de modélisation et de simulation pour concevoir des systèmes dynamiques complexes et qui peut générer automatiquement du code C pour des implémentations temps réel.

Le modèle du véhicule possède 10 degrés de libertés, qui incluent les mouvements latéraux et longitudinaux, lacet et le roulis du véhicule, la dynamique des roues ainsi que du groupe motopropulseur. En ce qui concerne le groupe motopropulseur, le moteur est représenté par le système de caractéristiques moteur universel (Zhang Z.P. and Zhang J.Z., 2012), alors qu'un modèle de friction venant de la littérature (Bățăuș, M. et al., 2011) a été utilisé pour modéliser l'embrayage connecté à une boite de vitesse manuelle. De plus, le système de freinage

comprenant l'ABS a été intégré comme un système aérodynamique à une dimension (axe longitudinal). Dans cette version, la dynamique verticale n'est pas considérée, ainsi la suspension a été simplifiée en ajoutant une barre anti-dévers à l'avant et à l'arrière. La dernière version de la formule magique bien connue de Pacejka (Pacejka, H.B., 2002) a été utilisée pour modéliser les pneus. En particulier, l'état transitoire est pris en compte. L'intégration des distances de relaxation dans les directions longitudinale et latérale permettent la stabilisation du véhicule aux faibles vitesses, et la conduite en marche arrière (Bernard, J.E. and Clover, C.L., 1995). La simulation globale s'exécute à 1000Hz indépendamment de la fréquence d'affichage qui est typiquement de 60Hz grâce à l'architecture modulaire. Une version temps réel du modèle de dynamique de véhicule fonctionnant sur un Raspberry Pi a été développée pour améliorer les performances. Le Raspberry Pi est un nano-ordinateur mono-carte de la taille d'une carte de crédit qui est peu couteux (25$) et populaire.

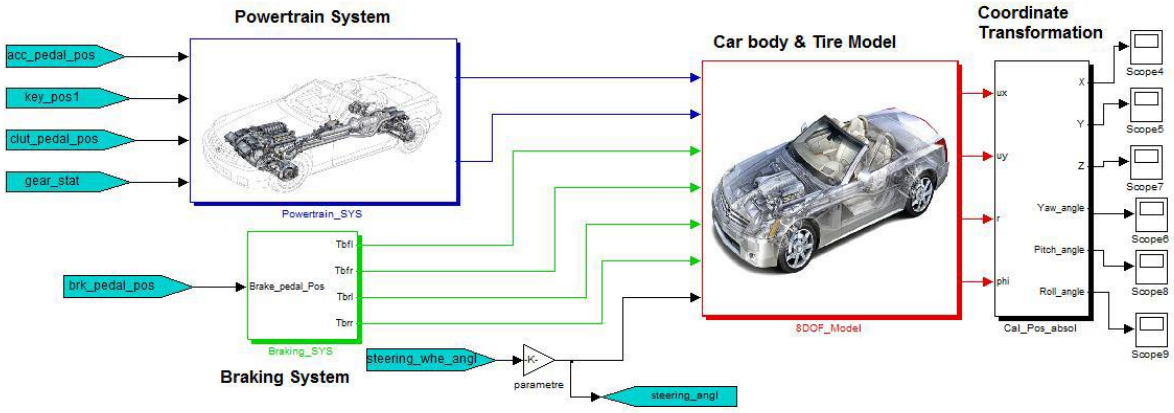

Figure 3 : Modèle de dynamique de véhicule dans Simulink