

Sommaire

Dédicace	I
Remerciements	II
Résumé	III
Abstract	III
Liste des figures.....	V
Liste des tableaux	VI
Liste des abréviations	VII
Introduction générale.....	1
Chapitre I : Contexte générale du projet et cahier de charge.....	3
I. L'organisme d'accueil:.....	3
1.1. Présentation du Groupe Altran :	3
1.2. Présentation Altran Maroc :.....	5
II. Contexte du projet et cahier de charge	8
1. Contexte du projet	8
2. Cahier de charge	11
Conclusion :	12
Chapitre II : Etude de la fonction OAM et plateforme du travail.....	13
Introduction :	13
I. La vidange de l'huile moteur	13
1. Principe de la vidange moteur	13
2. Utilité de la vidange moteur	13
3. Fréquence de la vidange	14
II. La fonction de contrôle moteur OAM	17
1. Maintenance Auto-adaptative.....	17
2. Fonctions principales OAM :	17
3. Le cycle de vie de la fonction OAM : cycle en V	19
III. Evaluation des performances du système de commande de la fonction OAM	25
1. Outil de tests : plan d'intégration-validation existant.....	27
2. Script de lancement :	28
IV. Plateforme de projet :	30
1. MATLAB	30

2. Simulink	32
3. L'Interface Graphique sous Matlab.....	33
Conclusion :	36
Chapitre 3 : Réalisation du projet et résultats obtenus	12
Introduction :	37
I. Analyse fonctionnelle :.....	37
1. Identification des besoins :	37
2. Définition des besoins :	39
II. Présentation de l'interface homme machine IHM_OAM.....	41
1. Introduction:	41
2. Organigramme de fonctionnement de l'outil.....	43
3. Lancer une simulation sur IHM_OAM :	44
4. Récupérer et visualiser les sorties de la fonction OAM :	47
5. Modifier les paramètres d'un roulage:	49
6. Ajouter des paramètres d'un roulage :.....	50
7. Créer un nouveau scénario :	50
III. Gains obtenus avec l'outil :	51
Conclusion générale	54
Bibliographie	55
Webographie	55

Liste des figures

Figure 1: Altran, un leader mondial	3
Figure 2 : Organigramme général d'Altran Maroc	6
Figure 3: Département Spécificité.....	7
Figure 4 : Les phases de réalisation du projet	9
Figure 5 : planning du stage PFE	10
Figure 6 : GANTT du projet.....	11
Figure 7 : circuit de lubrification.....	13
Figure 8 : Exemple de programme d'entretien	15
Figure 9 : Exemple sur configuration des cinq Intervalles de vidange.....	18
Figure 10 : Cycle de vie d'une fonction contrôle moteur.....	19
Figure 11 : Schéma du processus de test.....	26
Figure 12 : Environnement du modèle.....	27

Figure 13 : Première planche du modèle	28
Figure 14 : commande window	31
Figure 15 : Espace de travail.....	31
Figure 16 : Fenêtre des paramètres de solver	32
Figure 17 : Fenêtre des paramètres du Workspace.....	33
Figure 18 : hiérarchie de classement des objets graphique	34
Figure 19 : Exemple d'une interface graphique Matlab	35
Figure 20: bête à corne.....	37
Figure 21 : Diagramme pieuvre de l'IHM	38
Figure 22 : Présentation de l'interface IHM_OAM	42
Figure 23 : Organigramme fonctionnement de l'interface.....	43
Figure 24 : Affichage du roulage choisi	44
Figure 25 : Choix d'un roulage.....	44
Figure 26 : Fenêtre de calibration par défaut.....	45
Figure 27 : fenêtre de calibration.....	45
Figure 28 : Liste des moteurs	46
Figure 29 : Modification du couple moteur calibration	46
Figure 30 : Choix du fichier à ouvrir	47
Figure 31 : Liste des flux de sortie	48
Figure 32 : Liste des sorties à visualiser	48
Figure 33 : exemple pour Flux Masse_suies_ext	48
Figure 34 : Changement d'un scénario.....	49
Figure 35 : Liste des types des données	50
Figure 36 : Les calibrations du nouveau scénario	51

Liste des tableaux

Tableau 1 : Fiche technique de la société ALTRAN.....	4
Tableau 2: Tableau du plan de maintenance systématique.....	16
Tableau 3 : Gains obtenus par IHM_OAM	52

Liste des abréviations

A:

AIT : Automobile, Infrastructure et Transport

ASD : Aéronautique, Spatial et Défense

B:

BPO: Business Process Outsourcing

C:

CIS: Consulting in Information System

E:

EILoS: Energy, Industry & Life Sciences

F:

FSTF: Faculté des Sciences et Technique

G:

GMP: Groupe Moteur Propulseur

H:

HW: Hardware

I:

IHM: Interface Homme Machine

ITO: Information Technology Outsourcing

INCA: INtegrated Calibration and Acquisition

L:

LCC : Lois de Contrôle Commande

M:

MATLAB : MATrix LABoratory

O:

OAM : Oil Adaptative Maintenance

P:

PSA: Peugeot Citroën

S:

Spec : Spécification

SW: Software

SA: Société Anonyme

T:

TEM: Telecom & Media

TI: Technologies & Industry

V:

VPN: Virtual Private Network

Introduction générale

Le marché international automobile a connu, depuis longtemps, la prédominance de quelques industriels classiques qui, aujourd'hui, redoutent de plus en plus l'arrivée de la concurrence des pays asiatiques, notamment la Chine et l'Inde qui promettent de "casser les prix".

Face à ce contexte économique caractérisé par une concurrence accrue et une demande de plus en plus exigeante pour la livraison de produits diversifiés de bonne qualité et dans le plus court des délais, les entreprises doivent remettre en cause leurs organisations en vue de répondre à ces exigences. Dans ce contexte industriel, chaque entreprise souhaite réduire ses encours, optimiser la charge de ses ressources tout en augmentant la fluidité des flux, et clarifier ses ateliers pour véhiculer une image attractive. Pour ce faire, elle opte pour une politique d'amélioration continue de ses ressources de production.

De même, dans le cadre de cette stratégie, Altran se prépare à améliorer son processus de production et d'assurer la bonne qualité des livrables vis-à-vis les exigences de ses clients. Ainsi, dans un milieu industriel de plus en plus compétitif, ALTRAN Maroc se trouve dans l'obligation de satisfaire les impératifs CQD (coût, qualité, et délais). Pour pouvoir préserver cet équilibre, elle cherche à éliminer les sources de gaspillage et mettre en œuvre une organisation des opérations plus légère, agile et efficace, partant du principe que tout problème est une opportunité d'amélioration.

Dans cette optique, le constructeur automobile PSA a lancé un projet de développement d'un système de maintenance auto-adaptatif de l'huile qui permet d'alerter le client de la détérioration des propriétés lubrifiantes de l'huile. Ce système intègre une fonction de contrôle moteur dédiée à la surveillance de la dégradation d'huile en fonction de la quantité du carbone et du gazole dilués dans l'huile. Et là, intervient ALTRAN Maroc, qui s'occupera au cours de ce projet de l'optimisation du plan d'intégration et de validation de cette fonction.

Dans le premier chapitre de ce rapport je vais présenter l'organisme d'accueil, le cahier des charges et la problématique du projet en indiquant la méthodologie du travail et le

planning adopté.

Le deuxième chapitre s'intéressera de la fonction de contrôle moteur OAM (maintenance auto-adaptative de l'huile). Je commencerai d'abord par quelques généralités sur la vidange, puis le principe de la fonction en question avec son cycle de vie, ensuite je présenterai le plan d'intégration-validation existant, ainsi la plateforme utilisée pour la réalisation du projet.

Le troisième chapitre sera consacré au développement de la solution adoptée qui traitera en l'occurrence l'étude technique de la solution puis l'étude quantitatif.

Chapitre I : Contexte générale du projet et cahier de charge

Dans ce premier chapitre nous présentons l'organisme d'accueil, le client du projet la problématique, le cahier des charges général, nos objectifs ainsi que le planning adopté pour la réalisation du projet.

I. L'organisme d'accueil:

1.1. Présentation du Groupe Altran :

Leader mondial du conseil en innovation et ingénierie avancée, Altran accompagne les entreprises dans leurs processus de création et développement de nouveaux produits et services. Les Innovation Makers du Groupe interviennent depuis 30 ans auprès des plus grands acteurs des secteurs. Les offres du groupe, déclinées depuis les phases du plan stratégique en matière de technologies nouvelles jusqu'aux phases d'industrialisation, assurent la capitalisation du savoir au sein de cinq domaines principaux : Intelligent Systems, Innovative Product Development, Life-cycle Experience, Ingénierie Mécanique, et Systèmes d'Information.

■ Domaines d'activités :

- Aérospatial,
- Automobile,
- Energie,
- Ferroviaire,
- Finance,
- Santé,
- Télécommunications,
- Autre.

1.1.1. Les différents sites :

Présent sur de nombreux sites en Europe, en Amérique, et en Asie, le groupe Altran se situe à proximité immédiate de ses clients, leaders dans des secteurs d'activités comme ceux de l'automobile, du ferroviaire, de l'aéronautique, de l'énergie, du spatial ou des télécoms.

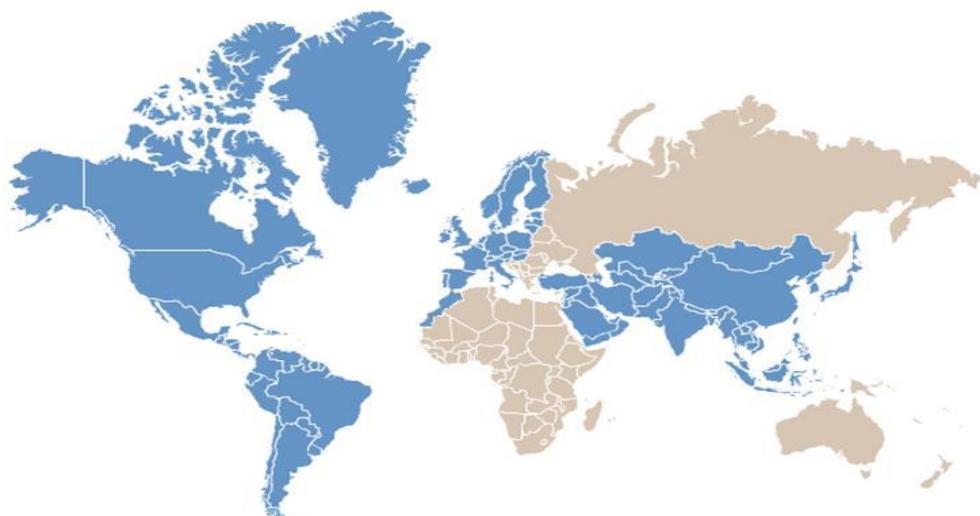


Figure 1: Altran, un leader mondial

1.1.2. Organisation :

Altran a été organisée en quelque 200 filiales, toutes autonomes dans leur gestion opérationnelle et leur stratégie commerciale. Le groupe Altran compte maintenant environ 80 filiales opérationnelles.

Le 29 décembre 2006 toutes les filiales d'Île-de-France ont fusionné au sein d'Altran Technologies SA qui a été décomposé en deux grands pôles :

- Altran TI : Technologies & Industry (Technologies & Industrie) qui est constitué des entités (marques commerciales) :
 - ❖ Altran AIT : Automotive, Infrastructures, Transportation (Automobile, Infrastructures, Transports).
 - ❖ Altran EILiS : Energy, Industry & Life Sciences (Energie, Industrie et Sciences de la Vie).
 - ❖ Altran ASD : AeroSpace & Defence (Aérospatiale et défense).
- Altran CIS/TEM qui comprend :
 - ❖ Altran TEM: Telecom & Media (Télécom & Média).
 - ❖ Altran CIS: Consulting in Information System (Conseil en système d'information) devenu Altran Finance & Government (Finances et gouvernement).

Le Groupe est depuis 2010 organisé autour de ces cinq Industries dans l'ensemble des pays.

1.1.3. Fiche technique de la société :

Le tableau suivant présente la fiche technique d'ALTRAN :

Siège social	54/56, avenue Hoche, 75008 Paris,
Téléphone	+33 (0)1 46 41 70 00
Activité	Conseil en innovation et ingénierie avancée.
Capital	87 519 354 €
Chiffre d'affaire	1.756 M€ (2014)
Forme juridique	Société Anonyme
Date de création	1982
Effectif	21 000

Tableau 1 : Fiche technique de la société ALTRAN

1.1.4. Partenariat avec PSA :

Le partenariat avec PSA renforce les liens qui unissent Altran et PSA Peugeot Citroën. Partenaire industriel historique, Altran est labellisé « fournisseur stratégique » dans le domaine de la conception des produits et process automobiles dans le cadre du Plan de Performance 2012 de PSA Peugeot Citroën. Altran est donc désormais étroitement associé aux projets de Recherche et Développement sur lesquels se jouent les performances de la marque.

1.2. Présentation Altran Maroc :

A travers son implantation au Maroc, Altran a disposé une plateforme nearshore afin d'accompagner le développement international du groupe dans les secteurs de l'automobile, de l'aéronautique et du transport. Il s'agit en effet d'accompagnement des clients Altran dans leur stratégie d'innovation, d'optimisation de coût et d'internationalisation.

L'entité marocaine a également pour ambition d'être un acteur de proximité au service des grands comptes clients d'Altran installés sur le territoire national. Dans le cadre de la stratégie « émergence » lancé par le gouvernement marocain, de nombreuses sociétés étrangères, et à fort développement, s'y sont installées. Altran Maroc s'intéresse notamment à celles évoluant dans les secteurs de l'automobile, de l'aéronautique et de l'énergie renouvelable.

Enfin, Altran Maroc s'appuie sur la stratégie offshoring mise en place par le gouvernement marocain offrant des avantages optimisant fortement la composante compétence / coût (spécialisation des parcours supérieurs dans les métiers de l'offshoring, plans de formation, attractivité des salaires, fiscalité,).

Altran Maroc est installé au parc Casanearshore à Casablanca. Ce parc dédié aux activités de nearshore (BPO, ITO...) avec plus de 300 000 m² de bureaux, est le plus grand parc de ce type au Maroc et en Afrique du Nord. Il se caractérise également par une proximité de l'aéroport international de Casablanca et des principaux accès autoroutiers.

1.2.1. L'organigramme général :

L'organigramme général comporte une direction générale qui supervise les différentes directions et divisions de l'entreprise :

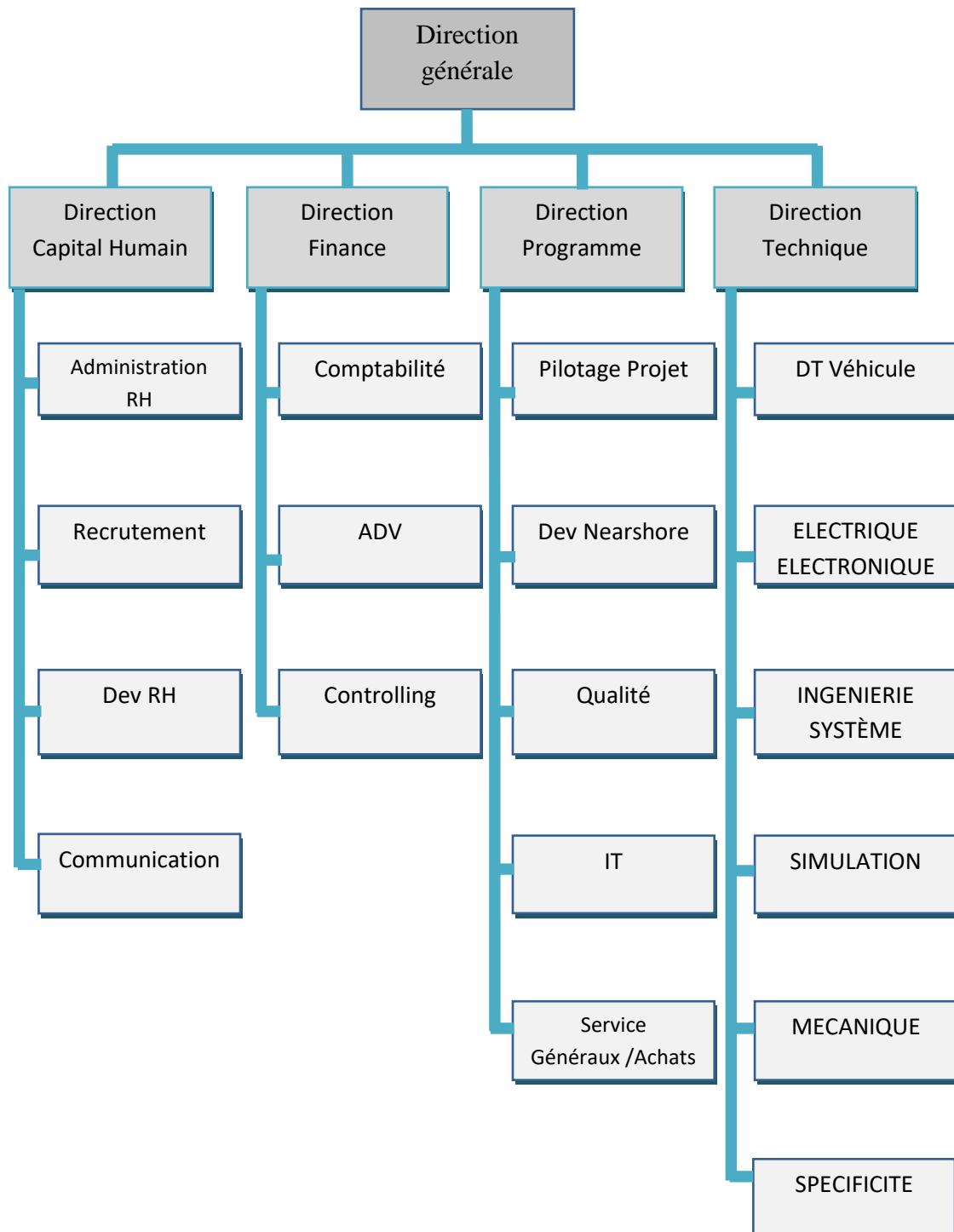


Figure 2 : Organigramme général d'Altran Maroc

1.2.2. Département spécificité :

La figure suivante présente l'organisation du département spécificité :

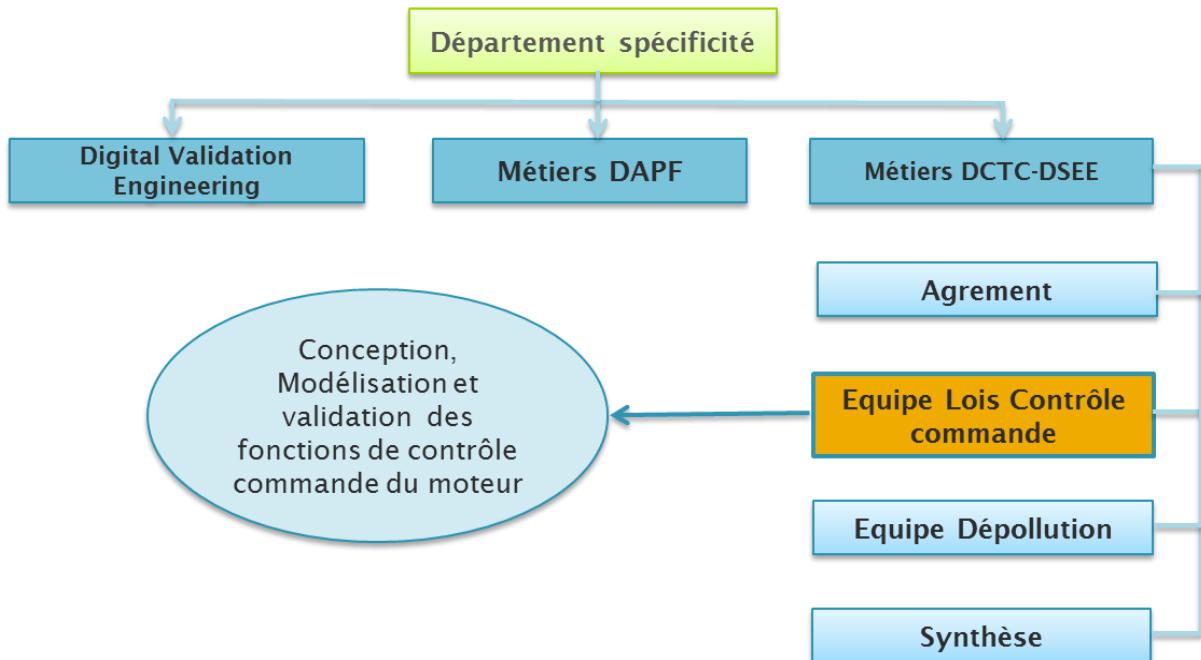


Figure 3: Département Spécificité

DAPF : Direction Architecture Physique et Fonctionnelle

DCTC : Direction Chaine de Traction et Chassis

DSEE : Direction systèmes électricité et électroniques

1.2.3. L'équipe LCC :

L'équipe à laquelle je suis affecté est l'équipe LCC d'un GMP Essence/Diesel qui s'occupe de :

- 1- Concevoir, modéliser et industrialiser des fonctions de pilotage dans le cadre des applications automobiles sur calculateur embarqué du Groupe PSA.
- 2- Développer un cœur de métier en contrôle commande.
- 3- Validation et préparation des moyens HW/SW pour la validation.
- 4- Participer au cycle de développement des lois de contrôle commande du GMP.

Missions de l'équipe en détails consistent notamment à :

- Lecture et compréhension des spécifications techniques **T3** ou des exigences clients.
- Suivi des spécifications techniques durant le processus de modélisation (**tracabilité**)

- Construction des modèles et des flux (traduction des exigences en modèle, adaptation des modèles Simulink)
- Génération des scénarios de test.
- Vérification des nouvelles stratégies par simulation, et validation des exigences par des outils de prototypage rapide (en temps réel)

Les outils de travail de l'équipe LCC: Matlab/Simulink/Script/Stateflow, INCA

II. Contexte du projet et cahier de charge

1. Contexte du projet

La stratégie pour traiter le sujet de stage s'appuie sur un recueil d'informations, notamment des relevés de l'état actuel. Dans ce chapitre, nous allons présenter d'une part le cahier de charges du projet tout en détaillant les besoins. D'autre part, nous allons décrire la méthodologie utilisée afin d'organiser le travail ainsi que la planification adoptée.

1.1. Problématique :

La méthode de validation de la fonction OAM (maintenance adaptative de l'huile) consiste à tester le modèle avec des Outils définis. En effet, ces outils qui ne sont enfin de compte que des scripts (des programmes sur des fichiers (.m) sous Matlab), qui vont charger des acquisitions et calibrations permettant de lancer la simulation.

Etant donné que la reconfiguration de l'ensemble de ces scripts constitue une charge lourde à l'utilisateur, du point de vue temps alloué à cet effet, précision que nécessite chaque cas de test et complexité des scripts en eux-mêmes.

Le but principal de ce projet est de rendre d'une part le processus de tests plus rapides et facile à configurer. Et d'autre part aider l'équipe dans l'optimisation des scénarii de tests de la fonction OAM (maintenance adaptative de l'huile).

1.2. Acteur du projet

- **Maitre d'œuvre :** **Mohammed EL HILALI** élèves ingénieurs en ingénierie mécatronique à la Faculté des Sciences et Techniques de Fès.
Encadrant pédagogique : **Mr Driss Tahri** Enseignant-Chercheur et professeur à la FST de Fès.
Encadrant de stage : **Younes AIT EL MAATI** Consultant Ingénieur au sein de

l'équipe LCC.

- **Maitre d'ouvrage :** Le maître d'ouvrage est l'entreprise Altran, qui est un centre d'innovation et de recherche, implanté au Maroc en 2013 à Casablanca, présenté par **Mr. Ahmed Benslimane** Team Leader au département Spécificité, **Mr. El Mahdi Abghour** Liaison Team et présentant du client PSA au sein d'Altran.

1.3. Les phases de réalisation du projet

La procédure suivie pendant l'élaboration du projet comporte trois phases principales à savoir

- La phase de documentation,
- La phase d'étude,
- La phase de réalisation.

On peut présenter le travail à faire de chaque phase dans la figure suivante :

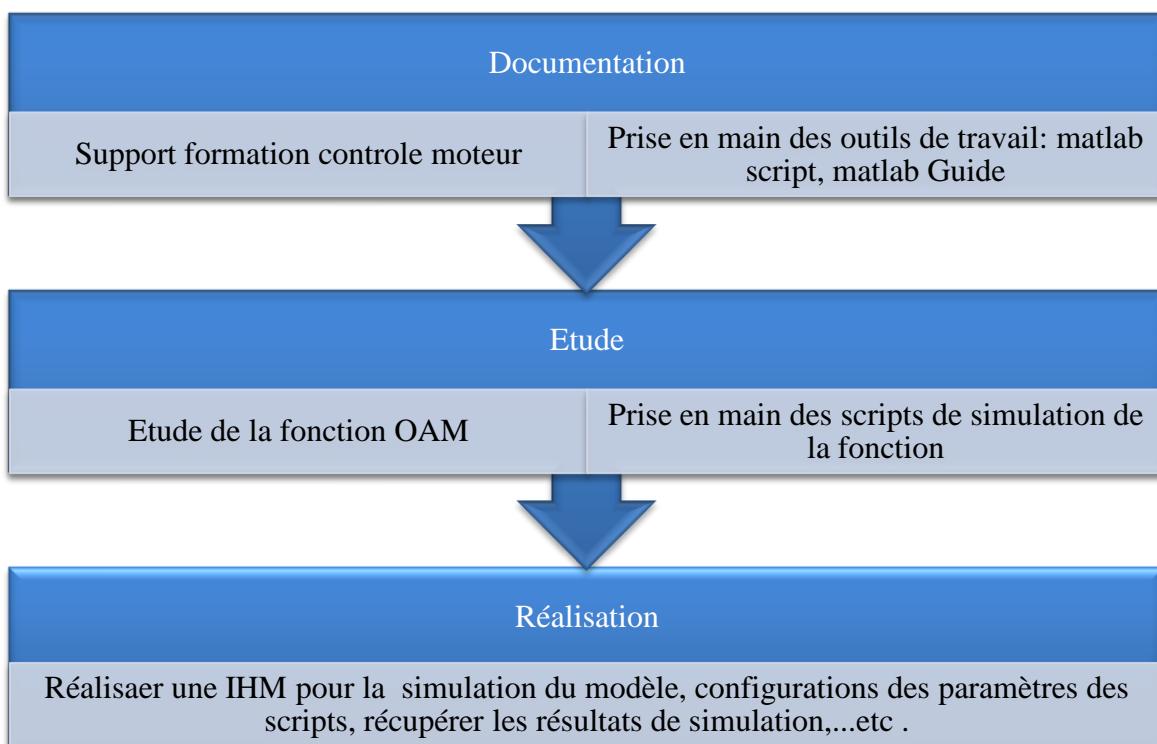


Figure 4 : Les phases de réalisation du projet

1.4. Méthodologie et planning pour résoudre le problème

1.4.1. Identification des moyens

Avant de détailler la méthodologie suivie pour la réalisation du projet, il serait judicieux d'identifier les moyens qui nous permettront de réaliser le projet.

1.4.1.1. Identification des moyens humains

- Trois encadrants (Deux encadrants d'Altran et un encadrant à l'établissement).
- Une équipe des consultants Altran.

1.4.1.2. Identification des techniques utilisées

- Scripting Matlab, les interfaces graphiques
- Classification des activités selon la typologie.

1.4.1.3. Identification des moyens matériels et logiciels

- Outil de modélisation, Matlab Simulink.
- Outil de programmation, Matlab Script.
- Moyen de création IHM, Matlab Guide.

1.4.2. Planification du projet

Dès la première semaine de stage de fin d'étude, nous avons décrit un planning d'avancement de projet, réalisé sous forme d'un diagramme. Dans cette planification, nous avons déterminé et ordonné les tâches du projet, avec estimation des charges (Délai et ressources) et on a mobilisé les profils nécessaires à leur réalisation. Ce planning est illustré dans la figure suivante :

		Tas Mo	Task Name	Duratio	Start	Finish
1	✓	➡	Prise en main de l'interface GUI+ réalisation d'exemples simples	15 days	Mon 06/03/17	Fri 24/03/17
2	✓	➡	Entrainement sur le Scripting	10 days	Mon 20/03/17	Fri 31/03/17
3	✓	➡	Prise en main de la fonction OAM	5 days	Mon 03/04/17	Fri 07/04/17
4	✓	➡	Prise en main du script OAM	10 days	Mon 10/04/17	Fri 21/04/17
5	✓	➡	Liaison script OAM / Interface	15 days	Mon 24/04/17	Fri 12/05/17
6		➡	Développement de l'interface	10 days	Mon 15/05/17	Fri 26/05/17
7	✓	➡	Rapport de stage	60 days	Mon 20/03/17	Fri 09/06/17
8	✓	➡	Présentation PPT	5 days	Mon 12/06/17	Fri 16/06/17

Figure 5 : planning du stage PFE

Le GANTT du projet est présenté sur la figure suivante:

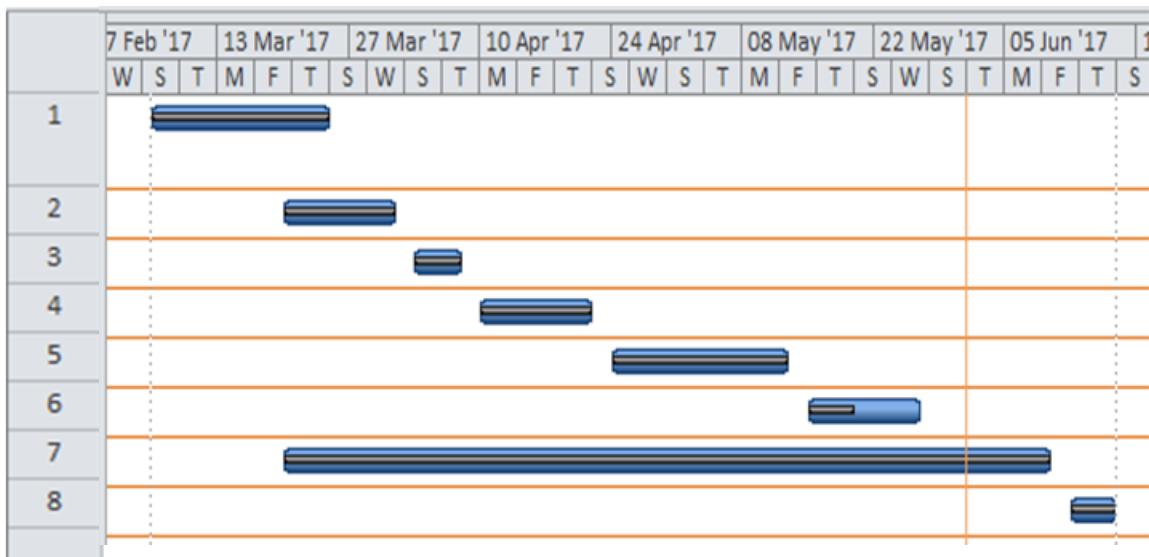


Figure 6 : GANTT du projet

2. Cahier de charge

2.1. Contexte pédagogique

Ce travail s'inscrit dans le cadre du stage de projet de fin d'études, durant lequel je suis censé de participer avec l'équipe LCC pour réaliser un projet industriel. Au cours de ce stage, je dois mettre en place mes acquis et mes compétences pour résoudre l'ensemble des difficultés rencontrées dans la réalisation du travail qui m'a été confié.

2.2. Besoin exprimé :

Le besoins exprimé par le maître d'ouvrage est de réaliser une IHM (interface homme machine) sous Matlab pour la simulation du modèle de la fonction auto-adaptative de l'huile moteur. Ce besoin consiste à optimiser les scénarios de test de cette fonction de contrôle moteur en paramétrant des scripts à partir d'une interface graphique pour rendre le lancement des simulations plus rapide, facile et interactive avec l'utilisateur.

2.3. Contraintes à respecter

La gestion de ce projet doit tenir en compte un certain nombre de contraintes détaillées ci-dessous, elles sont de trois types :

2.3.1. Contraintes pédagogiques :

Apprendre à être autonome dans la réalisation d'un projet. Acquérir de nouvelles connaissances techniques.

2.3.2. Contraintes temporelles :

Rendre le livrable du travail réalisé à des dates précises.

2.3.3. Contraintes de réalisation :

Réaliser des programmes permettant de lier l'interface graphique avec le modèle et avec les scripts de simulation.

Garantir la qualité de l'application afin d'éviter les bugs lors de son fonctionnement.

Conclusion :

Après avoir donné un aperçu sur l'organisme d'accueil et son domaine d'activité, ainsi que le travail demandé dans le cadre de ce projet de fin d'études, on abordera le deuxième chapitre qui explique en détail la fonction de contrôle moteur OAM qui présente l'objet de notre sujet.

Chapitre II : Etude de la fonction OAM et plateforme du travail

Dans ce deuxième chapitre nous présentons la fonction OAM, le cycle de validation de la fonction et la plateforme du travail

Introduction :

L'entretien du moteur de l'automobile est indispensable pour le bon fonctionnement, la fiabilité et la longévité de celle-ci. Il doit être régulier et complet, tel que l'indique le constructeur dans le carnet d'entretien.

Parmi les premières opérations à effectuer, on trouve la **vidange** (moteur), avant la vérification du niveau du liquide de refroidissement, le contrôle de la courroie de distribution et celui de l'échappement.

I. La vidange de l'huile moteur

Le terme « vidange » est utilisé dans le langage courant pour désigner la « vidange moteur ». Il existe pourtant d'autres vidanges : celles du filtre à air, du filtre à carburant et du filtre d'habitacle.

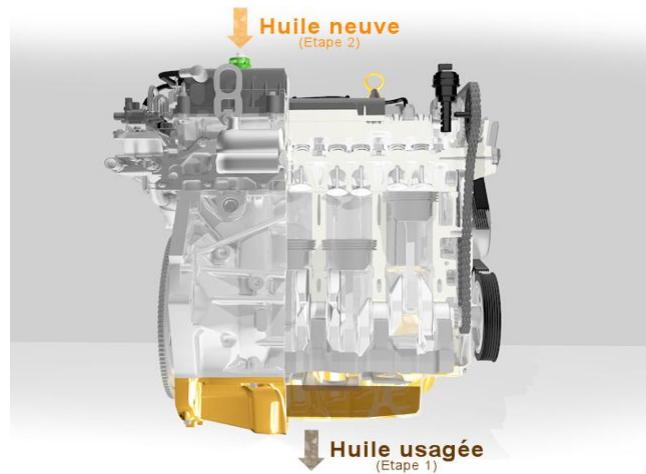


Figure 7 : circuit de lubrification

1. Principe de la vidange moteur

C'est une opération simple mais essentielle. Elle doit être effectuée souvent, et est très recommandée avant de longs trajets. Elle consiste :

- à vider l'huile usée du moteur pour la remplacer par une huile propre ;
- à remplacer impérativement le filtre qui, étant sale lui-même, souillerait évidemment la nouvelle huile.

2. Utilité de la vidange moteur

Cette huile est utile à la lubrification du moteur. Elle sert à réduire le frottement des pièces mobiles métalliques entre elles, qui pourraient se dégrader au fil du temps et abîmer le moteur.

Elle est également utile à :

- l'évacuation des particules de métal, les saletés et autres impuretés présentes dans le moteur ;
- la protection des organes internes contre la corrosion ;
- l'amélioration du refroidissement du moteur.

3. Fréquence de la vidange

La vidange moteur est facile à faire (par un professionnel ou soi-même), et le contrôle du niveau d'huile encore davantage. Ce contrôle est préconisé pour pratiquement tous les véhicules, sauf les véhicules électriques, qui n'utilisent pas de carburant.

a. Fréquences de changement d'huile

La fréquence préconisée par les différents constructeurs est de :

- tous les **10 000 à 15 000 km** pour les moteurs **essence** ;
- tous les **5 000 à 7 000 km** pour les moteurs **diesel**.

Cette fréquence varie évidemment beaucoup selon le type de votre véhicule, mais aussi selon votre type de conduite et l'environnement dans lequel vous évoluez.

- En effet, un véhicule roulant surtout sur autoroute sera moins sujet à ce genre d'opération qu'un autre parcourant uniquement la ville.
- De même, un diesel sera vidangé plus rapidement qu'un moteur essence (6 mois contre 1 an).

b. Fréquence des révisions : âge et kilométrage

En général, les révisions s'effectuent tous les 15 000, 20 000, 30 000 ... 100 000 km.

Si vous roulez peu et qu'il vous faudra 3 ans pour arriver à 20 000 km, l'**âge est prioritaire**, et vous devez donc effectuer votre révision tous les deux ans.

Un exemple d'un programme d'entretien est présenté sur la figure suivante :

	Entretien tous les ... (au premier des deux termes atteint)		Séquence de réalisation
	10 000 km ou 1 an	15 000 km ou 1 an	
Modèles Essence		*	1 – Révision de service (15 000 km ou 1 an) 2 – Révision d'entretien (30 000 km ou 2 ans) 3 – Révision de service (45 000 km ou 3 ans) 4 – Révision générale (60 000 km ou 4 ans) Ainsi de suite...
Modèles Diesel	*		1 – Révision de service (10 000 km ou 1 an) 2 – Révision d'entretien (20 000 km ou 2 ans) 3 – Révision de service (30 000 km ou 3 ans) 4 – Révision d'entretien (40 000 km ou 4 ans) 5 – Révision de service (50 000 km ou 5 ans) 6 – Révision générale (60 000 km ou 6 ans) Ainsi de suite...

Figure 8 : Exemple de programme d'entretien

Un calendrier d'entretien pour connaître la périodicité des opérations

À chaque révision, tous les éléments ne sont pas contrôlés. Le carnet d'entretien délivre donc, pour chaque opération de révision ou d'entretien, la périodicité qui s'y rapporte. Il sert ainsi d'**agenda** pour les opérations de contrôle, de réparation, de remplacement ou de vidange.

Voici, à titre indicatif, un tableau indiquant, pour chaque organe ou pièce ou lubrifiant, une opération et une périodicité en kilomètres (ou en années, au cas où le véhicule n'aurait pas encore atteint le kilométrage cité au bout de ce nombre d'années).

c. Plan d'entretien (Europe)

Le tableau suivant présente un plan des opérations de maintenance pour l'utilisation normale et sévère :

	Utilisation Normale [*]	Utilisation Sévère
OPERATIONS SYSTEMATIQUES (vidange huile, etc...)	30 000 km ou 1 an	20 000 km ou 1 an
filtre à air	60 000 km ou 4 ans	40 000 km ou 4 ans
filtre à gazole		
Plein AdBlue® (***)	30 000 km ou 4 ans	
KIT accessoires (courroie, galets)	120 000 km ou 6 ans (****)	120 000 km ou 6 ans (****)

	Utilisation Normale [*]	Utilisation Sévère
KIT distribution (courroie, galets, pompe à eau)	150 000 km ou 10 ans pour DS5 180 000 km ou 10 ans pour c4 Picasso (*****)	140 000 km ou 10 ans pour DS5 180 000 km ou 10 ans pour C4 Picasso (*****)
Contrôle du niveau d'additif filtre à particules	A chaque entretien à partir de 90 000 km	A chaque entretien à partir de 80 000 km
Contrôle du filtre à particules	A chaque entretien à partir de 180 000 km	A chaque entretien à partir de 180 000 km

Tableau 2: Tableau du plan de maintenance systématique

(*) Selon les conditions de roulage l'indicateur de maintenance au tableau de bord peut indiquer la nécessité d'anticiper l'entretien (opérations systématiques et remplacement des pièces associées) à une échéance de 20 000 km au lieu de 30 000 km depuis l'entretien précédent.

(**) Selon le type de roulage, un appoint d'AdBlue®¹ peut-être nécessaire entre deux pas d'entretien.

(***) Un code défaut remonté à Diagbox peut préconiser le changement anticipé de la courroie accessoire. Le changement est impératif en cas d'alerte défaut STT au tableau de bord accompagné d'un code défaut spécifique.

(****) L'intervalle de remplacement du kit distribution des applications véhicules futures DW10F sera documenté ultérieurement dans le plan d'entretien de chaque véhicule.

¹ L'AdBlue® : est une solution aqueuse composée à 32,5% d'urée et à 67,5% d'eau déminéralisée. Son utilisation permet de transformer 85% des polluants (oxydes d'azote, appelés NOx) en vapeur d'eau et azote inoffensif.

L'Adblue® est utilisé par les véhicules équipés de la technologie SCR (Selective Catalytic Reduction, en français Réduction Catalytique Sélective).

II. La fonction de contrôle moteur OAM

1. Maintenance Auto-adaptative

Le calculateur de contrôle moteur intègre une fonction de surveillance de la dégradation d'huile en fonction de la quantité de carbone (résidus de combustion) et de gazole dilués dans l'huile. Le calculateur détermine automatiquement l'intervalle kilométrique (30 000 km ou 20 000 km) approprié entre le prochain entretien et le précédent, en fonction des conditions de roulage du véhicule (par ex. conduite urbaine fréquente). Le calculateur tient également compte de l'intervalle de temps maxi de 1 an depuis la dernière révision.

A l'approche de l'échéance calculée, l'indicateur de maintenance au tableau de bord affiche le kilométrage restant et le temps restant avant la prochaine échéance. A noter que le voyant de service s'allume après dépassement de l'échéance d'entretien. La remise à zéro de l'indicateur de maintenance est obligatoire à l'issue de l'entretien pour le bon fonctionnement de la surveillance automatique.

2. Fonctions principales OAM :

La fonction OAM permet de calculer l'intervalle de vidange selon la sévérité du client en estimant :

- le pourcentage de carbone introduit dans le lubrifiant conforme.
- le pourcentage de gazole dilué dans le lubrifiant conforme.
- la consommation d'huile cohérente de la politique d'appoints d'huile client.

La fonction OAM sert de :

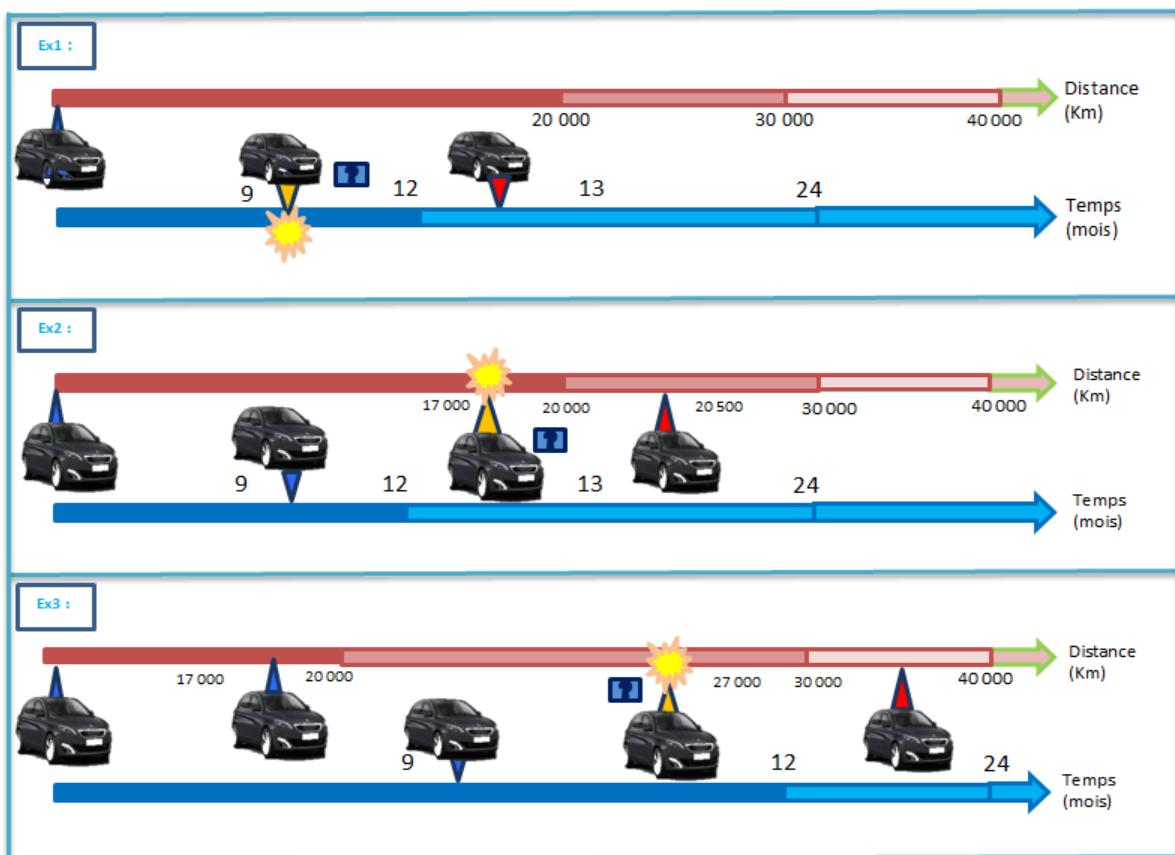
- **Alerter le client** de la détérioration des propriétés lubrifiantes de l'huile.
- **Adapter les fréquences de RG (régénération du filtre à particules) pour respecter le critère dilution.**
- **Adapter le pas de maintenance** d'huile en fonction du profil client.
- **Afficher au tableau de bord** l'échéance du prochain changement d'huile
 - Affichage kilométrique et temporel (IHM dite cible)
 - Affichage kilométrique (IHM dite dégradée)

- **5 Intervalles de Vidange possibles :**

- 3 Intervalles de Vidange kilométriques
- 2 Intervalles de Vidange temporels (1 an, 2 ans)

Les projections carbone/dilution estiment le taux de carbone/dilution pour **5 intervalles** et le superviseur de maintenance adaptive utilise ces projections pour détecter la sévérité du client.

La figure suivante présente un exemple sur configuration des cinq Intervalles de vidange :



Pour le premier exemple l'intervalle de vidange est temporel

Pour les exemples 2 et 3 l'intervalle de vidange est kilométrique

- La couleur jaune désigne la détection de la sévérité du conducteur.
- La couleur rouge désigne que l'échéance de l'intervalle de vidange est dépassé ; il faut changer l'huile.

3. Le cycle de vie de la fonction OAM : cycle en V

Le cycle de vie du développement, appelé cycle en V, regroupe la totalité des opérations de développement d'un logiciel depuis l'analyse du besoin jusqu'à la recette client. La conception, au sens de la norme ISO9001, représente pour nous l'ensemble des activités du cycle en V. La branche montante du V représente les contrôles et essais. Pour chaque phase du cycle de développement, différentes activités complémentaires sont nécessaires:

- Les activités de développement,
- Les activités de vérification effectuées par l'équipe projet,
- Les activités de préparation à la validation lors des phases du cycle descendant,
- Les activités d'exécution de la validation lors des phases du cycle montant,
- Les activités de fin de phase.

La figure qui suit indique le cycle en v de la fonction :

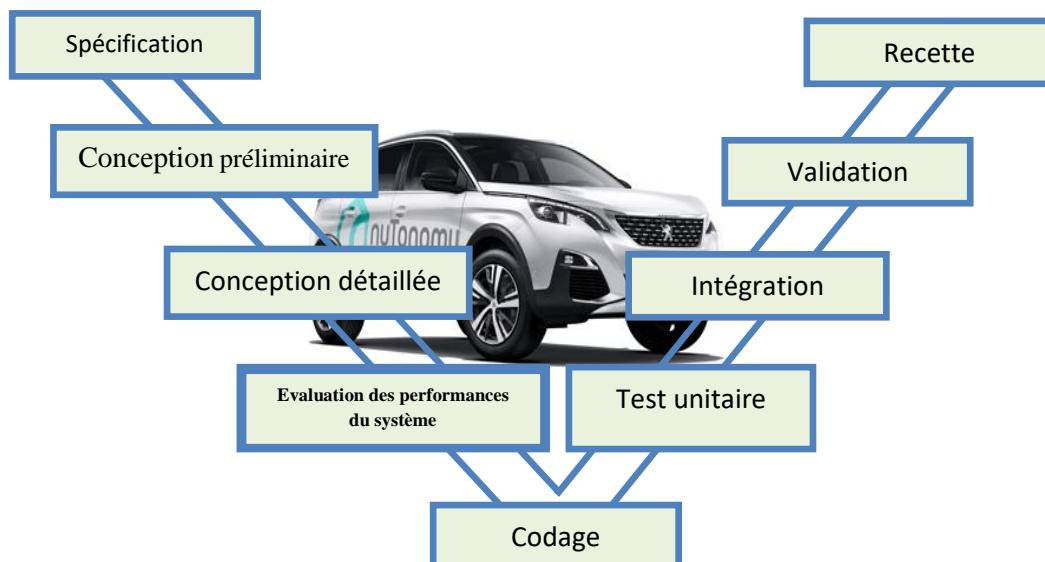


Figure 10 : Cycle de vie d'une fonction contrôle moteur

1. Phase de spécification

Avant d'entamer la conception, il est indispensable de savoir ce que l'on veut faire pour satisfaire les exigences du client, quelles sont les fonctionnalités à réaliser et les contraintes à respecter.

Une **spécification** est un ensemble explicite d'exigences à satisfaire par un matériau, un produit ou un service. Si un matériau, un produit ou un service ne parvient pas à satisfaire à une ou plusieurs des spécifications applicables, il peut être désigné comme étant hors spécification.

Les activités de développement des spécifications sont :

- Analyser l'expression des besoins du client et la proposition technique et commerciale
- Dégager les fonctionnalités à réaliser.
- Décrire les fonctionnalités et sous-fonctionnalités en quantifiant les exigences et caractéristiques de fonctionnement.
- Identifier les interfaces du logiciel.
- Faire l'inventaire des différents modes de fonctionnement du système.
- Prendre en compte les contraintes spécifiées dans l'expression de besoin du client.
- Préciser les exigences relatives à la phase de conception préliminaire.
- Rédiger le document de spécification.
- Rédiger le manuel utilisateur et le manuel d'exploitation.

Les activités de préparation à la validation sont les suivantes:

- Prévoir le plan de validation et le cahier de recette des tests " boîte noire " du système.
- Définir les objectifs, et l'environnement des tests de validation.
- Préciser les responsabilités des intervenants en validation et en recette.
- Définir les méthodes et stratégie des tests.
- Définir le déroulement et l'acceptation de la recette.
- Identifier les tests que doit satisfaire le logiciel pour obtenir sa validation, les mesures à effectuer et les critères d'acceptation associés.
- Prévoir, si besoin, un outillage de test.
- Prévoir la mise en œuvre et les résultats attendus des tests.

2. Phase de conception préliminaire

Cette phase regroupe l'ensemble des activités conduisant à l'élaboration de l'architecture du système. L'objet de la phase de conception préliminaire est donc de répondre à la question:

COMMENT?

Les activités de développement faites dans cette phase sont :

- Élaborer l'architecture statique et dynamique du logiciel.
- Définir la décomposition du logiciel en constituants/ modules tâches, objet...
- Décrire ces constituants, montrer la logique, l'imbrication, l'enchaînement de ces différents constituants ainsi que la circulation des données.
- Elaborer les solutions techniques pour gérer les différents modes de fonctionnement.
- Respecter les contraintes spécifiées dans le cahier des charges.
- Respecter les exigences relatives à la phase de conception.
- Rédiger le document de conception préliminaire.

Les activités de préparation à la validation dans cette phase sont les suivantes :

- Prévoir le plan d'intégration des tests "boîte blanche" du logiciel:
 1. Choisir la méthode d'intégration et la stratégie adoptée
 2. Prévoir l'outillage nécessaire aux tests
- Identifier les tests qui valident la conception préliminaire et les critères d'acceptation associés.

3. Phase de conception détaillée

C'est l'ensemble des activités consistant à détailler les résultats de la conception préliminaire tant sur le plan algorithmique que sur celui de la structure des données, jusqu'à un niveau suffisant pour permettre le codage.

Les activités de développement faites dans la phase de conception sont les suivantes:

- Achever la décomposition effectuée en conception préliminaire jusqu'aux entités les plus élémentaires (composants).
- Identifier les données d'interfaces de ces composants.
- Effectuer la description algorithmique des composants développés et la description des données de façon suffisante pour permettre le codage et les tests unitaires des composants.
- Définir les traitements d'erreur.

- Respecter les exigences relatives aux phases de conception.
- Rédiger le document de conception détaillée.

Les activités de préparation à la validation sont les suivantes :

- Prévoir le dossier des tests unitaires:
 1. Définir la stratégie de tests
 2. Définir les scénarios de tests s'appuyant sur la description fonctionnelle algorithmique et en essayant de couvrir la structure des composants
 3. Définir les résultats attendus des tests
 4. Prévoir l'outillage nécessaire aux tests.

4. Phase Evaluation des performances du système de commande :

Les activités de validation faites dans cette phase sont les suivantes :

- Déroulement de la validation de conception sur moyen physique (planning, résultats, gestion des anomalies, ...).
- Comparer les résultats obtenus aux résultats attendus.
- Consigner les résultats des tests.
- Finalisation du plan de validation sur moyen physique.
- Bilan validation de la fonction : Synthèse exhaustive des validations de conception
- bilan des faits techniques, planning du reste à faire
- Modèles d'optimisation.

Le développement de la fonction contrôle moteur OAM arrive à cette étape.

5. Phase de codage

Traduire les résultats de la conception détaillée en un programme.

Les activités de développement faites dans la phase de codage sont les suivantes :

- Respecter les standards de programmation
- Déclarer les données.
- Traduire la conception détaillée en langage source.
- Commenter le code.

- Effectuer les compilations et éditions de liens.
- Respecter les exigences relatives aux phases de conception.
- Finaliser les scénarios des tests unitaires.
- Préparer l'outillage nécessaire aux tests unitaires.

Les activités de vérification faites dans la phase de codage sont les suivantes :

Vérifier que toutes les exigences définies dans les documents d'entrée sont bien prises en compte dans les documents de sortie.

- Vérifier le respect des exigences qualité définies dans le plan d'assurance qualité spécifique.
- Vérifier que les métriques prévues dans le plan d'assurance qualité spécifique donnent les résultats attendus.

6. Phase de tests unitaires

Activité ayant pour but de vérifier, que chaque composant du logiciel pris isolément donne des résultats conformes au dossier de conception détaillée.

Les activités de validation faites dans la phase de test unitaire sont les suivantes :

- Finaliser les scénarios des tests unitaires
- Réaliser l'outillage de test
- Comparaison des résultats obtenus par rapport aux résultats attendus
- Evaluer le taux de couverture fonctionnelle et structurelle des tests en fonction des objectifs définis par le plan d'assurance qualité spécifique et compléter les tests si nécessaire.

7. Phase d'intégration

Activité consistant à assembler et à tester progressivement les composants du logiciel identifiés lors de la conception préliminaire et contrôlés lors des tests unitaires.

Les activités de validation faites dans la phase d'intégration sont les suivantes :

- A partir des fiches de tests, préparer le dossier des tests d'intégration.

- Compléter, si besoin, les scénarii de tests.
- Mettre en place la configuration matérielle et logicielle prévue dans le plan d'intégration logiciel ainsi que l'outillage de test.
- Effectuer les tests "boîte blanche" du logiciel selon la méthode prévue dans le plan d'intégration.
- Comparer les résultats obtenus aux résultats attendus.
- Evaluer le taux de couverture des tests en fonction des objectifs définis par le plan d'intégration et compléter les tests si nécessaire
- Consigner les résultats des tests.

8. Phase de validation

La validation est l'activité conduisant à s'assurer essentiellement au moyen de tests qu'un logiciel est conforme à ses spécifications et au manuel d'utilisation.

Les activités de validation faites dans la phase de validation sont les suivantes :

- A partir des fiches de tests, préparer le dossier des tests de validation.
- Compléter, si besoin, les scénarii de tests.
- Finaliser 'les manuels utilisateur' et d'exploitation.
- Mettre en place la configuration matérielle et logicielle prévue dans le plan de validation du logiciel ainsi que l'outillage de test.
- Effectuer les tests "boîte noire" du logiciel selon la méthode prévue dans le plan de validation:
 1. tests de fonctionnement nominal
 2. tests des cas aux limites et des cas d'erreurs
 3. tests des performances du logiciel.
- Comparer les résultats obtenus aux résultats attendus.
- Evaluer le taux de couverture des tests en fonction des objectifs définis par le plan de validation et des critères qualités du logiciel définis dans le plan d'assurance qualité spécifique et compléter les tests si nécessaire
- Consigner les résultats des tests.
- Rédiger la fiche de version du logiciel.

9. Phase de Recette

C'est l'activité qui consiste pour un fournisseur à présenter au client le logiciel étudié et/ou réalisé, et, pour ce dernier, à contrôler que celui-ci est conforme à toutes les exigences contractuelles.

Les activités de validation faites dans la phase de recette sont les suivantes :

- A partir des fiches de tests, préparer le dossier des tests de recette.
- Compléter, si besoin, les scénarii de tests.
- Finaliser ‘les manuels utilisateur’ et d’exploitation.
- Mettre en place la configuration matérielle et logicielle prévue dans le cahier de recette du logiciel ainsi que l’outillage de test.
- Effectuer les tests du logiciel avec client selon la méthode prévue dans le cahier de recette du logiciel:
 1. tests de fonctionnement nominal
 2. tests des cas aux limites et des cas d’erreurs
 3. tests des performances du logiciel.
- Comparer les résultats obtenus aux résultats attendus.
- Consigner les résultats des tests.
- Rédiger des fiches d’anomalies pour tout constat d’anomalie lors de la recette.

III. Evaluation des performances du système de commande de la fonction OAM

L’objectif du processus d’évaluation est de déterminer si le modèle de la fonction satisfait aux exigences, de démontrer qu’il est apte aux objectifs et de détecter des anomalies.

L’évaluation des performances du système ce fait par l’exécution des ’scénarii de test’² et la comparaison des résultats obtenus avec les résultats attendus pour que après si les résultats sont corrects on garde le test et les résultats dans une base de donnée, or si les résultats sont incorrects une analyse de ceci est obligatoire pour savoir la source d’erreur est-ce qu’elle est dans le test ou dans le modèle.

L’évaluation des performances du système de commande de la fonction OAM se fait suivant un processus de tests. L’objectif de tests est la couverture du domaine fonction, du

² Un scénario de test : une technique de conception de tests boîte noire dans laquelle les cas de tests sont conçus pour exécuter des scénarios de cas d’utilisation

domaine données entrées et sorties, du domaine contrôle et du domaine des enchainements, tel que :

- ✓ Toutes les fonctions sont exécutées au moins une fois.
- ✓ Toutes les régions de code (selon granularité) sont visitées au moins une fois.
- ✓ Toutes les entrées sont sollicitées au moins une fois, y compris les limites.
- ✓ Toutes les sorties attendues sont produites au moins une fois.
- ✓ Les comportements les plus fréquents sont sollicités au moins une fois. Ce qui implique une excellente connaissance du contexte d'emploi du système.
- ✓ Toutes les exceptions et les messages d'erreurs sont levés au moins une fois.

Un schéma récapitulatif du processus de test est présenté sur la figure suivante :

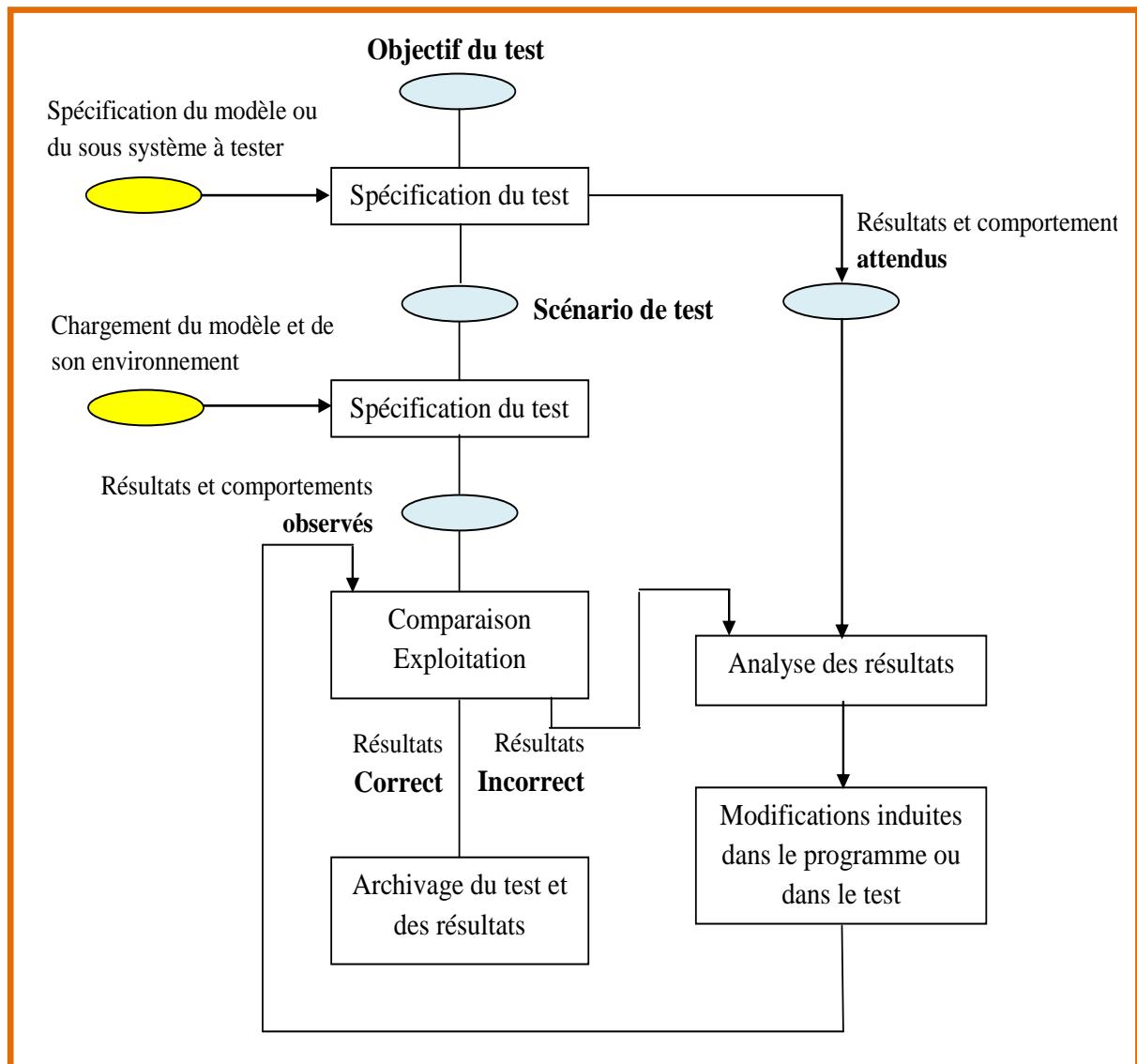


Figure 11 : Schéma du processus de test

1. Outil de tests : plan d'intégration-validation existant

Le plan d'intégration-validation existant de la fonction OAM consiste à tester le modèle avec des Outils définis. En effet, ces outils qui ne sont enfin de compte que des scripts³ vont charger des acquisitions et calibrations permettant de lancer la simulation, ils permettent aussi de sauvegarder les sorties du modèle.

- Représentation simplifié du Modèle Simulink

Le modèle Simulink de la fonction est constitué de plusieurs sous système. Chaque sous-système correspond à une fonction spécifique du système. Une fonction est réaliser pour faire un calcul ou une estimation d'une grandeur précise : par exemple calcul de la quantité de carbone, masse de suisse..., etc.

Les figures 1 et 2 indiquent l'environnement du modèle et sa première planche :

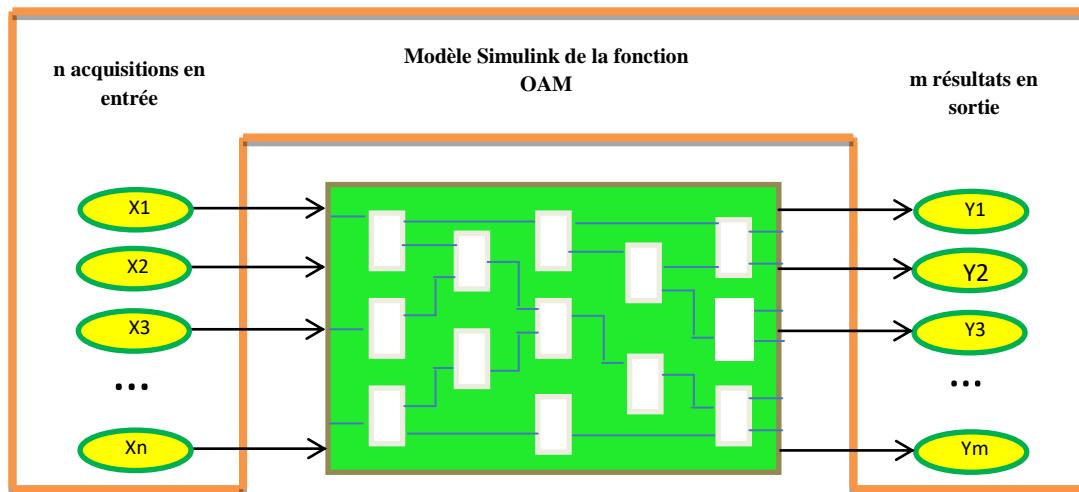


Figure 12 : Environnement du modèle

³ Un script est un ensemble d'instructions MATLAB qui joue le rôle de programme principal. Il est en effet beaucoup plus simple de modifier des instructions dans un fichier à l'aide d'un éditeur de texte que de retaper un ensemble d'instructions MATLAB dans la fenêtre de commande.

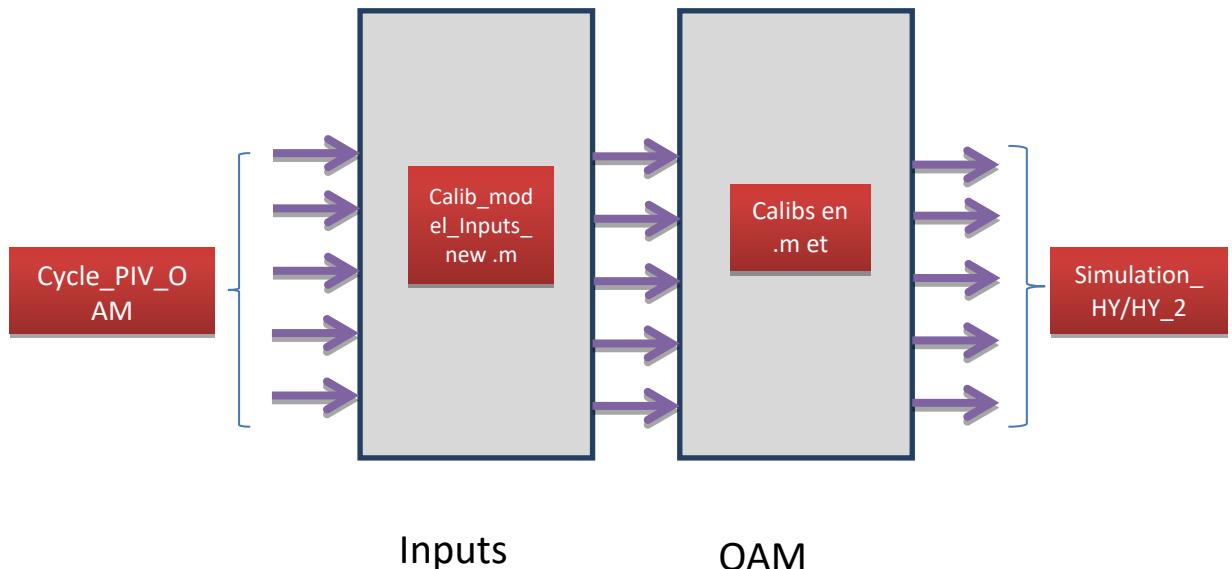


Figure 13 : Première planche du modèle

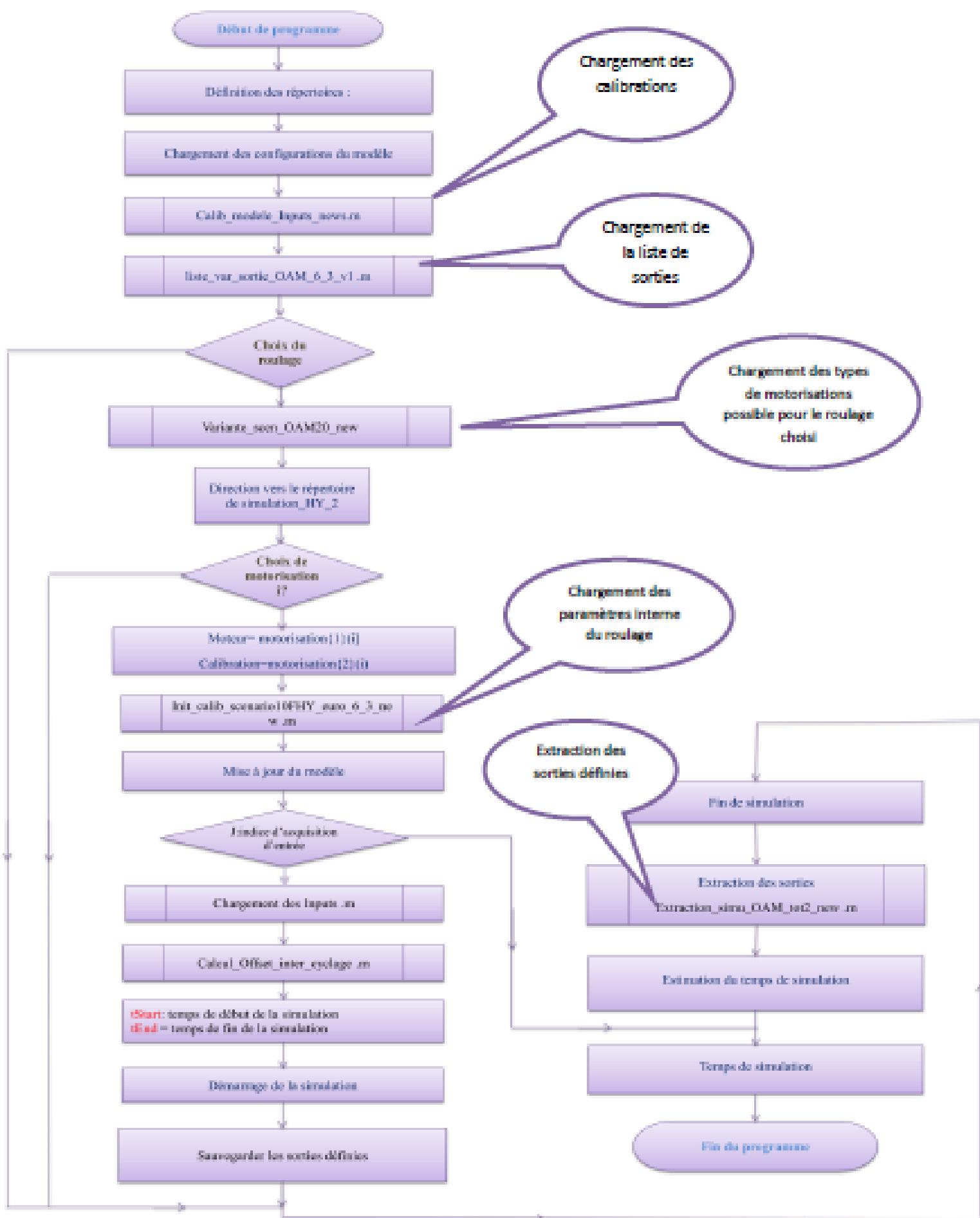
La première planche du modèle est constitué de deux sous. Le premier nommé Inputs, son rôle est de charger les entrées de la fonction. Le deuxième nommé OAM, il contient tout les blocs qui représentent les différentes fonctionnalités du système de commande.

2. Script de lancement :

C'est le programme principal qui démarre la simulation du modèle et qui fait appel aux autres scripts de calibrations et de chargement des Inputs.

2.1. Organigramme du script principal :

Pour faciliter la compréhension du programme j'ai fait un organigramme simplifié du script de lancement représenté sur la page qui suit.



2.2. Fonctionnement générale du programme:

Le programme commence par des déclarations des répertoires de travail, du modèle et des configurations de quelques variables qui concernent le type de roulage, le type des acquisitions d'entrée et d'autres spécifications. Après le programme fait appel à des scripts contenant les calibrations du modèle Inputs et OAM, et à d'autres conteneurs des paramètres interne qui vont être utilisé par la suite pour le chargement des acquisitions.

Dans le programme il y a trois boucles for sont le cœur du script. La première permet de déterminer le type de roulage qui représente un scénario, et le chargement d'un script qui détermine le cycle des acquisitions à charger et un autre qui contient les types de motorisation utilisés pour chaque type de roulage. La deuxième boucle permet de choisir le type de motorisation pour faire le test. La troisième boucle permet de charger les acquisitions suivant un cycle donnée avant, puis elle lance la simulation du modèle pour chaque acquisition.

A la fin du script principale le programme fait appel à un fichier .m qui s'occupe de l'extraction des sorties à la fin de la simulation, puis après le programme sauvegarde les résultats dans le répertoire de simulation.

Le programme contient d'autres commandes permettant de calculer le temps de simulation d'afficher l'avancement de simulation et d'écrire un rapport de simulation sur un fichier texte.

IV. Plateforme de projet :

1. MATLAB

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques. MATLAB est un interpréteur: les instructions sont interprétées et exécutées ligne par ligne. MATLAB fonctionne dans plusieurs environnements tels que X-Windows, Windows, Macintosh.

Il existe deux modes de fonctionnement:

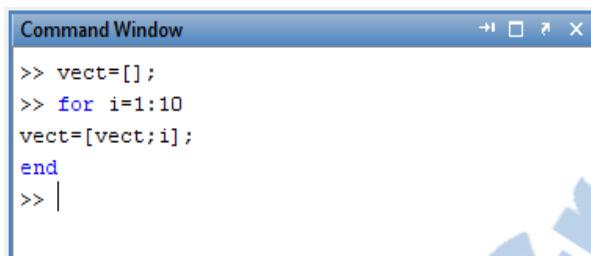
1. mode interactif: MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'usager.

2. mode exécutif: MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).

1.1. Espace de travail (workspace)

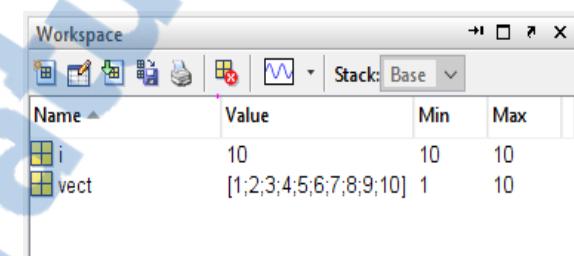
Les variables sont définies au fur et à mesure que l'on donne leurs noms et leurs valeurs numériques ou leurs expressions mathématiques. Les variables ainsi définies sont stockées dans l'espace de travail et peuvent être utilisées dans les calculs subséquents.

Un exemple du résultat sur le workspace représenté sur la figure 16 et le code écrit sur la commande window dans la figure 15 :



```
Command Window
>> vect=[];
>> for i=1:10
vect=[vect;i];
end
>>
```

Figure 14 : commande window



Name	Value	Min	Max
i	10	10	10
vect	[1;2;3;4;5;6;7;8;9;10]	1	10

Figure 15 : Espace de travail

1.1.1. Information sur les espaces de travail « Workspace » :

Pour obtenir une liste des variables dans l'espace de travail, on utilise les instructions suivantes:

Who Affichage des variables dans l'espace de travail.

Whos Affichage détaillé des variables dans l'espace de travail.

1.1.2. Enregistrer les variables de l'espace de travail dans un fichier

Pour enregistrer les variables de l'espace de travail dans un fichier, on utilise les instructions suivantes:

Save : Enregistrer toutes les variables dans un fichier matlab.mat.

1.2. Fichiers .m

Les fichiers M sont des fichiers ASCII contenant des suites d'instructions MATLAB dont le nom a comme extension (.m) Par exemple «test1.m». Dans la fenêtre Commande, si

l'on tape test1, les instructions contenues dans le fichier test1.m seront exécutées une par une. On peut créer des fichiers M à l'aide de «Text Editor»

2. Simulink

Simulink est une extension graphique de Matlab permettant de représenter des systèmes sous forme de diagrammes en blocs, et de simuler le fonctionnement de ces systèmes. Ces blocs se combinent entre eux pour former des systèmes complexes, auxquels on pourra soumettre divers signaux d'entrée, et dont on pourra visualiser la sortie.

2.1. Simulation avec Simulink

Choisir les paramètres de simulation :

Avant de lancer une simulation, on doit choisir les paramètres appropriés au modèle du système. Dans le menu Simulation, lorsqu'on choisit Parameters, une fenêtre Simulation Parameters s'ouvrira. On doit alors choisir les paramètres pour Solver, Workspace I/O et Diagnostics.

2.2. Les paramètres de «solver» :

Pour changer les paramètres de solver la figure suivante montre la fenêtre qui contient ces paramètres :

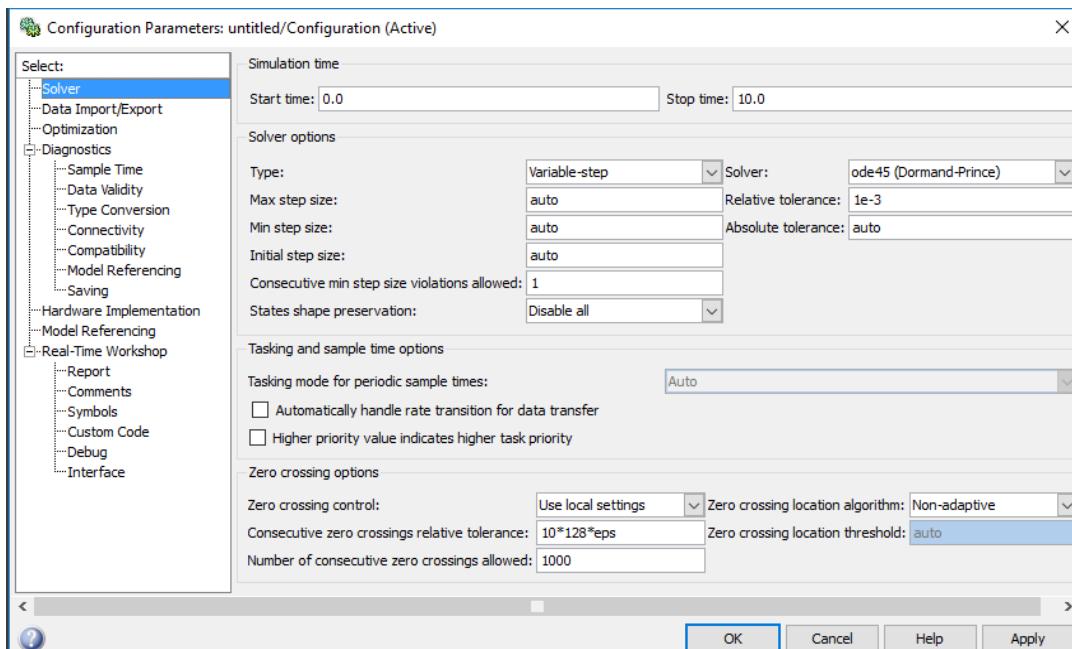


Figure 16 : Fenêtre des paramètres de solver

2.3. Les paramètres de «workspace i/o» :

Pour configurer le chargement des entrées/sorties vers l'espace de travail Matlab, on utilise la fenêtre Data Import/Export qui contient les paramètres de chargement et de sauvegarde des données. Cette fenêtre est représentée sur la figure suivante :

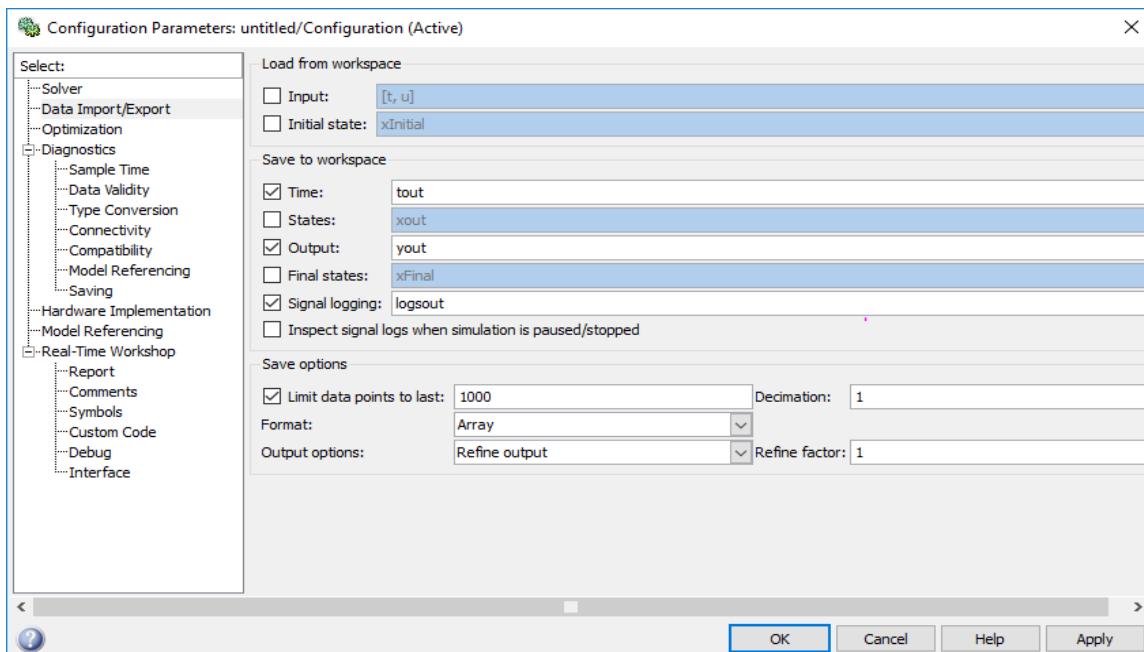


Figure 17 : Fenêtre des paramètres du Workspace

3. L'Interface Graphique sous Matlab

Les interfaces graphiques (ou interfaces homme-machine) sont appelées GUI (pour Graphical User Interface) sous MATLAB. Elles permettent à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques (boutons, menus, cases à cocher...). Ces objets sont généralement actionnés à l'aide de la souris ou du clavier.

3.1. Les objets graphiques

3.1.1. Présentation

Sous MATLAB, les objets graphiques sont classés selon une hiérarchie parent-enfant présenté sur la figure suivante :

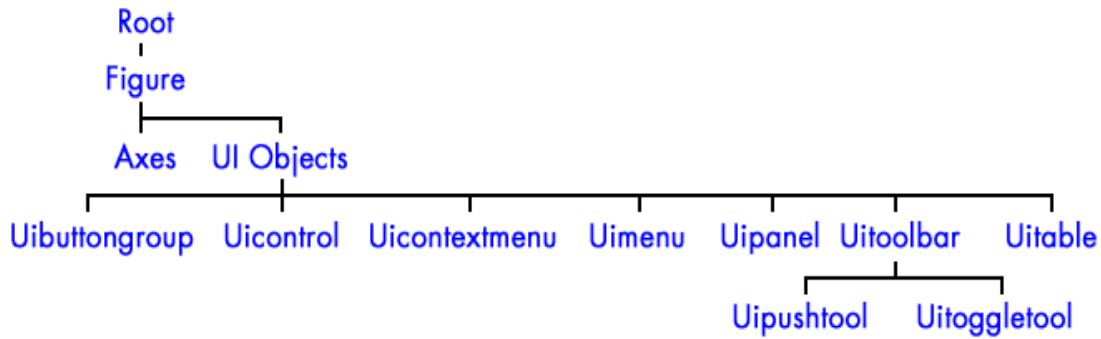


Figure 18 : hiérarchie de classement des objets graphique

3.1.2. Objet Root

Au sommet de la hiérarchie se trouve l'objet Root. Cet objet est invisible (on peut le représenter comme étant la surface de l'écran de l'ordinateur). L'utilisateur n'interagit que très rarement avec cet objet.

3.1.3. Objets Figure

Les objets Figure sont les conteneurs visibles où sont disposés tous les autres objets ‘enfants’. Ces objets sont couramment appelés « fenêtres ». Plusieurs objets Figure peuvent être ouverts simultanément et peuvent éventuellement communiquer entre eux.

3.1.4. Objets Axes

Les objets Axes sont les zones de traçage des graphiques (2D ou 3D). Un objet Figure peut contenir plusieurs objets Axes simultanément.

Les objets Axes sont ‘parents’ d’objets graphiques qui ne font pas l’objet de ce tutoriel.

3.1.5. Objets UI

Au même niveau hiérarchique que les objets Axes, on trouve les objets UI (pour User Interface). Certains de ces objets (comme les boutons, les menus, les cases à cocher) permettent à l’utilisateur d’interagir avec l’interface graphique grâce à la souris ou au clavier. D’autres objets (comme les panels, les tables...) servent à la mise en forme de l’interface graphique.

La figure suivante montre un exemple d'une interface graphique :

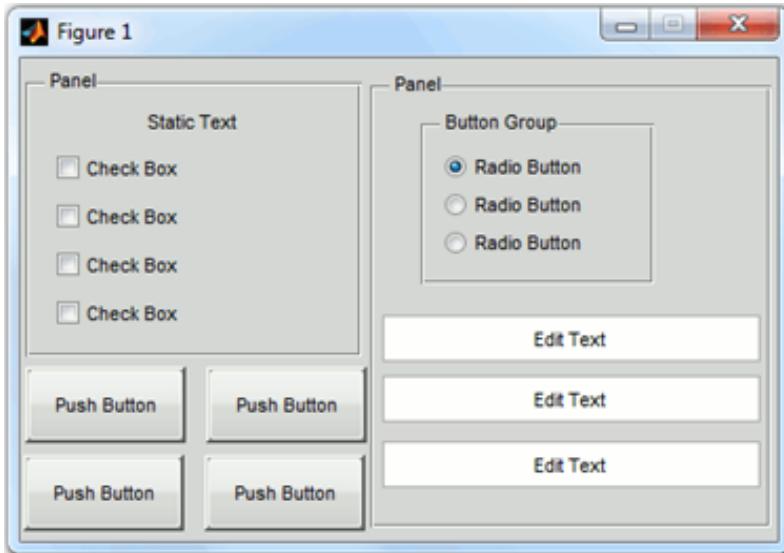


Figure 19 : Exemple d'une interface graphique Matlab

3.2. Gestion

A la création d'un objet, MATLAB lui attribue automatiquement **un identifiant** (ou **handle**), sous la forme d'une valeur numérique unique qui peut être stockée dans une variable. Ceci permet de retrouver à tout moment un objet graphique au cours du fonctionnement d'une interface. Cet identifiant existe tant que l'objet existe. Dès que l'objet est détruit, cet identifiant disparaît.

Propriétés :

Chaque objet graphique possède des propriétés (**position**, **couleur**, **action**, etc.) Elles sont définies à la création de l'objet et peuvent être modifiées dynamiquement au cours du fonctionnement de l'interface graphique. La difficulté principale consiste à apprendre et à maîtriser ces nombreuses propriétés.

Ces propriétés peuvent être récupérées et modifiées en utilisant l'identifiant de l'objet et les fonctions get et set.

3.3. Méthodes de développement

Le développement des interfaces graphiques peut être séparé en deux parties :

- gestion de la mise en place et des propriétés des objets ;

- programmation des interactions avec les objets.

Il existe deux méthodes de développement des interfaces graphiques sous MATLAB. La première utilise un outil graphique dédié et la seconde nécessite de programmer entièrement à la main.

Conclusion :

Après avoir défini le vidange de l'huile moteur, la fonction de contrôle moteur OAM et son rôle pour calculer l'intervalle de vidange. Ainsi que l'explication des différentes phases de cycle de vie de la fonction en parlant phase d'évaluation où on a présenté le plan d'intégration-validation existant, on abordera le troisième chapitre où on va développer le notre travail avec les résultats obtenus.

Chapitre 3 : Réalisation du projet et résultats obtenus

Dans ce troisième chapitre on présente l'analyse du besoin pour réaliser l'outil de simulation OAM, le fonctionnement de l'outil et le gain obtenu.

Introduction :

Le travail consiste à optimiser le processus d'évaluation et de validation du modèle Simulink par une interface graphique qui va rendre ce processus plus simple et plus rapide et interactive avec l'utilisateur.

Pour concevoir une interface qui va répondre au besoin exprimé une analyse fonctionnelle a été faite à partir de laquelle on peut identifier les besoins que doit satisfaire cette interface.

I. Analyse fonctionnelle :

IHM_OAM -interface graphique- est outil de validation de la fonction OAM Qui lance l'exécution du modèle en se basant sur les scripts de simulation.

Les scripts sont des fichiers (.m) contenant des commandes et des fonctions permettant de charger les entrées, les paramètres et les calibrations de la fonction.

Cette interface permet la calibration des différents scripts et permet de choisir les scénarii des tests du modèle Simulink pour vérifier la dynamique et le comportement de la fonction.

1. Identification des besoins :

1.1. Bête à corne :

Pour verbaliser le besoin il faut se poser trois question (... et y répondre !). Grâce à l'outil graphique « bête à corne », on représente le besoin :

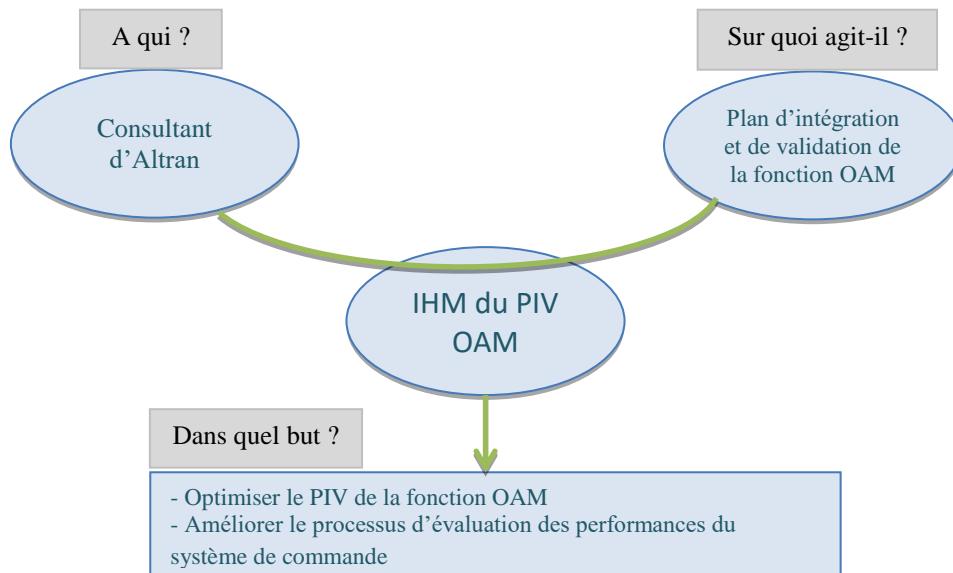


Figure 20: bête à corne

1.2. Pieuvre :

Pour caractériser les fonctions de service attendues et générées par l'usage de l'interface graphique, on utilise le diagramme « pieuvre » représenté sur la figure suivante :

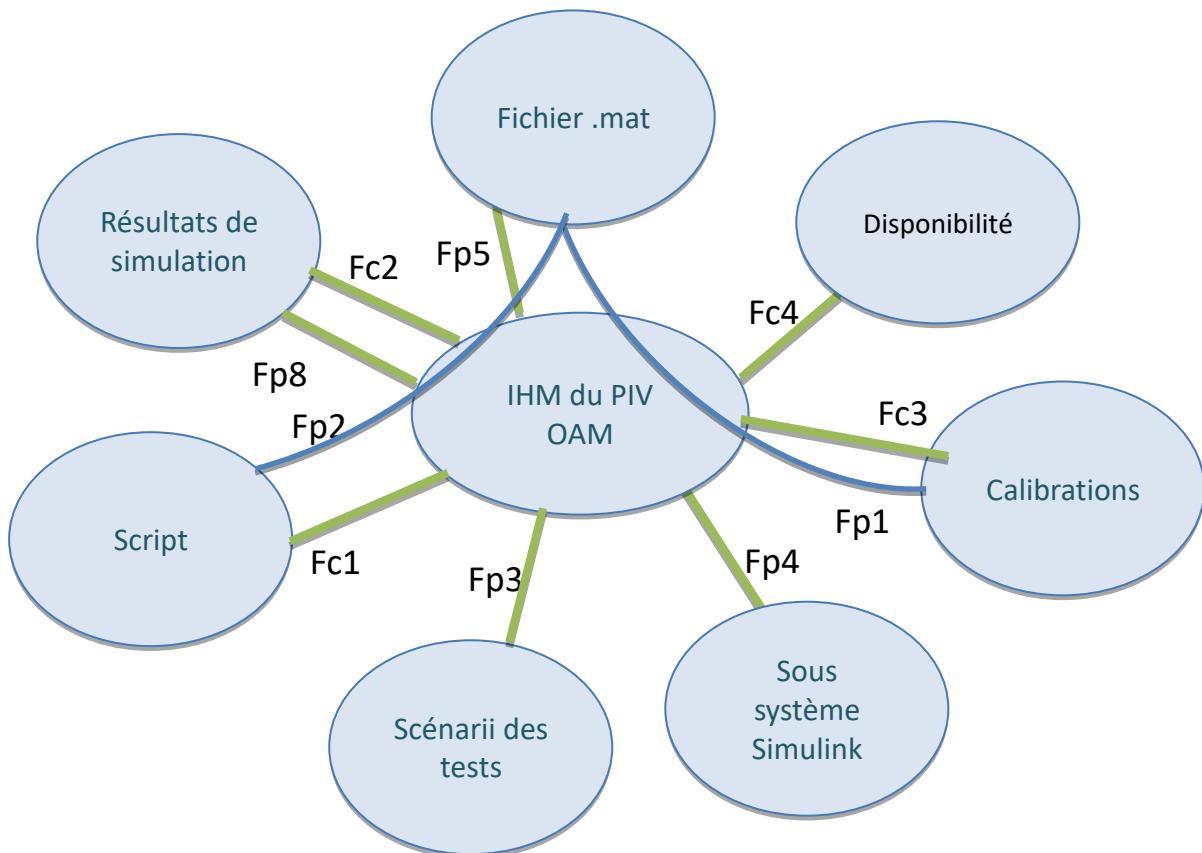


Figure 21 : Diagramme pieuvre de l'IHM

Fp1: Ecrire dans le script les données d'un fichier .mat

Fp2: Charger les calibrations à partir d'un fichier (.mat).

Fp3: Permettre de simuler plusieurs scénarii à la fois

Fp4: Suivre un sous-système Simulink

Fp5: Lire le contenu d'un fichier .mat

Fp6: Permettre une bonne visualisation des résultats

Fc1 : Assurer que le reste du programme ne soit pas affecté par l'écriture dans le script.

Fc2 : Assurer que les résultats obtenus par l'interface sont les mêmes attendus

Fc3 : modifier les calibrations en gardant les valeurs par défaut.

Fc4: Assurer la disponibilité de l'interface.

2. Définition des besoins :

Pour définir l'ensemble des besoins exprimés, j'ai supposé ces dernières comme des solutions de la problématique en répondant à la question pourquoi pour définir les manques et les besoins à satisfaire par chaque une des solutions :

Besoins	Solutions
<p>Pour choisir le scénario de test l'utilisateur doit saisir dans le script principal, dans la première boucle le numéro qui correspond au scénario de test. Alors pour laisser le script loin de l'utilisateur et afin d'éviter les erreurs humains, minimiser le risque de perdre une donnée dans le script et en vue de rendre ce choix interactive et lisible pour l'utilisateur, on a proposé de regrouper l'ensemble des scénarios dans une liste sur l'interface.</p>	<p>Choix du type de roulage et le type de motorisation à partir de l'interface</p>
<p>Pour tester l'ensemble des fonctionnalités de la fonction on doit vérifier tous les cas possibles que peut prendre les paramètres de configuration du modèle.</p> <p>Les paramètres sont trop nombreux, ce qui rend la commande du système difficile en termes de temps et de précision sur l'emplacement des paramètres dans les scripts.</p>	<p>Paramétrage des configurations, calibrations et des variables internes</p>

Besoins	Solutions
Dans le cadre d'optimisation des scénarios de tests qui prennent beaucoup de temps d'exécution, on aura besoin de supprimer, de remplacer ou d'ajouter un scénario ou modifier des paramètres dans un scénario.	Ajouter, modifier ou supprimer un roulage à partir de l'interface.
Pour créer un nouveau scénario, l'utilisateur se trouve obligé de saisir dans l'interface plusieurs paramètres et calibrations. D'une part, cela nécessite beaucoup plus de concentration et de temps et d'autre part, le risque de commettre des erreurs devient de plus en plus important, d'où vient l'idée d'automatiser cette opération.	charger un .mat pour la calibration d'un scénario en complément de la saisie manuelle
Pour avoir plus de combinaison entre les types de moteur et leur calibrations pour chaque scénario.	Modifier le couple {calibration, motorisation}
Créer un historique des tests Pour laisser une traçabilité des différentes simulations faites et ainsi faciliter le suivi des simulations par d'autres utilisateurs. Ça va servir aussi à trouver les sources d'écart en comparant les paramètres du test qui donne l'écart avec les autres tests faits précédemment.	Création d'un fichier Excel qui récapitule les différents paramètres caractérisant les tests disponibles

Besoins	Solutions
<p>Pour évaluer les entrées et le fonctionnel du modèle, on fait plusieurs simulation pour chaque scénario, en variant quelque paramètres ou quelque calibrations, et on visualise les sorties.</p> <p>Donc, pour savoir l'influence de ces paramètres sur le système une comparaison entre les différentes simulations sera nécessaire.</p> <p>Ça Permet aussi de détecter l'écart entre le résultat obtenu et le résultat attendu.</p>	Comparaison des résultats de plusieurs simulations
Afin d'enregistrer que les sorties définies comme à enregistrer on peut modifier la liste des variables à enregistrer.	Ajouter / modifier la liste des variables à enregistrer.

II. Présentation de l'interface homme machine IHM_OAM

1. Introduction:

IHM_OAM -interface graphique- est outil de validation de la fonction OAM qui lance l'exécution du modèle en utilisant des scripts.

Les scripts sont des fichiers .m contenant des commandes et des fonctions permettant de charger les entrées, les paramètres et les calibrations de la fonction.

Cette interface permet de calibrer les paramètres et choisir les scénarios des tests du modèle Simulink pour vérifier la dynamique et le comportement de la fonction.

La fenêtre principale de l'interface est présentée sur la figure suivante :

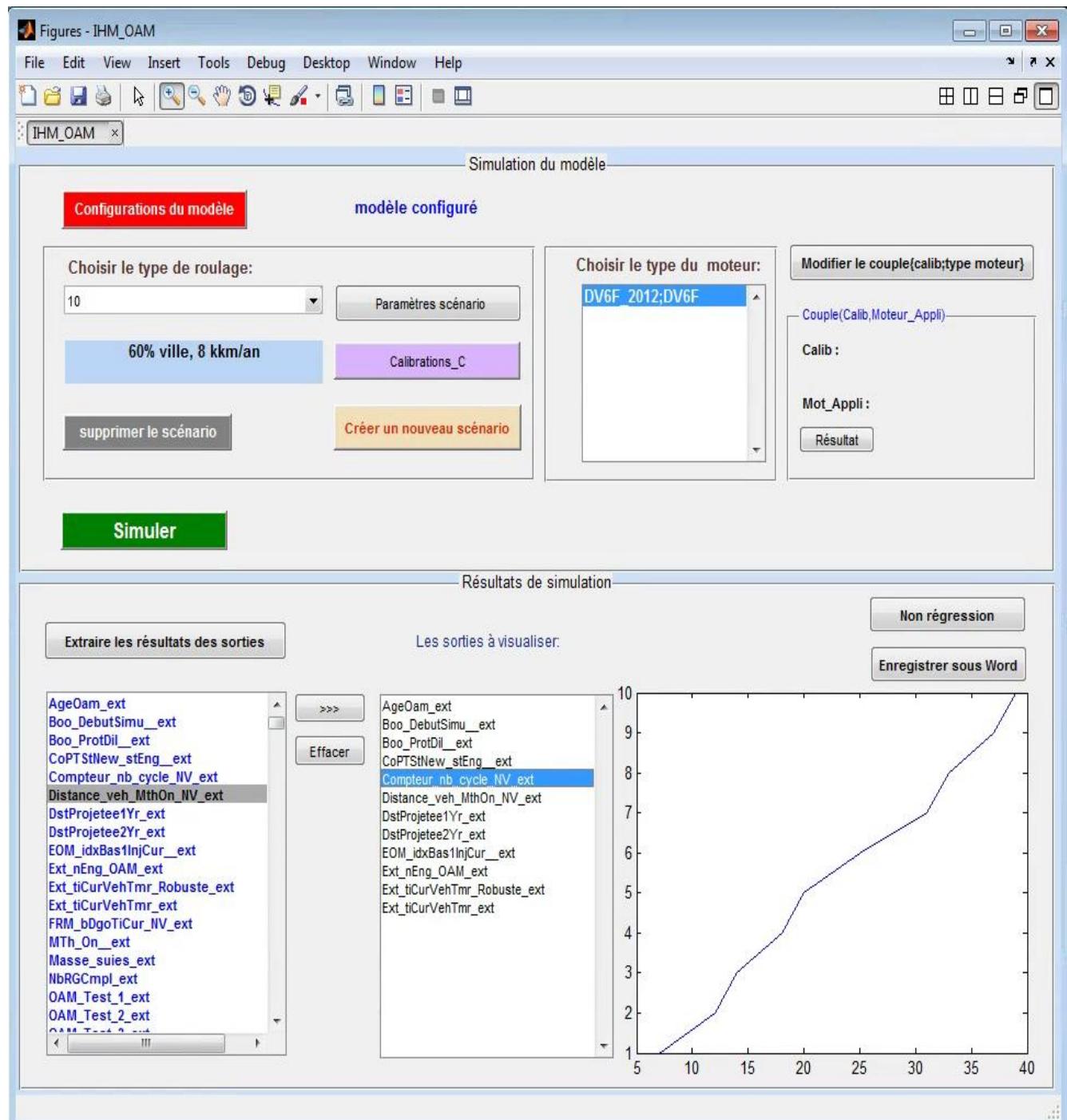


Figure 22 : Présentation de l'interface IHM_OAM



2. Organigramme de fonctionnement de l'outil

L'organigramme suivant présente la méthode d'utilisation de l'interface graphique :

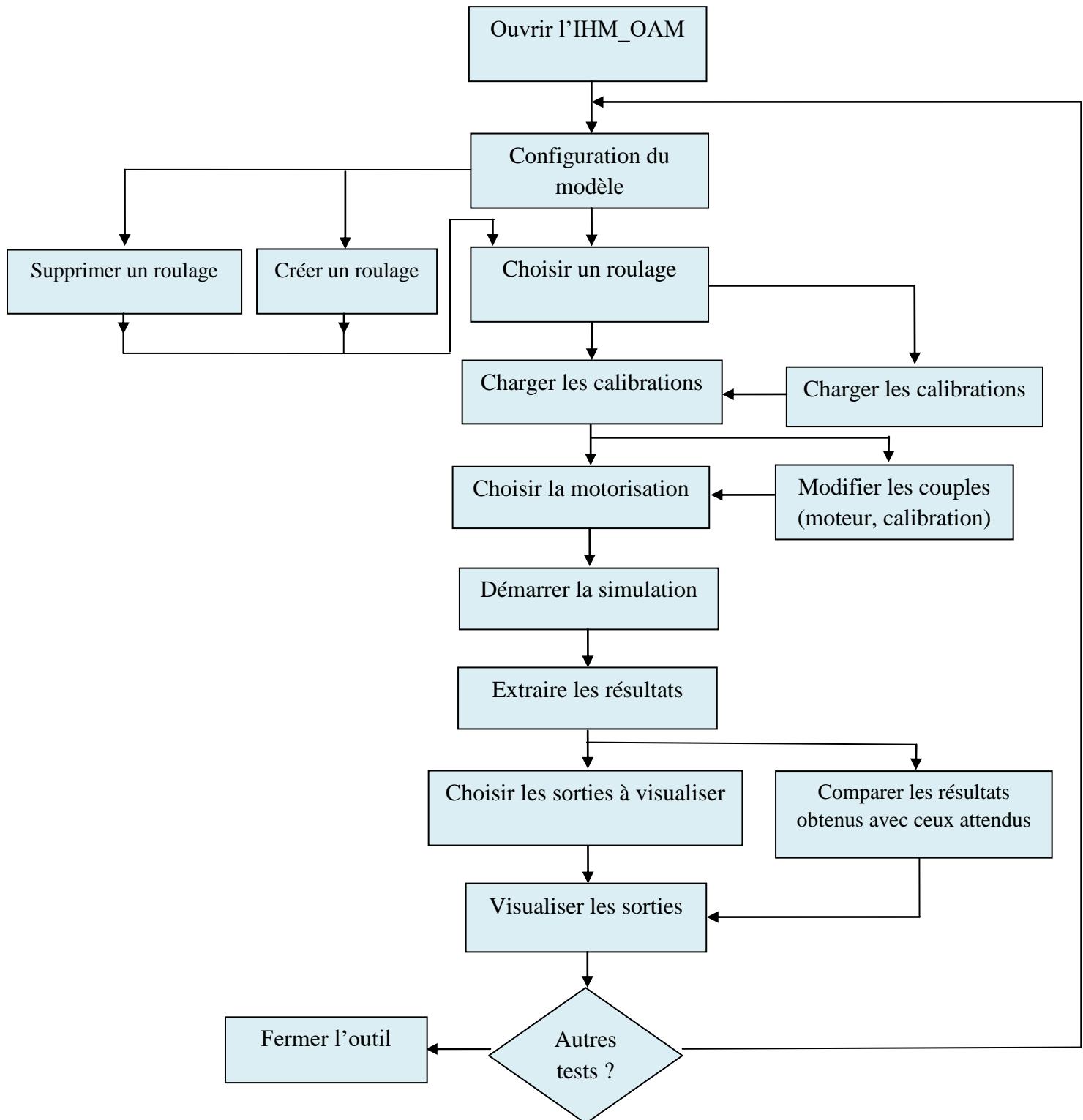


Figure 23 : Organigramme fonctionnement de l'interface

3. Lancer une simulation sur IHM_OAM :

Les étapes du lancement de la simulation sont développées comme suit :

Etape1 :

- Cliquer sur le bouton « Configuration du modèle »:

Configurations du modèle

Une fenêtre s'affiche contient des variables qui correspondent à des spécifications. Celles-ci servent au choix des calibrations et de types d'acquisitions.

Ces spécifications sont affectées à des valeurs par défaut. On peut modifier ces valeurs par le bouton « modifier ».

- Cliquer sur « appliquer » puis « OK ».

Maintenant le modèle est configuré.

Etape 2:

Sélectionner un scénario de la liste « **choisir le type de roulage** ». Voir les figures suivantes :

La liste contient des chiffres qui désignent les types de rouages.

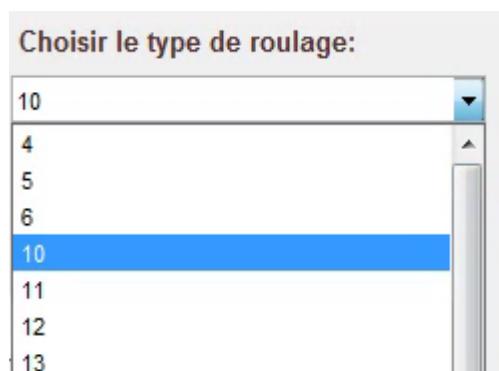


Figure 25 : Choix d'un roulage

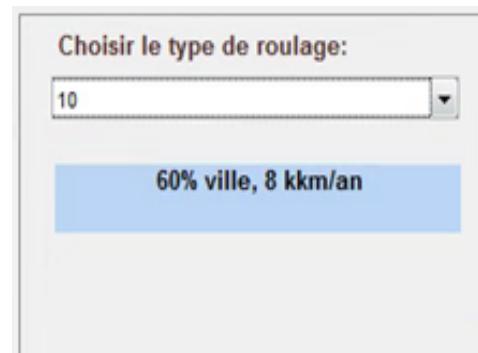


Figure 24 : Affichage du roulage choisi

Etape 3:

- Cliquer sur le bouton **Calibrations_C** pour charger les calibrations du modèle. Cette fenêtre s'affiche comme l'indique la figure suivante :

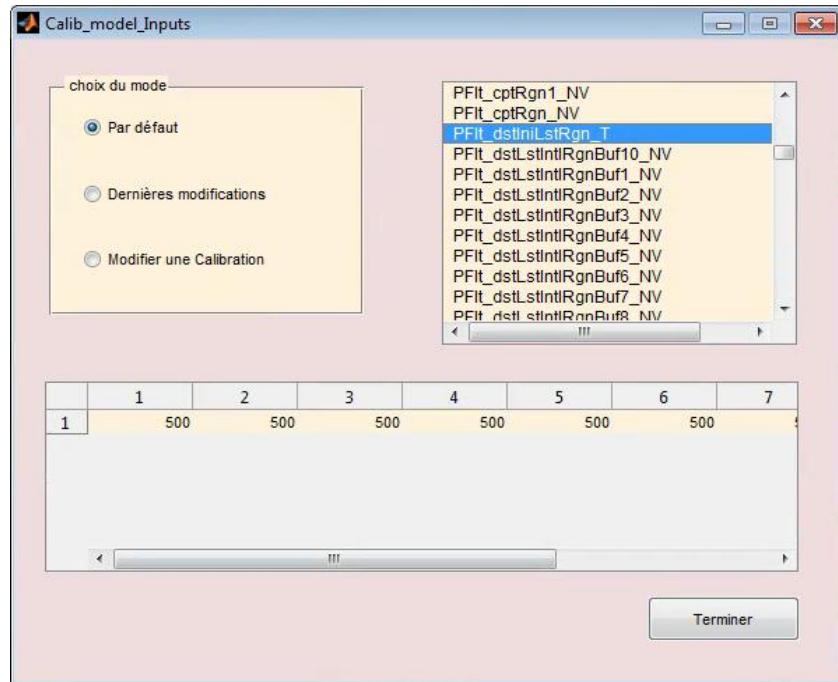


Figure 26 : Fenêtre de calibration par défaut

- Choisir le mode Par défaut pour charger les calibration par défaut.
- Pour modifier une calibration appuyer sur : **Modifier une Calibration** puis sélectionner une Variable. (voir la **figure ci-dessous**).

Saisir dans le tableau la nouvelle valeur, ensuite confirmer les modifications en cliquant sur les Boutons « ok », « appliquer » puis « enregistrer »

Pour reprendre la dernière modification des calibs Appuyer sur **Dernières modifications**

NB: les valeurs par défaut sont enregistrées, on peut les reprendre à chaque fois même

après une modification.

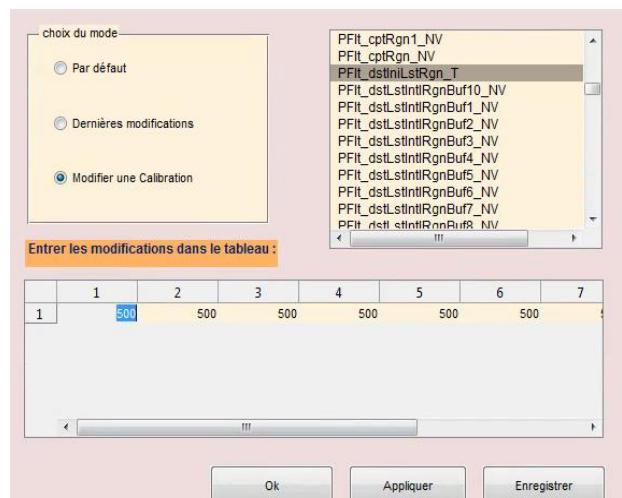


Figure 27 : fenêtre de calibration

Etape 4:

- Sélectionner un couple {calib,moteur_apply} de la liste indiqué sur la **figure 28** qui

contient les éléments du vecteur Var_scen:



Figure 28 : Liste des moteurs

- Pour modifier le vecteur Var_scen en modifiant les combinaisons du couple

{calib, type moteur} cliquer sur le bouton .Dans la fenêtre qui s'affiche (figure 29), sélectionner la calibration et la motorisation, ensuite appuyer sur le bouton pour remplir la liste des couple puis cliquer sur le bouton pour écrire le vecteur des couples (Var_scen).

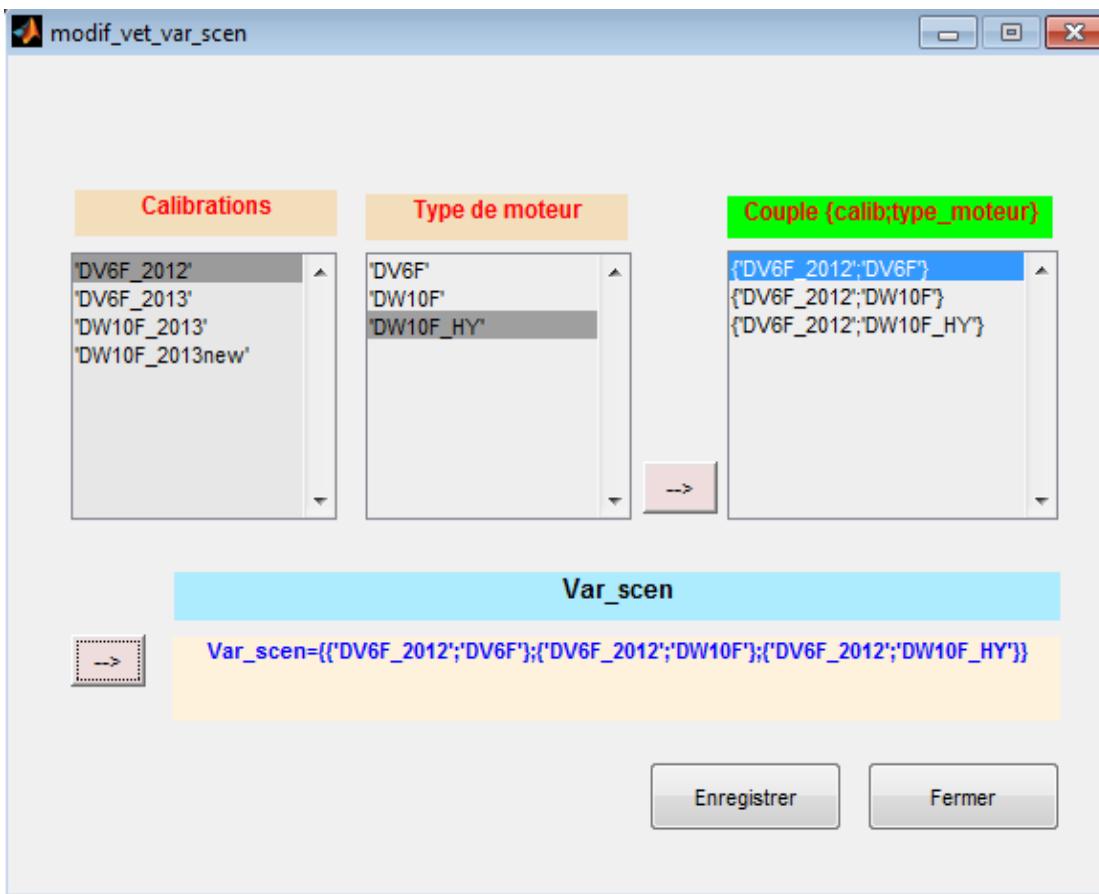


Figure 29 : Modification du couple moteur calibration

Etape 5:

- Sélectionner le couple {calibration; type_moteur} de la liste « choisir le type de moteur »

- Pour modifier la liste des couples cliquer sur le bouton « modifier le couple {calib; type_moteur} ».
- Cliquer sur le bouton « simuler » pour démarrer la simulation puis attendre la fin de l'exécution.

4. Récupérer et visualiser les sorties de la fonction OAM :

On peut visualiser les sorties de la fonction OAM en utilisant les étapes suivantes :

Etape 1:

- Appuyer sur le **Extraire les résultats des sorties** bouton:
- Choisir le fichier .mat de l'extraction dans la fenêtre représenté sur la **figure 30**:

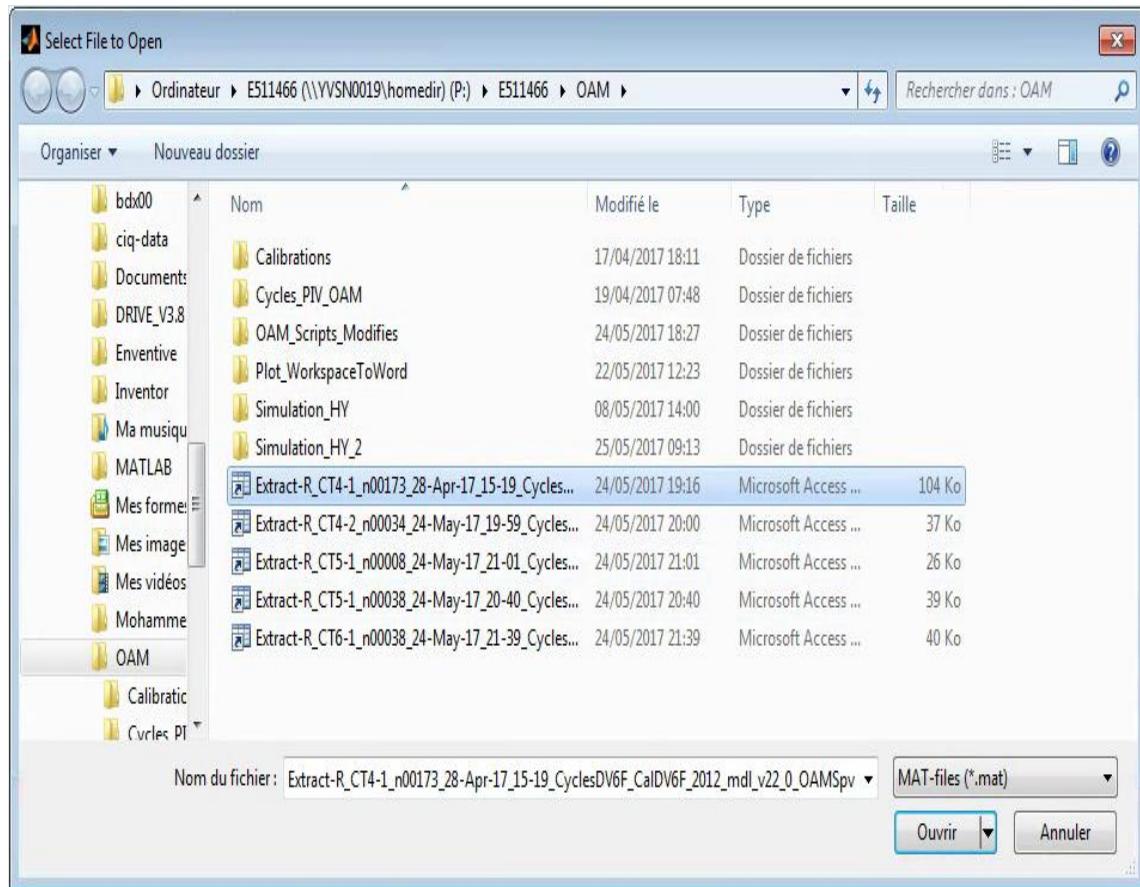


Figure 30 : Choix du fichier à ouvrir

Etape 2:

- Sélectionner les sorties à visualiser puis cliquer sur le bouton:

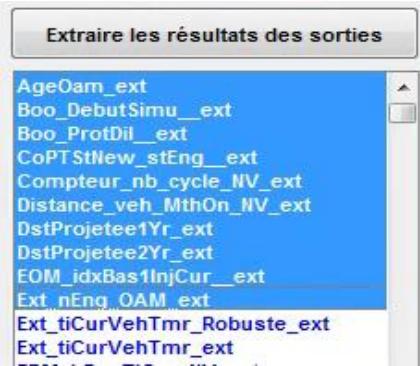


Figure 31 : Liste des flux de sortie



Figure 32 : Liste des sorties à visualiser

- Sélectionner l'une des sorties de la liste « sorties à visualiser »

La courbe de signal de sortie s'affiche sur l'interface comme le montre la **figure 33** :

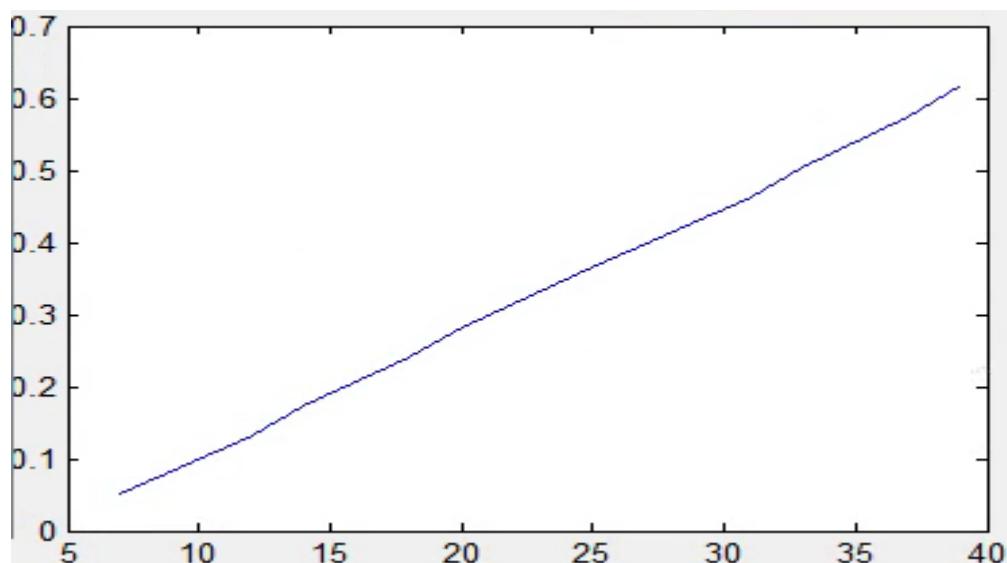


Figure 33 : exemple pour Flux Masse_suies_ext

Cette courbe présente la variation de la masse des suies⁴ émise dans le moteur en fonction du kilométrage du véhicule.

Etape 3:

Appuyer sur le bouton: pour enregistrer les sorties définis comme à enregistrer sous Word.

⁴ La suie est un dépôt carboné de couleur noire et d'apparence huileuse. Elle se forme lors de la combustion incomplète de la biomasse (bois) et des hydrocarbures fossiles (charbon, pétrole).

5. Modifier les paramètres d'un roulage:

Pour modifier une variable ou un paramètre qui caractérise un roulage on procède de la façon suivante :

- Appuyer sur le bouton suivant :

Une fenêtre apparaît indiquée sur la figure suivante :

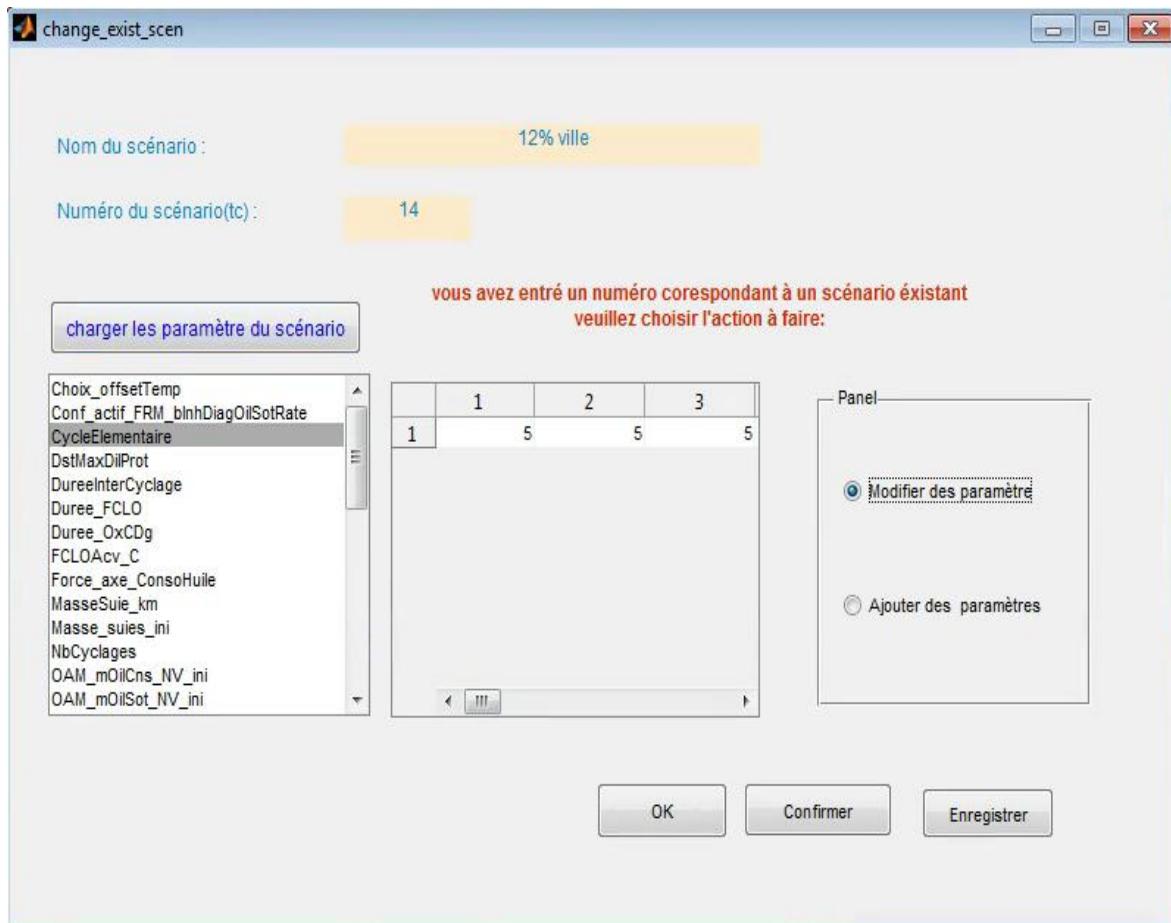


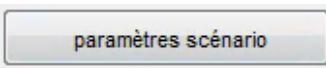
Figure 34 : Changement d'un scénario

- Cliquer sur pour charger les paramètres du scénario.
- Cocher sur le panel le choix :
- Saisir dans le tableau les nouvelles valeurs puis appuyer sur « ok », « appliquer » puis « enregistrer ».

6. Ajouter des paramètres d'un roulage :

Pour ajouter une variable ou un paramètre dans le script nommé « Init_calib_scenario10FHY_euro_6_3_new.m », suivre les étapes suivantes:

- Appuyer sur le bouton :



- Cocher sur le panel le choix



- Choisir le type de donnée dans la liste suivante:

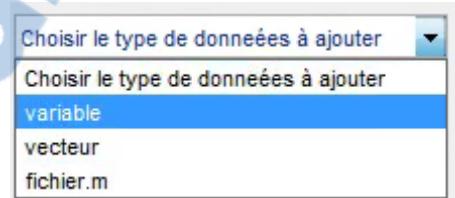


Figure 35 : Liste des types des données

- Saisir le nom et la valeur pour les données de type variable ou vecteur
- Sélectionner de la liste « fichier .m » si vous voulez ajouter dans le script un fichier à exécuter ou n’importe quelle commande.
- Après la saisie des données cliquer sur le bouton ensuite appuyer sur «OK », « confirmer » et « enregistrer ».

7. Créer un nouveau scénario :

Pour créer un nouveau scénario avec de nouveau cycle et des nouveaux paramètres cliquer sur le bouton

Dans la nouvelle fenêtre entrer le nom du scénario et un numéro puis sélectionner les couples {calib, moteur} puis appuyer sur « ajouter ».

Dans la fenêtre suivante ([parametre_calib_new_scen](#)), Entrer les paramètres du scénario

NB: à chaque fois que vous faites entrer un paramètre cliquer sur ajouter et à la fin appuyer

sur enregistrer.

La figure suivante montre la fenêtre « parametre_calib_new_scen » :

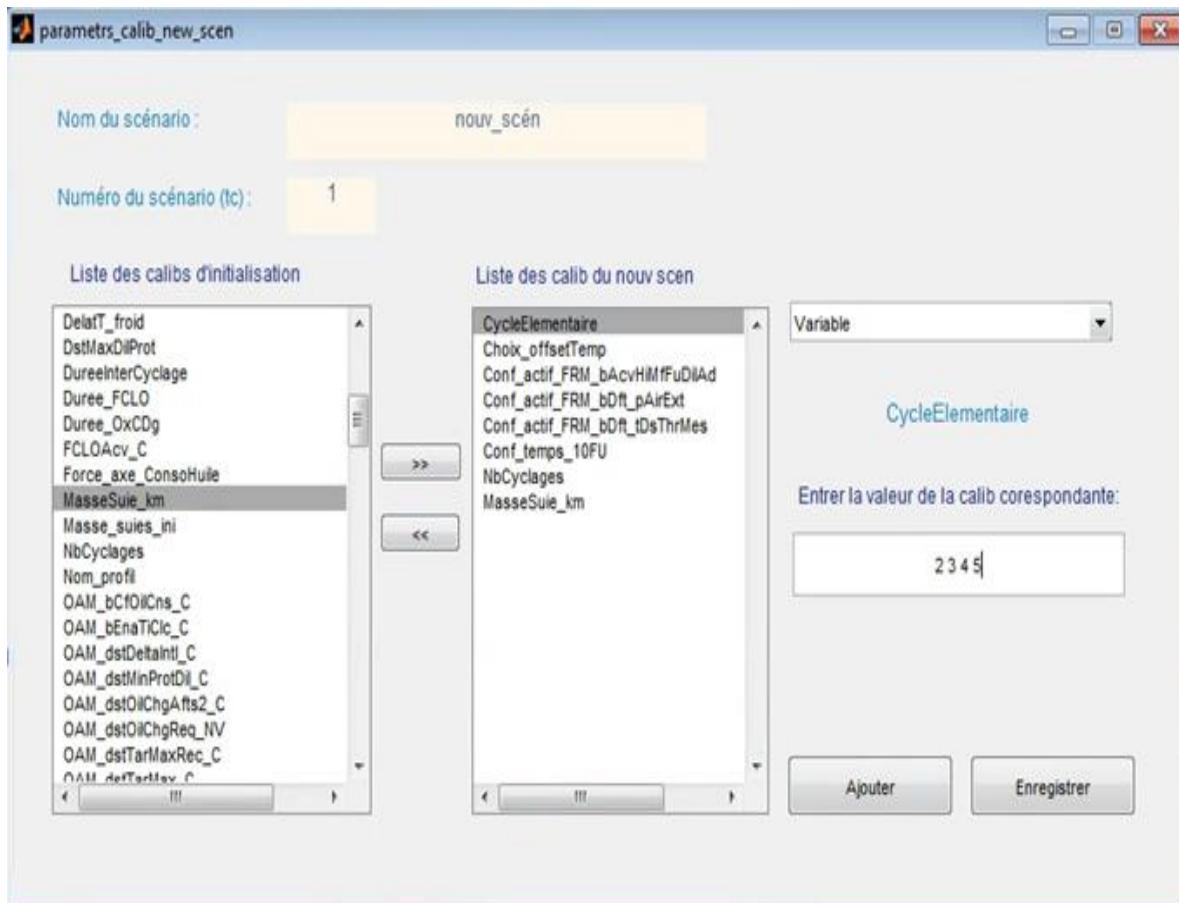


Figure 36 : Les calibrations du nouveau scénario

NB: à chaque fois que vous faites entrer un paramètre cliquer sur ajouter et à la fin appuyer sur enregistrer

III. Gains obtenus avec l'outil :

Les gains obtenus par ce projet sont présentés dans le tableau suivant :

Tache	Avec les scripts	Avec l'outil	Gains obtenus
Choisir le type de roulage et les motorisations	Durée : entre 5 et 10 min. Changement dans le programme	Durée : secondes, Script non touché	- Rapidité (-10min), - Sécuriser les programmes.
Modifier les paramètres et les calibrations du roulage	Durée : de 10min jusqu'à 30min ou plus, selon le nombre de paramètres à configurer	Durée : de 1 min jusqu'à 5 min	- Réduction 25min. - Lisibilité, simplicité de travail
Visualiser les sorties	Durée : 15 min- 30min, Exécution d'un script qui plot sous Word.	Durée : 5-10 secondes Visualiser directement sur l'outil.	- Rapidité (minimisation de 20min) - Visualisation facile, lisible + précision.
Comparer les résultats obtenus avec ceux attendus	Durée : 15 min- 30min, Exécution d'un script qui compare et plot sous Word.	-Durée : 5min - -Visualiser directement sur l'outil. -Délecter les écarts automatiquement	-Rapidité (minimisation de 30min) -Visualisation facile, lisible + précision.

Tableau 3 : Gains obtenus par IHM_OAM

Autres gains :

- Augmentation du taux de couverture du système par création de nouveau roulage.
- Simplifier les étapes d'optimisation des roulages.

Donc on peut réduire 85 min pour chaque test. Sachant qu'il y a 23 scénario et pour chaque scénario on fait plusieurs tests, par un calcul simple on trouve la durée réduite par

mois et par projet :

- Durée moyen d'un test 8h
- La durée minimisée pour un test est 85min et le gain en pourcentage est 17.7%
- Par mois on minimise 29 heures c.-à-d. 4 jours
- Par projet :

Si la durée du projet est 4 mois en utilisant cette interface on peut **terminer** le projet

16 jours avant, qui est équivalente à 3 semaines.

Conclusion générale

Notre projet intitulé « Optimisation du plan d'intégration-validation d'une fonction contrôle moteur » consiste à la conception et la réalisation d'une interface homme machine destiné pour l'évaluation et la validation des performances du système de commande de la fonction contrôle moteur.

En ce qui concerne la démarche, nous avons en premier lieu effectué une phase d'étude des différents outils de travail. En deuxième lieu nous avons étudié la fonction contrôle moteur concerné par le projet puis nous avons étudié le fonctionnement du plan d'intégration-validation existant. En troisième nous avons spécifié les fonctionnalités de notre application pour améliorer ce plan .En quatrième lieu, nous avons procédé à sa conception ainsi qu'aux choix techniques pour sa réalisation. Enfin, nous l'avons validée.

Contrairement au plan d'intégration-validation existant qui offre des fonctionnalités limités et nécessite un effort de configuration considérable, nous avons réalisé un système plus performant. En effet, celui qu'on a proposé permet à la fois de gérer les paramètres des différents roulages, créer des nouveaux roulages, simuler le modèle de la fonction, et avoir la possibilité de visualiser et d'analyser les résultats de simulation.

Toutes les fonctionnalités décrites dans le cahier des spécifications fonctionnelles ont été développées et validées. Néanmoins, notre projet pourra être amélioré par l'ajout d'autre fonctionnalités comme : l'Analyse en temps réel d'un sous-système Simulink et la création d'un fichier Excel qui récapitule les différents paramètres caractérisant les tests disponibles.

Il est important de noter que ce projet a été une véritable occasion de vivre de près l'expérience du travail au sein d'une société, qui exige la ponctualité et l'intégration dans un groupe de travail. Ce qui nous a permis d'améliorer nos capacités de communication et de nous adapter à la vie professionnelle. Nous avons fait de notre mieux pour bien laisser une bonne impression sur notre discipline, nos qualités et nos compétences techniques vis à vis du staff technique de la société Altran Maroc et présenter un travail à la hauteur de la formation qui nous avons eu au sein de l'FSTF.

Bibliographie

[B1] : documentation de l'équipe LCC sur le contrôle moteur

[B2] : documentation PSA sur la fonction OAM PSA

Webographie

[w1] : <http://altran.ma>

[w2] : <http://www.purflux.com/fr/actualites/sogefi-fournisseur-officiel-pour-les-nouveaux-moteurs-euro-6-de-psa.html>

[w3] : <http://www.ds3spirit.com/DiversDocuments/DW10F-Euro6.pdf>

[w4] : http://www.sherpa-eng.com/index.php?option=com_content&view=article&id=103&Itemid=858&lang=fr

[w5] : <https://entretien-voiture.ooreka.fr/comprendre/carnet-entretien-voiture>

[w5]: <https://www.mathworks.com/>