

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE**

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**



**Ecole Nationale Polytechnique
Département de Génie Electrique
Spécialité AUTOMATIQUE**



**Laboratoire de Commande
des Processus (LCP)**

Mémoire de Projet de Fin d'Etudes

En vue de l'obtention du Diplôme d'Ingénieur d'Etat en
Automatique

Thème :

Commande Neuro-Floue d'une suspension active de véhicule

Réalisé par:

**Mounir KARA-ZAÏTRI
Nouar MESSAOUDI**

Proposé et dirigé par:

**Mr. D. BOUKHETALA
Mr. F. BOUDJEMA**

Promotion Juin 2006

Nous remercions le bon Dieu de nous avoir accordé
toute la patience, le courage, la volonté et la
motivation qui nous ont permis d'achever ce travail.

Dédicace

A qui puis je dédier cet humble travail si ce n'est à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide, l'encouragement et les prières sont l'essence de ma réussite. Sans eux, je ne serais pas ce que je suis aujourd'hui.

*Je dédie ce travail également,
A mon frère Med el hadi, mes soeurs Meriem et Halima.
A la mémoire de ma regrettée grand mère.
A tous mes oncles, tantes, cousins et cousines.
A tous mes amis, que ce soit à Tébessa ou à Alger, les automaticiens, avec qui j'ai partagé les moments d'inquiétude et de joie
A toute personne qui porte de l'estime pour moi.*

Nouar

A mes très chers parents, à qui je dois ce que je suis, tous ce que je réalise, merci de m'avoir soutenu et aimé durant toutes ces années, et merci d'être si fiers de moi.

A ma sœur Myriam et mon frère Rafik, source d'amour, de joie et de bonne humeur.

Je ne peux pas oublier Abdou et le petit Samy pour toute la joie qu'ils ont apportée à ma famille.

A mes défunts grands parents qui ont tant voulu me voir arrivé là où je suis.

A toute ma grande famille paternelle et maternelle, mes oncles et tantes, cousins et cousines surtout mes très chers Yassine et Riad.

A tous mes amis à Oran et Alger ; Aux automaticiens beaucoup plus amis que collègues.

A tous ceux que j'aime et qui m'aiment ...

Mounir

Remerciements

Ce travail a été réalisé au sein du laboratoire de Commande des Processus (LCP) de l'Ecole Nationale Polytechnique d'Alger sous la direction de Monsieur Djamel BOUKHETALA et Monsieur Farès BOUDJEMA.

Nous tenons à les remercier vivement pour nous avoir encadrés et dirigés durant l'élaboration de ce travail ainsi que pour leur assistance et tous leurs conseils.

Nous remercions les membres du jury, qui nous ont honorés par leur participation à l'examen de notre travail.

Nous remercions également tous les professeurs de la spécialité Automatique de l'ENP, pour leur aide, leurs conseils et leur compréhension durant notre cursus.

Enfin, Nous exprimons notre gratitude à toutes les personnes ayant contribué de près ou de loin dans ce travail.

Sommaire

	Page
Introduction	1
Chapitre I : Etude de la suspension de véhicule	6
I.1 Introduction.....	7
I.2 Système de suspension.....	7
I.3 Fonctionnement de la suspension.....	10
I.4 Les différents types de la suspension.....	11
I.4.1 Suspension passive	11
I.4.2 Suspension semi-active.....	12
I.4.3 Suspension active.....	13
I.5 Modélisation de la suspension.....	14
I.5.1 Modélisation linéaire.....	14
I.5.1.1 Modèle quart de véhicule (mono roue)	14
I.5.1.2 Modèle demi véhicule.....	15
I.5.1.3 Modèle véhicule complet (3d)	17
I.5.2 Modélisation non linéaire.....	20
I.5.2.1 Dynamique de l'actionneur hydraulique.....	21
I.5.2.2 Modèle quart de véhicule non linéaire	22
I.5.2.3 Modèle demi véhicule non linéaire.....	23
I.5.2.4 Modèle véhicule complet (3d) non linéaire.....	26
I.5.3 Simulation des modèles en boucle ouverte.....	28
I.5.3.1 Introduction.....	28
I.5.3.2 Modèle quart de véhicule linéaire.....	28
I.5.3.2.1 Montée sur trottoir 20 cm	29
I.5.3.2.2 Passage sur une bosse 20 cm / base 1m	29
I.5.3.3 Modèle demi véhicule	30
I.5.3.3.1 Passage sur une bosse à faible amplitude.....	30
I.5.3.3.2 Passage sur une bosse à forte amplitude.....	31
I.5.3.4 Modèle véhicule complet (3d) linéaire	33
I.5.3.4.1 Réponse à un échelon sur roues gauches.....	33
I.5.3.4.2 Réponse à un échelon sur toutes les roues.....	35
I.5.3.5 Modèle véhicule complet non linéaire.....	36
I.5.3.5.1 Réponse à un échelon de commande sur tous les actionneurs.....	36
I.6 Conclusion.....	37
Chapitre II : Commande neurofloue	38
II.1 Introduction.....	39
II.2 Notions sur les réseaux de neurones artificiels.....	39
II.2.1 Introduction.....	39
II.2.2 Le neurone	39
II.2.3 Historique.....	40
II.2.4 Utilisation	43
II.2.5 Le Neurone formel.....	44
II.2.5.1 Fonctionnement.....	44
II.2.6 Architecture des réseaux de neurones.....	45
II.2.7 Notions sur l'apprentissage.....	49
II.2.7.1 Types d'apprentissage.....	49
II.2.7.1.1 Apprentissage par correction d'erreurs.....	49
II.2.7.1.2 Apprentissage par descente de gradient	50
II.2.7.1.2.1 Algorithme d'apprentissage de Widrow-Hoff	52
II.2.7.1.2.2 L'algorithme de rétropropagation du gradient	53
II.2.7.2 Application des algorithmes d'apprentissage sur les RNA	57
II.2.7.2.1 Perceptron (F. Rosenblatt, 1958).....	57
II.2.7.2.2 Perceptron à niveaux multiples	57
II.2.7.2.3 Rétro propagation	58
II.2.7.2.4 Hopfield (J.J. Hopfield, 1982)	59
II.2.7.2.5 Kohonen	59
II.2.7.2.6 Différences temporelles	60
II.2.8 Conclusion	60

II.3 Bases de la logique floue et du réglage flou	60
II.3.1 Introduction.....	60
II.3.2 Bref historique.....	61
II.3.3 Principes.....	61
II.3.3.1 Ensemble flou.....	61
II.3.3.2 Opérations sur les ensembles flous.....	64
II.3.3.3 Règles d'inférence.....	65
II.3.3.4 Structure d'un régulateur flou.....	66
II.3.3.4.1 Principe.....	68
II.3.3.4.1.1 Interface de fuzzification	68
II.3.3.4.1.2 Moteur d'inférence	72
II.3.3.4.1.3 Interface de défuzzification.....	73
II.3.4 Conclusion	75
II.4 Systèmes hybrides neuroflous	75
II.4.1 Introduction.....	75
II.4.2 Principe de fonctionnement du réseau neuroflou	76
II.4.2.1 Structure ANFIS	77
II.4.2.2 Principe de fonctionnement	78
II.4.2.3 Entraînement des réseaux ANFIS.....	79
II.4.3 Application de la commande neurofloue au système de suspension de véhicule	80
II.4.3.1 Variable de commande	80
II.4.3.2 Architecture du régulateur neuroflou utilisé.....	81
II.4.3.2.1 Modèle quart de véhicule	81
II.4.3.2.1.1 Fonctionnement du NFC.....	83
II.4.3.2.1.2 Apprentissage	83
II.4.3.2.2 Modèle demi véhicule	85
II.4.3.2.3 Modèle complet.....	85
II.4.4 Critères de performance	85
II.4.4.1 Critères pour le quart de véhicule	85
II.4.4.2 Critères pour le modèle demi véhicule	86
II.4.4.3 Critères pour le modèle complet	86
Chapitre III : Simulations et résultats	87
III.1 Introduction	88
III.2 Simulations	89
III.2.1 Modèle quart de véhicule.....	89
III.2.1.1 Configuration « confort optimal »	89
III.2.1.2 Configuration « tenue de route optimal »	90
III.2.1.3 Configuration « hybride »	91
III.2.1.4 Réponse de la suspension passive	92
III.2.1.5 Comparaison des valeurs des critères de performance.....	92
III.2.1.6 Remarques.....	92
III.2.2 Modèle demi véhicule.....	93
III.2.2.1 Configuration « confort optimal »	93
III.2.2.2 Configuration « tenue de route optimal »	95
III.2.2.3 Configuration « hybride »	97
III.2.2.4 Réponse de la suspension passive	99
III.2.2.5 Comparaison des valeurs des critères de performance.....	100
III.2.2.6 Remarques.....	100
III.2.3 Modèle véhicule complet	101
III.2.3.1 Configuration « confort optimal »	101
III.2.3.2 Configuration « tenue de route optimal »	103
III.2.3.3 Configuration « hybride »	105
III.2.3.4 Réponse de la suspension passive	107
III.2.3.5 Comparaison des valeurs des critères de performance.....	108
III.2.3.6 Remarques.....	108
III.2.4 Tests d'interconnexion.....	109
III.2.5 Test de robustesse vis-à-vis des paramètres.....	111
III.2.5.1 Sur modèle quart de véhicule.....	111
III.2.5.2 Sur modèle demi véhicule.....	111
III.2.5.3 Sur modèle complet	112

III.2.5.4 Remarques.....	113
III.3 Conclusions et remarques	113
Conclusions et perspectives	115
Références bibliographiques	118

Introduction Générale

L'industrie automobile a vécu ces dernières années une véritable explosion accompagnée de l'introduction de nouveaux segments et concepts ; tout ceci a eu une grande influence sur les usagers qui sont devenus plus exigeants, d'ici est née une véritable concurrence entre constructeurs de l'automobile, et dans la perspective d'attirer plus de clientèle et afin de présenter le meilleur véhicule, de grands travaux de recherche et de développement technologique avancé sont entrepris où tous les organes du véhicule sont considérés.

Parmi les composants essentiels de l'automobile, la suspension qui est le système assurant la liaison des roues à la caisse, de façon à ce que celle-ci soit isolée des chocs qui se produisent quand le véhicule circule sur une surface inégale.

La suspension maintient la caisse à une faible distance au-dessus du sol, et à une hauteur relativement constante, pour l'empêcher de tanguer, de se balancer, et de pencher. Pour conserver de bonnes capacités d'accélération, de freinage, et de virage — les composantes d'une bonne tenue de route —, la suspension doit également maintenir les quatre pneus fermement en contact avec le sol.

Le rôle de la suspension fut à l'origine d'assurer le contact entre le châssis et la chaussée, à travers les roues. Ce rôle ne représente plus la seule vocation de suspension qui, suite au développement technologique détermine plusieurs facteurs, citons :

- La tenue de route ; facteur déterminant de la sécurité du véhicule car elle détermine la stabilité globale et la direction ; le freinage ainsi que le maintien du poids du véhicule, elle représente la faculté du véhicule à garder le contact avec la chaussée.
- Le confort, à travers le filtrage des vibrations causées par les irrégularités de la route et les changements de vitesse.

Il est difficile de concevoir une suspension assurant à la fois un bon confort de roulement et de bonnes caractéristiques de tenue de route. La berline familiale type possède une suspension relativement souple, qui favorise un roulement en douceur et confortable, mais sacrifie un peu les capacités d'accélération, de freinage et de virage. Inversement, la voiture de sport ou le véhicule utilitaire de loisir possèdent, dans leur

version standard, une suspension ferme qui assure une bonne tenue de route au détriment partiel du confort de roulement. Les suspensions avant et arrière des automobiles sont habituellement différentes, parce que ces deux zones ont des exigences différentes en matière de suspension [3].

Initialement le système de suspension automobile -conçu de ressorts et amortisseurs jouant le rôle de filtre mécanique- ; était passif. Ces systèmes – conventionnels – présentent l'inconvénient d'être à paramètres fixes et figés, au fait la raideur et le taux d'amortissement ne varient qu'avec l'usure des composants, ce qui ne permet pas d'adapter le comportement de suspension aux conditions où opère le véhicule ; qui peuvent varier et devenir extrêmes, d'où la nécessité d'introduire la notion de commande à ce mécanisme.

Pour ce faire on peut agir de plusieurs façons et ceci va modifier la structure même de la suspension qui ne sera plus '**passive**' mais va devenir '**active**' et réagira aux variations de l'environnement. Le système qui va régir cette modularité de la suspension est un système de commande qui aura pour tâche de prendre les différentes décisions quant aux différentes actions effectuées par l'**actionneur** introduit dans la suspension pour le fonctionnement optimal du système.

Les caractéristiques de confort et de tenue de route des suspensions ordinaires sont déterminées lors de leur conception. Cependant, les caractéristiques des suspensions **actives** peuvent être modifiées, manuellement ou automatiquement, de façon à favoriser le confort ou la tenue de route.

Une suspension active possède plusieurs composants en plus de ceux d'une suspension normale. On peut citer entre autres le processeur, qui interprète les informations provenant de différents capteurs, lesquels relèvent en permanence des données, telles que la hauteur de la voiture, son tangage et son roulis, la vitesse de rotation des roues et la rapidité du virage.

Les suspensions électroniques les plus simples se contentent de maintenir une hauteur de caisse constante, en compensant la tendance de l'arrière à se baisser sous l'effet du poids des passagers et des bagages. Les systèmes avec réglage de la hauteur

des quatre roues abaissent la hauteur de caisse sur autoroute, de façon à réduire la traînée aérodynamique et à améliorer la consommation. Sur les véhicules tout-terrain, ces systèmes peuvent relever le véhicule pour augmenter la garde au sol en terrain accidenté [1].

D'autres systèmes sont réglables et permettent au conducteur de basculer manuellement entre un mode souple, privilégiant le confort et un mode ferme, privilégiant la tenue de route. Certains systèmes offrent également des choix intermédiaires. Les systèmes les plus avancés passent automatiquement du mode souple au mode ferme, et inversement, en fonction du profil de la route et de la vitesse du véhicule. Ces systèmes contribuent également à conserver une hauteur constante et à réduire le roulis. En conséquence, l'automobile possède des caractéristiques offrant un meilleur compromis entre confort et tenue de route que les voitures munies de suspensions ordinaires.

Parmi les systèmes de commande de plus en plus utilisés ces derniers temps, les systèmes basés sur l'intelligence artificielle, grâce au développement des calculateurs électroniques accompagné par la baisse générale des prix de composants électroniques. Dans ce travail on va s'intéresser à la combinaison de deux grands outils : **la logique floue & les réseaux de neurones** ce qui donne La technique: **neurofloue**.

Le choix de cette technique est motivé par deux aspects :

- * La logique floue offre un grand potentiel d'adaptabilité aux systèmes non linéaires (ce qui est le cas ici) et aux perturbations, incertitudes sur les modèles ainsi qu'aux autres facteurs perturbants ; ceci va nous permettre de concevoir une commande robuste vis-à-vis des paramètres variables du système.
- * Les réseaux de neurones artificiels (RNA) permettent d'effectuer un grand volume de calculs sur des fonctions non linéaires ou linéaires d'une façon simple, rapide et précise, de plus leur propriété d'approximation universelle parcimonieuse fait qu'ils deviennent l'outil préféré de la modélisation et de l'estimation paramétrique, enfin leur adaptabilité par apprentissage nous donne l'opportunité de concevoir des commandes robuste et optimales.

Il reste à signaler que le problème de la commande de suspension active a déjà été traité par d'autres méthodes dont :

- Commande par placement de pôles pour contrôler le débattement (oscillations) de la suspension (1997 – Richard & Al).
- Commande par logique floue sur modèles quart et demi véhicule (1995 -Cherry & Jones).
- Commande LQG sur modèle 3 dimensions à 7 degrés de liberté (1996-Elbeheiry & Al).
- Commande non linéaire sur quart de véhicule (2002 –Fialho & Balas)

Notre travail sera décomposé en plusieurs parties :

- Dans la première partie on va s'intéresser à la modélisation mathématique de la suspension, les différents modèles , les composants et les critères de qualité , ainsi que la modélisation non linéaire de l'actionneur hydraulique qui sera employé par la suite pour la commande.
- En suite on va introduire la technique Neurofloue en traitant la logique floue et les RNA à part, puis la combinaison des deux méthodes pour la synthèse des régulateurs neuroflous.
- Dans la troisième partie on va synthétiser des régulateurs de suspension active en se basant sur les résultats des autres parties, en suite on synthétisera le comportement dynamique du système afin de le comparer avec la suspension passive.
- La dernière partie sera réservée aux conclusions & résultats finaux ainsi qu'aux perspectives.

**Chapitre I :
Etude de la suspension
de véhicule**

I.1 Introduction

Initialement les systèmes de suspension étaient conçus pour améliorer la qualité du trajet effectué par le véhicule et pour cela, on employait des éléments mécaniques dissipatifs, amortissants et filtrants afin de réduire le retour de vibration vers la cabine, ces systèmes étaient passifs, leur caractéristiques ne changeaient qu'avec l'usure des composants. Actuellement on conçoit des systèmes modulaires qui s'adaptent aux circonstances du déplacement de la cabine dans les trois directions de l'espace en ajoutant des générateurs d'effort dans la chaîne assurant la suspension, commandés par des régulateurs prenant en compte différents paramètres et critères.

Afin d'élaborer une commande de suspension ; on doit d'abord procéder à son étude, incluant sa constitution, son mode de fonctionnement et sa modélisation mathématique ; les modèles mathématiques obtenus, serviront dans une première étape à simuler son comportement , et plus tard dans la validation des commandes qu'on va élaborer.

I.2 Système de suspension [1]

Afin d'assurer ses différentes tâches ; le système de suspension est constitué de plusieurs composants spécifiques comprenant les bras de positionnement, les ressorts, les amortisseurs, les stabilisateurs et les articulations. Chaque roue possède deux bras de positionnement, situés au-dessus et au-dessous de l'axe de roue (sur lequel tourne la roue). Les bras de positionnement contrôlent l'attitude de la roue lorsque celle-ci se déplace sur une surface inégale ou le long d'un virage. La longueur et la plage de débattement des bras de positionnement déterminent la longueur et les angles de débattement de la roue. Par exemple, un système avec un bras supérieur plus court que le bras inférieur aura tendance à tirer légèrement le sommet de la roue vers l'intérieur quand la roue s'enfonce.

Les ressorts amortissent le choc quand une roue passe sur une bosse. Le nombre et l'emplacement des ressorts dépendent du système de suspension mais, en général, on trouve quatre ressorts hélicoïdaux, situés à proximité de chacune des roues. Quand une roue rencontre une bosse, elle est poussée vers le haut, ce qui comprime le ressort. Le ressort résiste au mouvement de la roue puis se détend, repoussant la roue vers le bas. La plupart des ressorts de voiture sont hélicoïdaux, fabriqués en acier. Les pneus se comportent également comme des ressorts en caoutchouc remplis d'air.

On trouve un amortisseur à chaque roue. Les amortisseurs inhibent la tendance naturelle des ressorts à continuer à osciller une fois qu'ils ont été déformés, puis relâchés. Sans les amortisseurs, une voiture ne cesserait pas de s'agiter, de tanguer et de se balancer, perturbant le confort de roulement et interférant avec le comportement du véhicule. L'amortisseur le plus courant est constitué d'un piston monté sans jeu dans un cylindre plein d'huile. Le piston monte et descend à l'intérieur du cylindre quand les roues passent sur les inégalités de la route. Le cylindre, qui est fixé à la caisse, demeure immobile. Quand la roue passe sur une bosse, elle déplace le piston à l'intérieur du cylindre. Des clapets laissent passer l'huile d'un côté à l'autre du piston, régulant la vitesse à laquelle le piston peut se déplacer et le degré d'amortissement fourni. De petits clapets inhibent pratiquement toute oscillation et procurent le roulement ferme et le comportement agile caractéristiques des voitures de sport. De grands clapets permettent au piston de se déplacer relativement librement et procurent un roulement plus doux.

Les stabilisateurs apportent un supplément de précision à la suspension. À la différence des bras de positionnement, qui sont normalement rigides, les stabilisateurs présentent une certaine flexibilité, comme des ressorts. Un type de stabilisateur, la barre antiroulis, relie les deux roues avant ou les deux roues arrière. La barre antiroulis inhibe la tendance de la caisse à s'incliner dans les virages. Un autre type de stabilisateur, la barre (bielle) de contrôle du pincement, améliore la direction en s'assurant que les roues avant et arrière demeurent pointées dans la bonne direction au cours des débattements de la suspension.

Les stabilisateurs demandent une certaine flexibilité au niveau des articulations de suspension. On peut assurer cette flexibilité en utilisant des articulations élastiques possédant une garniture en caoutchouc (constituées d'un bloc de caoutchouc enserré entre la partie fixe et la partie mobile de l'articulation). Les articulations élastiques agissent également comme des ressorts et des amortisseurs miniatures qui absorbent les vibrations.

Le poids d'une automobile est rarement réparti de façon égale entre l'avant et l'arrière. Le poids — et, par suite, les exigences de la suspension — est généralement plus grand à l'avant, où se trouve le moteur, qu'à l'arrière. Bien que quelques automobiles soient encore équipées d'une suspension arrière non indépendante (dans laquelle les deux roues arrière sont

reliées par un essieu rigide, qui les empêche de se mouvoir indépendamment l'une de l'autre), toutes les voitures modernes possèdent une suspension avant à roues indépendantes.

La suspension avant, joue un rôle plus critique dans le contrôle de la direction de la voiture. Les deux systèmes doivent faire pointer les roues de façon que la voiture se déplace dans la direction voulue et que les pneus demeurent perpendiculaires à la route, sans s'user anormalement.

Les axes d'articulation des bras (des triangles) avants présentent une disposition anti-plongée qui s'oppose à la tendance à piquer du nez sous l'effet du transfert de poids vers l'avant qui se produit au freinage. Les axes d'articulation arrière présentent une disposition anti-accroupissement qui compense la tendance à lever le nez sous l'effet du transfert de poids vers l'arrière qui se produit à l'accélération.

Il existe plusieurs conceptions de suspension avant et arrière, notamment les types à double triangle et les types à jambe de force Mac Pherson (tous deux avec des ressorts hélicoïdaux), les ressorts à barre de torsion, et les ressorts à lame(s). Le système de suspension à double triangle est plus coûteux que les autres ; on le trouve plus fréquemment sur les voitures de sport qui, normalement, possèdent des suspensions complètement indépendantes à l'avant et à l'arrière. Le double triangle est plus compact ; il permet d'abaisser le capot, au bénéfice de la visibilité et de l'aérodynamique. Le système à jambe de force Mac Pherson, un ressort hélicoïdal entourant un amortisseur qui sert de bras de positionnement supérieur, est moins coûteux que le double triangle ; on le trouve sur la plupart des voitures modernes à traction avant. Il est plus encombrant en hauteur que le double triangle, et demande un capot plus haut. Les ressorts à lame(s) sont caractéristiques des suspensions arrière à essieu rigide et des véhicules utilitaires ou de loisir.

Tous ces éléments fonctionnent afin d'assurer les tâches suivantes :

- Absorption des différents chocs
- Elimination des vibrations
- support du poids du véhicule et de sa charge
- tenue de route (contact permanent avec la chaussée)
- précision de la direction et puissance du freinage
- empêcher le roulis, tangage et lacet (mouvements indésirables), (voir figure)

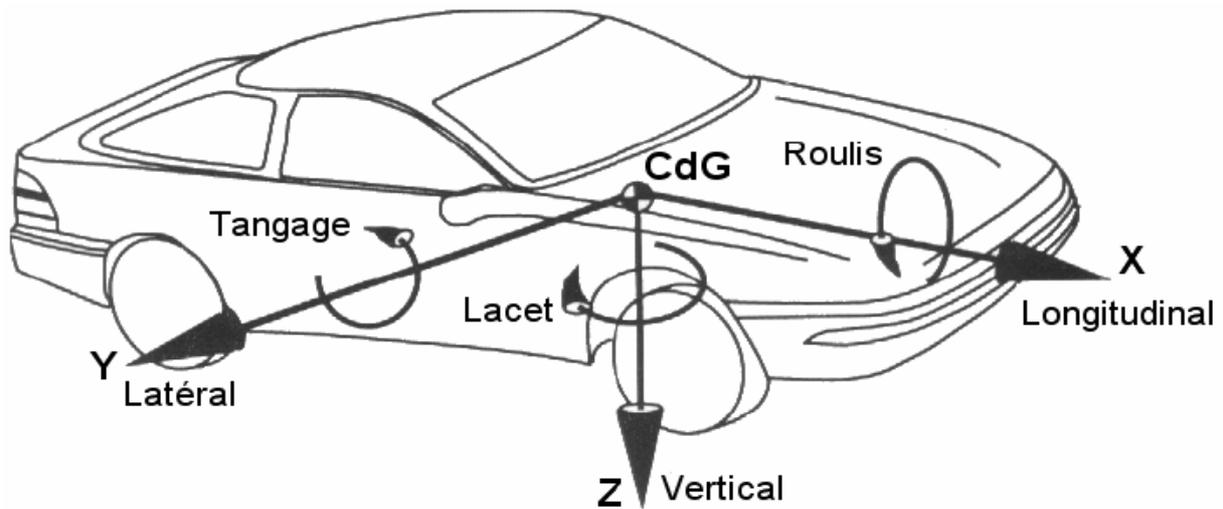


Figure 1.1 : mouvements de la caisse dans l'espace [6]

I.3 Fonctionnement de la suspension [6]

Au cours d'un parcours, si une roue rencontre une dénivellation lorsqu'elle quitte sa trajectoire horizontale, elle est projetée vers le bas par une force égale à son poids suspendu par la tension du ressort. Au cours de la chute, la tension du ressort diminue, ce qui provoque l'abaissement de la caisse (le corps rigide) de la voiture et ceci va comprimer le ressort de nouveau, quand la roue touche le sol au fond de la dénivellation, la position d'équilibre sera dépassée et la tension du ressort supérieure au poids de la caisse va rejeter celle-ci vers le haut.

Ceci produit une oscillation amortie de l'ensemble qui finit par s'atténuer.

Par contre, si la roue rencontre un obstacle en saillie, le ressort sera d'abord comprimé, et la caisse -projetée vers le haut- dépassera la position d'équilibre et, comme dans précédemment, ceci produira une série d'oscillations. Si un deuxième obstacle se présente avant que l'ensemble ait repris sa position d'équilibre, il en résultera une nouvelle action qui, suivant le cas, pourra augmenter ou diminuer l'amplitude des oscillations.

Lorsqu'une roue passe sur un obstacle, il en résulte un mouvement brusque ascendant, qui détermine une énergie assimilée par une force vive de valeur $\frac{1}{2}mv^2$, où m est la masse du train roulant et v sa vitesse verticale.

La plus grande partie de cette force vive s'annule par le mouvement ascendant consécutif imprimé à la caisse. Le mouvement de celle-ci engendre lors une force vive de valeur $\frac{1}{2}m'v'^2$, dans laquelle m' est la masse des organes suspendue et v' leur vitesse verticale, vitesse qu'il est nécessaire de réduire au minimum pour obtenir une suspension convenable. La valeur $\frac{1}{2}m'v'^2$ est approximativement égale à la force vive du train roulant.

Pour assurer une suspension confortable, il y a une fréquence maximale des oscillations à ne pas dépasser. Il est facile d'adapter cette fréquence sur une suspension ayant des qualités permanentes, mais ce n'est pas le cas en pratique et les charges peuvent varier considérablement d'un moment à l'autre pour un même véhicule, en plus l'usure agit d'une façon sensible sur les propriétés des organes de suspension, il est donc impossible d'avoir une suspension parfaite, car elle sera molle ou dure suivant qu'elle sera chargée ou vide.

I.4 Les différents types de suspension [2], [3], [10]

Les suspensions peuvent être classées selon l'apport externe de force ou bien d'énergie.

Vis-à-vis de la commande on peut distinguer trois types de suspensions: suspension passive, suspension semi-active et suspension active.

I.4.1 Suspension passive

Les systèmes de suspension passive sont des systèmes de suspension conventionnels qui comprennent des éléments passifs : ressorts, amortisseurs, pneus flexibles.

Une suspension passive est composée d'un ensemble d'éléments dissipatifs et absorbants, et est caractérisée par l'absence d'éléments apportant force ou énergie au système.

Les caractéristiques de cette suspension sont figées; c'est pour cela que la réalisation d'une suspension passive à paramètres et performances optimales pour couvrir les différentes conditions et intégrer les différents éléments environnementaux de la route est une tâche très difficile pour ne pas dire impossible, car elle se heurte au fameux compromis confort/sécurité qu'on va citer par la suite.

Le ressort est défini par sa constante de raideur Force/Position. Il fournit une force qui dépend de la position ou bien de l'écart de position entre la caisse et la roue.

La force développée par l'amortisseur dépend quant à elle du rapport Force/vitesse appelé aussi le coefficient d'amortissement.

L'amortisseur est caractérisé par la relation suivante:

$$u = c(v_{roue} - v_{caisse}) \quad (I.1)$$

Où C : le coefficient d'amortissement supposé constant.

I.4.2 Suspension semi-active

Ce type de suspension ne contient pas de source de force (actionneur). Cependant, en influant et en agissant sur les caractéristiques des composants de la suspension en fonction du mouvement de la caisse, la suspension semi-active opère la commande. Souvent, les amortisseurs ont deux taux ou plus, électriquement sélectionnables et qui peuvent donc durcir ou bien assouplir le comportement du système de suspension.

Autrement dit, dans ces systèmes de suspension, la commande recherchée est atteinte à travers la variation des coefficients d'amortissement de l'amortisseur. On peut dire que le besoin d'équipements est du coup moins important que celui exigé dans la suspension active dont il sera question plus loin dans notre présent travail.

La suspension semi-active ne s'accompagne pas d'apport d'énergie car son amortisseur semi-actif, tout comme d'ailleurs l'amortisseur passif (dans le cas d'une suspension passive), ne peut que dissiper l'énergie en produisant une force qui va s'opposer à la vitesse relative entre la caisse et la roue. Cette force est cependant modulable, contrairement à un amortisseur passif, d'où le qualificatif attribué de "semi-active" à cette suspension.

Ce type de suspension est à mi-chemin entre les systèmes passifs et actifs. Il est similaire au système passif puisque tous ses éléments génèrent respectivement leur propre force ; toutefois on considère que l'amortissement peut être changé instantanément. Le choix de ce système est motivé par le désir d'améliorer partiellement les performances des suspensions passives.

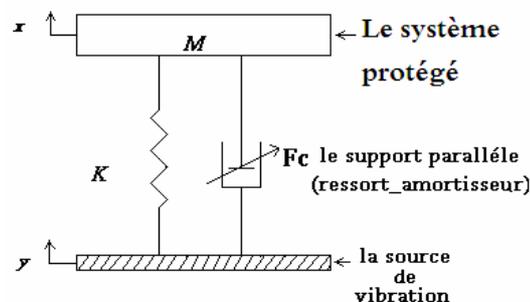


Figure 1.2 : Modélisation d'une suspension semi active [6]

Selon le type de la suspension semi active utilisée, et pour un amortisseur électromécanique, la force commandée F_C est en générale fonction de plusieurs signaux variants dans le temps. Ces signaux sont normalement obtenus par des capteurs de déplacement, de vitesse et d'accélération fixés sur la masse vibrante et parfois sur les amortisseurs pour les capteurs de force.

I.4.3 Suspension active

Les réglages effectués par les suspensions passive et semi-active dépendent des positions et des vitesses relatives entre la caisse et la roue. Ces suspensions n'effectuent leurs réglages que par la dissipation de l'énergie, comme nous venons de le voir plus haut.

Par contre, les suspensions actives sont des systèmes qui permettent de dépasser ces limitations.

Dans ces systèmes, les ressorts et les amortisseurs sont remplacés partiellement ou totalement par les actionneurs. La force développée par les actionneurs est produite suivant une loi de commande.

La suspension active offre des avantages par rapport à la suspension semi-active bien qu'elle contienne aussi plusieurs contraintes dont il faut tenir compte. Parmi ces contraintes, on peut notamment citer le coût et la consommation d'énergie.

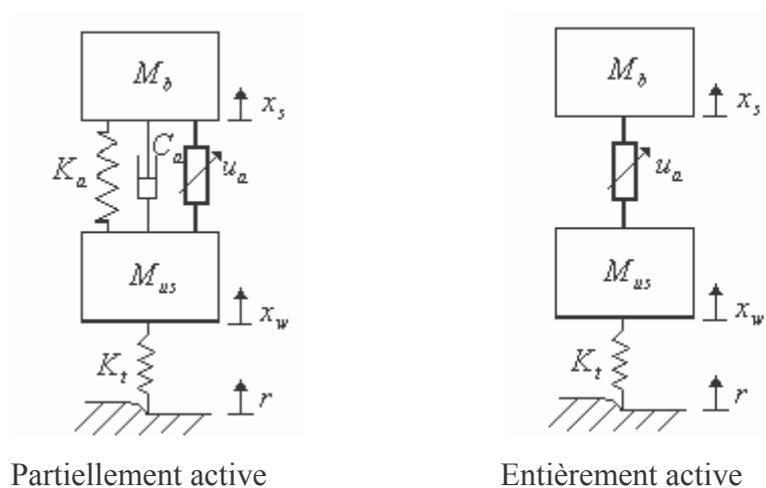


Figure I.3 : Suspensions actives [2]

I.5 Modélisation de la suspension [2]

I.5.1 Modélisation linéaire

Pour obtenir un modèle linéaire, on ne prend pas en compte la dynamique non linéaire des actionneurs hydrauliques utilisés généralement dans la commande de suspension, on emploie aussi des approximations relatives aux angles de tangage et roulis considérés petits.

I.5.1.1 Modèle quart de véhicule ou mono roue

Le modèle quart de véhicule, prend en compte le mouvement vertical de la caisse et celui de la roue. Il est modélisé par une masse suspendue, représentant le quart de la caisse, relié à une masse non suspendue, représentant la roue et une partie de l'essieu.

Les variables $m_s, m_{us}, k_s, k_{us}, C_s$ et C_{us} représentent, respectivement, la masse, la raideur et l'amortissement, correspondant aux masses suspendue et non suspendue. Les variables X_c, X_w et X_r sont les déplacements de la caisse, de la roue et le profil de la route considéré comme perturbation.

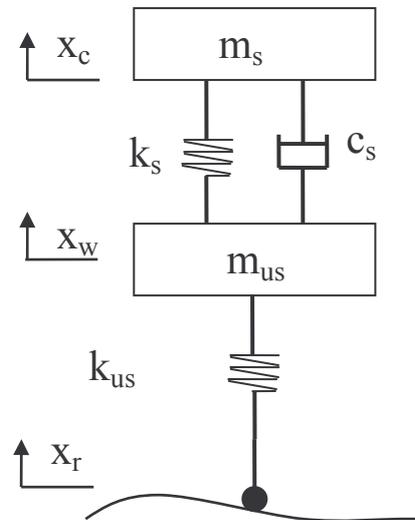


Figure 1.4 : *Modèle quart de véhicule* [3]

La dynamique linéaire de ce modèle est décrite par les équations suivantes :

$$m_s \cdot \ddot{X}_c = k_s \cdot (X_w - X_c) + c_s \cdot (\dot{X}_w - \dot{X}_c) \quad (I.2)$$

$$m_{us} \cdot \ddot{X}_w = k_{us} \cdot (X_r - X_w) - k_s \cdot (X_w - X_c) - c_s \cdot (\dot{X}_w - \dot{X}_c) \quad (I.3)$$

La représentation d'état associée aux équations précédentes est donc:

$$\dot{X} = A.X + b_d d \tag{I.4}$$

Où $X = [X_c, \dot{X}_c, X_w, \dot{X}_w]^T$

$d = X_r$ est le profil de la route. Les matrices d'état sont alors données par :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_s} & -\frac{c_c}{m_s} & \frac{k_s}{m_s} & \frac{c_s}{m_s} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_{us}} & \frac{c_s}{m_{us}} & -\frac{k_s + k_{us}}{m_{us}} & -\frac{c_s}{m_{us}} \end{bmatrix}, b_d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_{us}}{m_{us}} \end{bmatrix}$$

I.5.1.2 Modèle demi véhicule :

Deux modèles peuvent être définis, selon les mouvements pris en compte ;

- * Le modèle demi véhicule de type essieu, qui permet de prendre en compte, en plus des mouvements verticaux, les mouvements de roulis de la caisse et de l'essieu (dans le cas d'un essieu mono bloc).
- * Le deuxième type de modèles est le modèle demi véhicule de type bicyclette, qui est utilisé pour tenir compte, en plus des mouvements verticaux, du mouvement de tangage de la caisse du véhicule. C'est ce type de modèle qu'on va présenter ci-dessous.

Les mouvement linéaires (verticaux) et angulaires (tangage) de la caisse, et des roues peuvent être étudiés, en utilisant le modèle demi véhicule de type bicyclette de la figure (I.5).

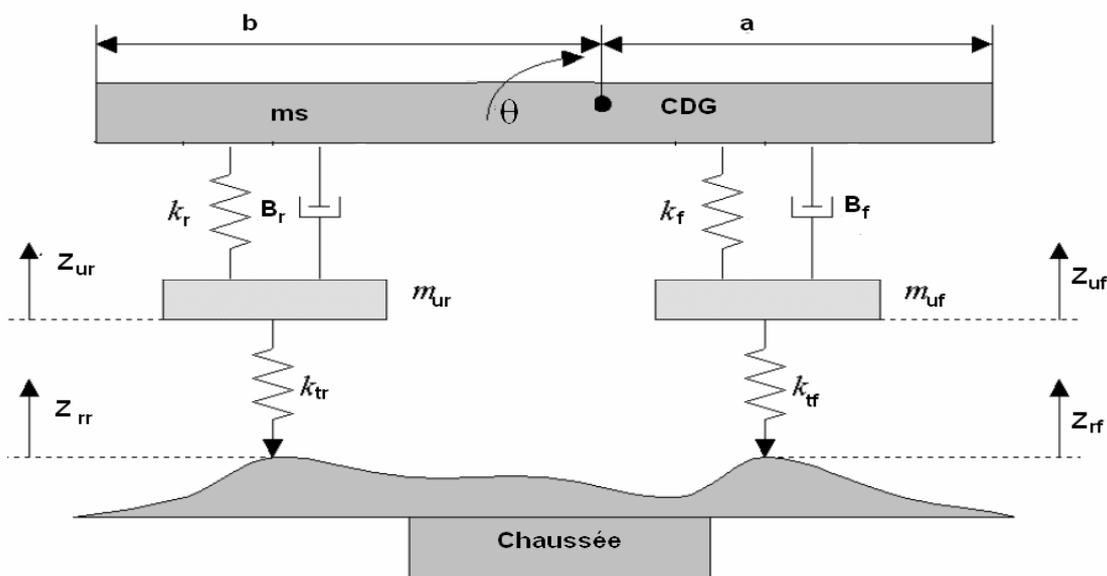


Figure 1.5 : Modèle demi véhicule du type bicyclette [8]

Pour simplifier les calculs, on considère que les angles de tangage sont assez petits ce qui permet d'obtenir un modèle linéaire. La masse de la caisse est m_s , son moment d'inertie est J_y . Les masses non suspendues avant et arrière sont respectivement, m_{uf} et m_{ur} , Z_{rr} et Z_{rf} sont les profils de la route (perturbations affectant les roues arrière et avant).

Les équations décrivant le comportement du système sont :

$$\text{Partie avant ; } Z_{sf} = Z - a.\sin(\theta) \approx Z - a.\theta \quad (\text{I.5})$$

$$\text{Partie arrière ; } Z_{sr} = Z + b.\sin(\theta) \approx Z + b.\theta \quad (\text{I.6})$$

Les forces équivalentes des deux cotés sont les suivantes ;

Partie avant ;

$$F_f = -k_f.(Z_{sf} - Z_{uf}) - B_f.(\dot{Z}_{sf} - \dot{Z}_{uf}) = -k_f.(Z - a.\theta - Z_{uf}) - B_f.(\dot{Z} - a\dot{\theta} - \dot{Z}_{uf}) \quad (\text{I.7})$$

Partie arrière ;

$$F_r = -k_r.(Z_{sr} - Z_{ur}) - B_r.(\dot{Z}_{sr} - \dot{Z}_{ur}) = -k_r.(Z + b.\theta - Z_{ur}) - B_r.(\dot{Z} + b\dot{\theta} - \dot{Z}_{ur}) \quad (\text{I.8})$$

En appliquant la deuxième lois de Newton, et en considérant la position d'équilibre statique, comme origine des déplacements linéaires et angulaires du centre de gravité, on aura ;

$$\ddot{Z} = \frac{1}{m_s} \left\{ \begin{array}{l} -(k_f + k_r).Z - (B_f - B_r).\dot{Z} + (a.k_f - b.k_r).\theta + (a.B_f - b.B_r).\dot{\theta} \\ +k_f.Z_{uf} + k_r.Z_{ur} + B_f.\dot{Z}_{uf} + B_r.\dot{Z}_{ur} \end{array} \right\} \quad (\text{I.9})$$

Pour le mouvement angulaire, on aura :

$$\begin{aligned} J_y.\ddot{\theta} &= -F_f.a.\cos(\theta) + F_r.b.\cos(\theta) = -F_f.a + F_r.b. \\ &= \left(\begin{array}{l} a.k_f.(Z - a.\theta - Z_{uf}) + a.B_f.(\dot{Z} - a.\dot{\theta} - \dot{Z}_{uf}) - b.k_r.(Z + b.\theta - Z_{ur}) \\ -b.B_r.(\dot{Z} + b.\dot{\theta} - \dot{Z}_{ur}) \end{array} \right) \end{aligned} \quad (\text{I.10})$$

Posons $J_y = m_s.r_y^2$ ou r_y est le rayon de giration.

$$\ddot{\theta} = \frac{1}{m_s.r_y^2} \left(\begin{array}{l} (a.k_f - b.k_r).Z + (a.B_f - b.B_r).\dot{Z} - (a^2.k_f + b^2.k_r).\theta - (a^2.B_f + b^2.B_r).\dot{\theta} \\ -a.k_f.Z_{uf} + b.k_r.Z_{ur} - a.B_f.\dot{Z}_{uf} + b.B_r.\dot{Z}_{ur} \end{array} \right) \quad (\text{I.11})$$

En appliquant la deuxième loi de Newton aux masses non suspendues :

Roue avant ;

$$\begin{aligned}
 m_{uf} \ddot{Z}_{uf} &= B_f \cdot (\dot{Z}_{sf} - \dot{Z}_{uf}) + k_f (Z_{sf} - Z_{uf}) - k_{tf} (Z_{uf} - Z_{rf}) \\
 &= B_f (\dot{Z} - a \dot{\theta} - \dot{Z}_{uf}) + k_f (Z - a \theta - Z_{uf}) - k_{tf} (Z_{uf} - Z_{rf}) \\
 &= B_f \dot{Z} - a B_f \dot{\theta} - B_f \dot{Z}_{uf} + k_f Z - a k_f \theta - (k_f + k_{tf}) Z_{uf} + k_{tf} Z_{rf}
 \end{aligned} \tag{I.12}$$

Roue arrière ;

$$\begin{aligned}
 m_{ur} \ddot{Z}_{ur} &= B_r \cdot (\dot{Z}_{sr} - \dot{Z}_{ur}) + k_r (Z_{sr} - Z_{ur}) - k_{tr} (Z_{ur} - Z_{rr}) \\
 &= B_r (\dot{Z} + b \dot{\theta} - \dot{Z}_{ur}) + k_r (Z + b \theta - Z_{ur}) - k_{tr} (Z_{ur} - Z_{rr}) \\
 &= B_r \dot{Z} + b B_r \dot{\theta} - B_r \dot{Z}_{ur} + k_r Z + b k_r \theta - (k_r + k_{tr}) Z_{ur} + k_{tr} Z_{rr}
 \end{aligned} \tag{I.13}$$

On définit l'espace d'état comme suit ;

$$\begin{aligned}
 X_1 &= Z. & X_2 &= \dot{Z}. & X_3 &= \theta. & X_4 &= \dot{\theta}. \\
 X_5 &= Z_{sf} - Z_{uf}. & X_6 &= \dot{Z}_{uf}. & X_7 &= Z_{sr} - Z_{ur}. & X_8 &= \dot{Z}_{ur}.
 \end{aligned}$$

On obtient donc la représentation d'état associée aux équations précédentes ;

$$\begin{aligned}
 \dot{X}_1 &= X_2. \\
 \dot{X}_2 &= -\frac{k_f + k_r}{m_s} X_1 - \frac{B_f + B_r}{m_s} X_2 + \frac{aB_f - bB_r}{m_s} X_4 - \frac{k_f}{m_s} X_5 + \frac{B_f}{m_s} X_6 - \frac{k_r}{m_s} X_7 + \frac{B_r}{m_s} X_8 \\
 \dot{X}_3 &= X_4. \\
 \dot{X}_4 &= \frac{aB_f - bB_r}{m_s r_y^2} X_2 - \frac{a^2 B_f + b^2 B_r}{m_s r_y^2} X_4 + \frac{aK_f}{m_s r_y^2} X_5 - \frac{aB_f}{m_s r_y^2} X_6 - \frac{bk_r}{m_s r_y^2} X_7 + \frac{bB_r}{m_s r_y^2} X_8 \\
 \dot{X}_5 &= X_2 - a X_4 - X_6 \\
 \dot{X}_6 &= -\frac{k_{tf}}{m_{uf}} X_1 + \frac{B_f}{m_{uf}} X_2 + \frac{a k_{tf}}{m_{uf}} X_3 - \frac{a B_f}{m_{uf}} X_4 + \frac{k_f + k_{tf}}{m_{uf}} X_5 - \frac{B_f}{m_{uf}} X_6 + \frac{k_{tf}}{m_{uf}} Z_{rf} \\
 \dot{X}_7 &= X_2 + b X_4 - X_8. \\
 \dot{X}_8 &= -\frac{k_{tr}}{m_{ur}} X_1 + \frac{B_r}{m_{ur}} X_2 + \frac{b k_{tr}}{m_{ur}} X_3 + \frac{b B_r}{m_{ur}} X_4 + \frac{k_r + k_{tr}}{m_{ur}} X_7 - \frac{B_r}{m_{ur}} X_8 + \frac{k_{tr}}{m_{ur}} Z_{rr}.
 \end{aligned}$$

On obtient ; $\dot{X} = AX + B_d \cdot d$ Avec, $d = [Z_{rf} \quad Z_{rr}]$ est le profil de la route

I.5.1.3 Modèle véhicule complet [5]

Le modèle complet du système de suspension est montré dans la figure suivante. Il est représenté comme un système linéarisé de 7 degré de liberté (7 DDL). Il se compose d'une masse suspendue (châssis) reliée à quatre roues (avant-gauche, avant-droite, arrière-gauche et arrière-droite). La masse suspendue est libre de se mouvoir autour des 3 axes tandis que les masses non

suspendue (roues) ne peuvent que se déplacer verticalement et pour simplifier les calculs, on assume que des angles sont petits et ceci nous permet d'obtenir un modèle linéaire.

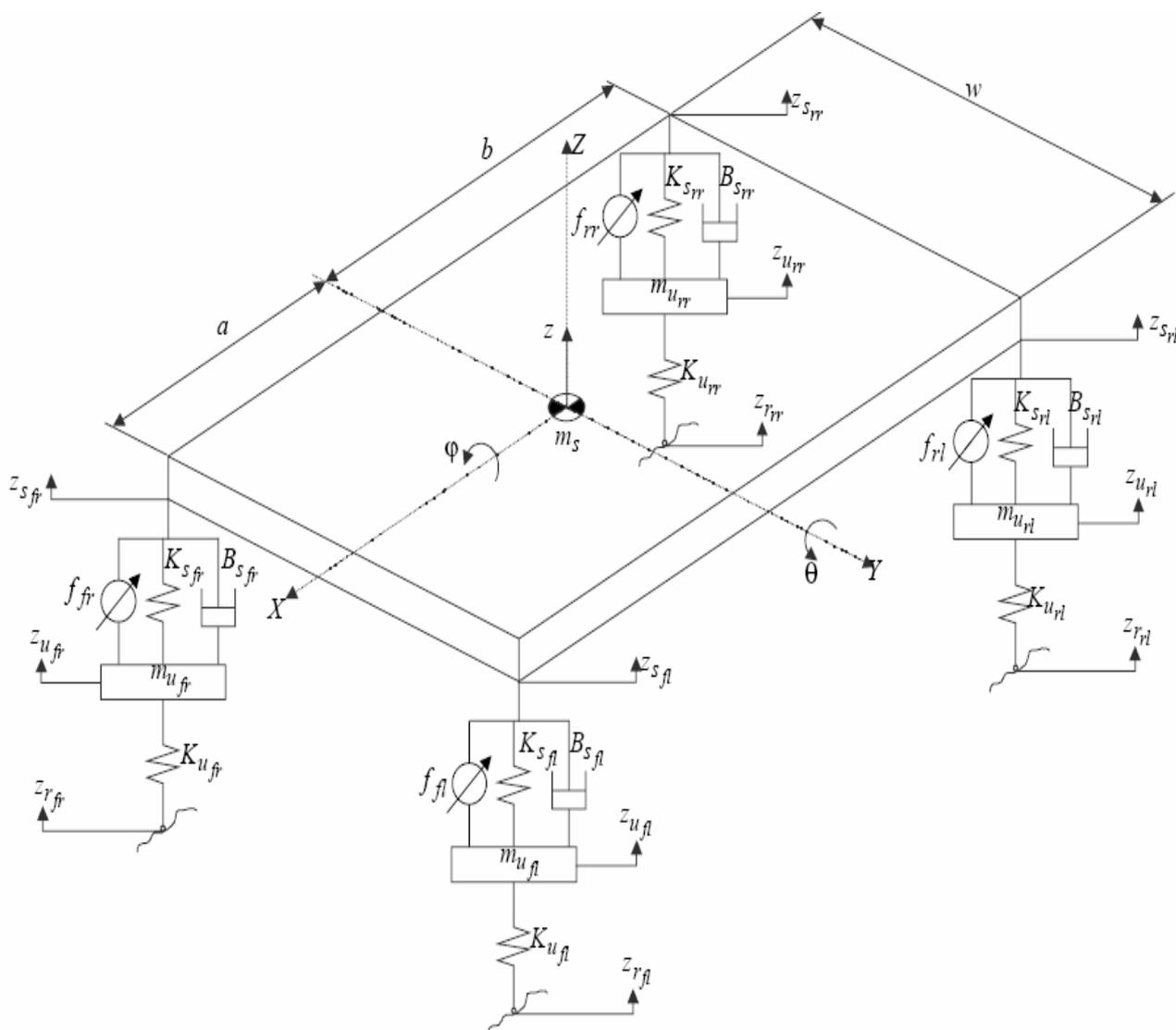


Figure 1.6 : modèle complet à 7ddl [5]

Les équations mécaniques décrivant le système [5] :

$$m_s \ddot{Z} = -(2K_{sf} + 2K_{sr})Z - (2B_{sf} + 2B_{sr})\dot{Z} + (2aK_{sf} - 2bK_{sr})\theta + (2aB_{sf} + 2bB_{sr})\dot{\theta} + K_{sf}Z_{ufl} + B_{sf}\dot{Z}_{ufl} + K_{sf}Z_{ufr} + B_{sf}\dot{Z}_{ufr} + K_{sr}Z_{url} + B_{sr}\dot{Z}_{url} + K_{sr}Z_{urr} + B_{sr}\dot{Z}_{urr} \tag{I.14}$$

$$I_{yy} \ddot{\theta} = (2aK_{sf} - 2bK_{sr})Z + (2aB_{sf} - 2bB_{sr})\dot{Z} - (2a^2K_{sf} + 2b^2K_{sr})\theta - (2a^2B_{sf} + 2b^2B_{sr})\dot{\theta} - aK_{sf}Z_{ufl} - aB_{sf}\dot{Z}_{ufl} - aK_{sf}Z_{ufr} - aB_{sf}\dot{Z}_{ufr} + bK_{sr}Z_{url} + bB_{sr}\dot{Z}_{url} + bK_{sr}Z_{urr} + bB_{sr}\dot{Z}_{urr} \tag{I.15}$$

$$I_{xx} \ddot{\phi} = -0.25w^2(2K_{sf} + 2K_{sr})\phi - 0.25w^2(2B_{sf} + 2B_{sr})\dot{\phi} + 0.5wK_{sf}Z_{ufl} + 0.5wB_{sf}\dot{Z}_{ufl} - 0.5wK_{sf}Z_{ufr} - 0.5wB_{sf}\dot{Z}_{ufr} + 0.5wK_{sr}Z_{url} + 0.5wB_{sr}\dot{Z}_{url} - 0.5wK_{sr}Z_{urr} - 0.5wB_{sr}\dot{Z}_{urr} \quad (I.16)$$

$$m_u \ddot{Z}_{ufl} = K_{sf}Z + B_{sf}\dot{Z} - aK_{sf}\theta - aB_{sf}\dot{\theta} + 0.5wK_{sf}\phi + 0.5wB_{sf}\dot{\phi} - (K_{sf} + K_u)Z_{ufl} - B_{sf}\dot{Z}_{ufl} + K_u Z_{rfl} \quad (I.17)$$

$$m_u \ddot{Z}_{ufr} = K_{sf}Z + B_{sf}\dot{Z} - aK_{sf}\theta - aB_{sf}\dot{\theta} - 0.5wK_{sf}\phi - 0.5wB_{sf}\dot{\phi} - (K_{sf} + K_u)Z_{ufr} - B_{sf}\dot{Z}_{ufr} + K_u Z_{rflr} \quad (I.18)$$

$$m_u \ddot{Z}_{url} = K_{sr}Z + B_{sr}\dot{Z} + bK_{sr}\theta + bB_{sr}\dot{\theta} + 0.5wK_{sr}\phi + 0.5wB_{sr}\dot{\phi} - (K_{sr} + K_u)Z_{url} - B_{sr}\dot{Z}_{url} + K_u Z_{rrl} \quad (I.19)$$

$$m_u \ddot{Z}_{urr} = K_{sr}Z + B_{sr}\dot{Z} + bK_{sr}\theta + bB_{sr}\dot{\theta} - 0.5wK_{sr}\phi - 0.5wB_{sr}\dot{\phi} - (K_{sr} + K_u)Z_{urr} - B_{sr}\dot{Z}_{urr} + K_u Z_{rrr} \quad (I.20)$$

Soient les variables d'état suivantes

$x_1 = z$: déplacement vertical du centre de masse

$x_2 = \dot{z}$: vitesse du centre de masse

$x_3 = \theta$: angle de tangage

$x_4 = \dot{\theta}$

$x_5 = \phi$: angle de roulis

$x_6 = \dot{\phi}$

$x_7 = Z_{rfl}$: déplacement roue avant gauche

$x_8 = \dot{Z}_{rfl}$

$x_9 = Z_{rfr}$: déplacement roue avant droite

$x_{10} = \dot{Z}_{rfr}$

$x_{11} = Z_{rrl}$: déplacement roue arrière gauche

$x_{12} = \dot{Z}_{rrl}$

$x_{13} = Z_{rrr}$: déplacement roue arrière droite

$x_{14} = \dot{Z}_{rrr}$

Les équations d'état décrivant le système sont:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{(2k_{sf} + 2k_{sr})}{m_s} x_1 - \frac{(2B_{sf} + 2B_{sr})}{m_s} x_2 + \frac{(2ak_{sf} - 2bk_{sr})}{m_s} x_3 + \frac{(2aB_{sf} - 2bB_{sr})}{m_s} x_4 + \frac{k_{sf}}{m_s} x_7 + \frac{B_{sf}}{m_s} x_8 + \frac{k_{sf}}{m_s} x_9 + \frac{B_{sf}}{m_s} x_{10} + \frac{k_{sr}}{m_s} x_{11} + \frac{B_{sr}}{m_s} x_{12} + \frac{k_{sr}}{m_s} x_{13} + \frac{B_{sr}}{m_s} x_{14}$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{(2ak_{sf} - 2bk_{sr})}{I_{yy}} x_1 + \frac{(2aB_{sf} - 2bB_{sr})}{I_{yy}} x_2 - \frac{(2a^2k_{sf} + 2b^2k_{sr})}{I_{yy}} x_3 - \frac{(2a^2B_{sf} + 2b^2B_{sr})}{I_{yy}} x_4 - \frac{ak_{sf}}{I_{yy}} x_7 - \frac{aB_{sf}}{I_{yy}} x_8 - \frac{ak_{sf}}{I_{yy}} x_9 - \frac{aB_{sf}}{I_{yy}} x_{10} + \frac{bk_{sr}}{I_{yy}} x_{11} + \frac{bB_{sr}}{I_{yy}} x_{12} + \frac{bk_{sr}}{I_{yy}} x_{13} + \frac{bB_{sr}}{I_{yy}} x_{14}$$

$$\dot{x}_5 = x_6$$

$$\dot{x}_6 = -\frac{w^2(2k_{sf} + 2k_{sr})}{4I_{xx}} x_5 - \frac{w^2(2B_{sf} + 2B_{sr})}{4I_{xx}} x_6 + \frac{wk_{sf}}{2I_{xx}} x_7 + \frac{wB_{sf}}{2I_{xx}} x_8 - \frac{wk_{sf}}{2I_{xx}} x_9 - \frac{wB_{sf}}{2I_{xx}} x_{10} + \frac{wk_{sr}}{2I_{xx}} x_{11} + \frac{wB_{sr}}{2I_{xx}} x_{12} - \frac{wk_{sr}}{2I_{xx}} x_{13} - \frac{wB_{sr}}{2I_{xx}} x_{14}$$

$$\dot{x}_7 = x_8$$

$$\dot{x}_8 = \frac{k_{sf}}{m_u} x_1 + \frac{B_{sf}}{m_u} x_2 - \frac{ak_{sf}}{m_u} x_3 - \frac{aB_{sf}}{m_u} x_4 + \frac{wk_{sf}}{2m_u} x_5 + \frac{wB_{sf}}{2m_u} x_6 - \frac{(k_{sf} + k_u)}{m_u} x_7 - \frac{B_{sf}}{m_u} x_8 + \frac{k_u}{m_u} z_{rfl}$$

$$\dot{x}_9 = x_{10}$$

$$\dot{x}_{10} = \frac{k_{sf}}{m_u} x_1 + \frac{B_{sf}}{m_u} x_2 - \frac{ak_{sf}}{m_u} x_3 - \frac{aB_{sf}}{m_u} x_4 - \frac{wk_{sf}}{2m_u} x_5 - \frac{wB_{sf}}{2m_u} x_6 - \frac{(k_{sf} + k_u)}{m_u} x_9 - \frac{B_{sf}}{m_u} x_{10} + \frac{k_u}{m_u} z_{rfl}$$

$$\dot{x}_{11} = x_{12}$$

$$\dot{x}_{12} = \frac{k_{sr}}{m_u} x_1 + \frac{B_{sr}}{m_u} x_2 + \frac{bk_{sr}}{m_u} x_3 + \frac{bB_{sr}}{m_u} x_4 + \frac{wk_{sr}}{2m_u} x_5 + \frac{wB_{sr}}{2m_u} x_6 - \frac{(k_{sr} + k_u)}{m_u} x_{11} - \frac{B_{sr}}{m_u} x_{12} + \frac{k_u}{m_u} z_{rrl}$$

$$\dot{x}_{13} = x_{14}$$

$$\dot{x}_{14} = \frac{k_{sr}}{m_u} x_1 + \frac{B_{sr}}{m_u} x_2 + \frac{bk_{sr}}{m_u} x_3 + \frac{bB_{sr}}{m_u} x_4 - \frac{wk_{sr}}{2m_u} x_5 - \frac{wB_{sr}}{2m_u} x_6 - \frac{(k_{sr} + k_u)}{m_u} x_{13} - \frac{B_{sr}}{m_u} x_{14} + \frac{k_u}{m_u} z_{rrr}$$

I.5.2 Modélisation non linéaire :

Pour obtenir un modèle non linéaire, on considère les angles de tangage et roulis, et on prend en compte la dynamique non linéaire des actionneurs hydrauliques illustrée dans le paragraphe suivant.

I.5.2.1 Dynamique de l'actionneur [9]:

Les actionneurs ont un rôle appréciable à jouer dans la conception des systèmes de suspension active dans la mesure où ils sont responsables de la fourniture de la force de commande désirée dans le système. Afin d'atteindre les différents objectifs que l'on se fixe, l'actionneur doit appliquer la force de commande d'une manière précise et au moment approprié et adéquat. Les actionneurs électro-hydrauliques se caractérisent par leur capacité à fournir de très grands efforts tout en étant assez rapides. La caractéristique non linéaire de l'actionneur est due à plusieurs facteurs tels que la relation non linéaire entre le débit et la pression, la compressibilité du fluide, la friction, etc ...

Nous allons à présent et dans ce qui suit, nous intéresser à la dynamique de l'actionneur employé ensuite pour établir la commande.

L'actionneur hydraulique utilisé ici est une électrovanne à quatre voies.

Soit u_a la commande des pistons hydrauliques, elle est donnée par la relation:

$$u_a = AP_L \quad (\text{I.21})$$

Où A est la surface du piston et P_L est la chute de pression à travers le cylindre (gradient de pression).

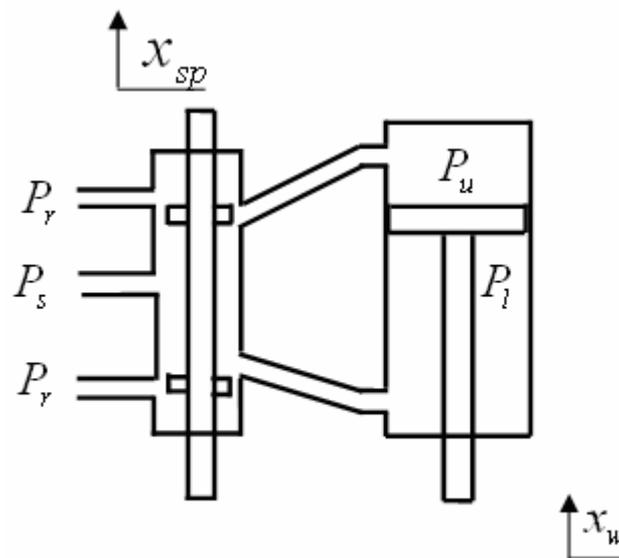


Figure 1.7 : actionneur hydraulique [9]

La dynamique de P_L est donnée par la relation:

$$\frac{V_L}{4\beta_e} \dot{P}_L = Q - C_{tp} P_L - A(\dot{x}_s - \dot{x}_w) \quad (\text{I.22})$$

Où V_t est le volume total du cylindre, β_e , C_{tp} sont des caractéristiques du fluide utilisé, Q est le flux hydraulique .

L'équation du flux hydraulique de la servovalve est donnée par :

Pour éviter le cas $P_s - \text{sgn}(x_v)P_L < 0$, l'équation précédente devient:

$$Q = \text{sgn}[P_s - \text{sgn}(x_v)P_L]C_d w x_v \sqrt{\frac{1}{\rho} |P_s - \text{sgn}(x_v)P_L|}. \quad (I.23)$$

En fait le déplacement de l'électrovanne est contrôlé par l'entrée u (commande électrique). La dynamique de la valve est d'ordre 1 de constante de temps τ , elle est de la forme :

$$\dot{x}_v = \frac{1}{\tau}(-x_v + u) \quad (I.24)$$

I.5.2.2 Modèle quart de véhicule non linéaire :

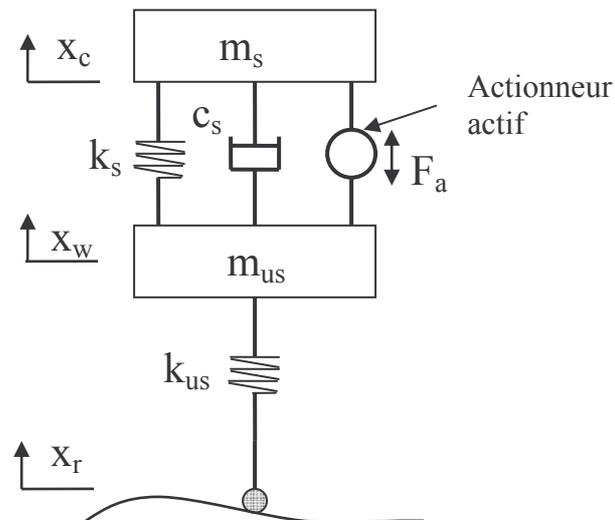


Figure 1.8 : Modèle quart de véhicule [3]

En tenant compte de des équations précédentes, et en posant :

$$\alpha = \frac{4\beta_e}{V_t} \quad \beta = \alpha.C_{tp} \quad \gamma = \alpha.C_d.\omega.\sqrt{\frac{1}{\rho}},$$

- $X_1 = X_C$: déplacement de la caisse.
- $X_2 = \dot{X}_C$: vitesse de la caisse.
- $X_3 = X_W$: déplacement de la roue.
- $X_4 = \dot{X}_W$: vitesse de la roue.

$X_5 = P_L$: Pression développée par le piston hydraulique

$X_6 = X_{SP} = X_v$: déplacement de la soupape.

u : entrée de commande.

On obtient le système d'état suivant :

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = -\frac{1}{m_s} \{k_s(X_1 - X_3) + C_s(X_2 - X_4) - AX_5\}$$

$$\dot{X}_3 = X_4$$

$$\dot{X}_4 = \frac{1}{m_{US}} \{k_s(X_1 - X_3) + C_s(X_2 - X_4) - k_{US}(X_3 - X_r) - AX_5\}$$

$$\dot{X}_5 = -\beta X_5 - \alpha A(X_2 - X_4) + \gamma \omega_3 X_6$$

$$\dot{X}_6 = \frac{1}{\tau} (u - X_6)$$

avec $\omega_3 = \text{sgn}[P_s - \text{sgn}(X_6)X_5] \sqrt{|P_s - \text{sgn}(X_6)X_5|}$

I.5.2.3 Modèle demi véhicule non linéaire [8], [3] :

De la même manière que le modèle quart de véhicule, l'introduction des dynamiques des actionneurs, donne lieu à des non linéarités dans la modélisation du demi véhicule. On reprend donc le schéma précédent du modèle demi véhicule pour extraire les équations décrivant le système.

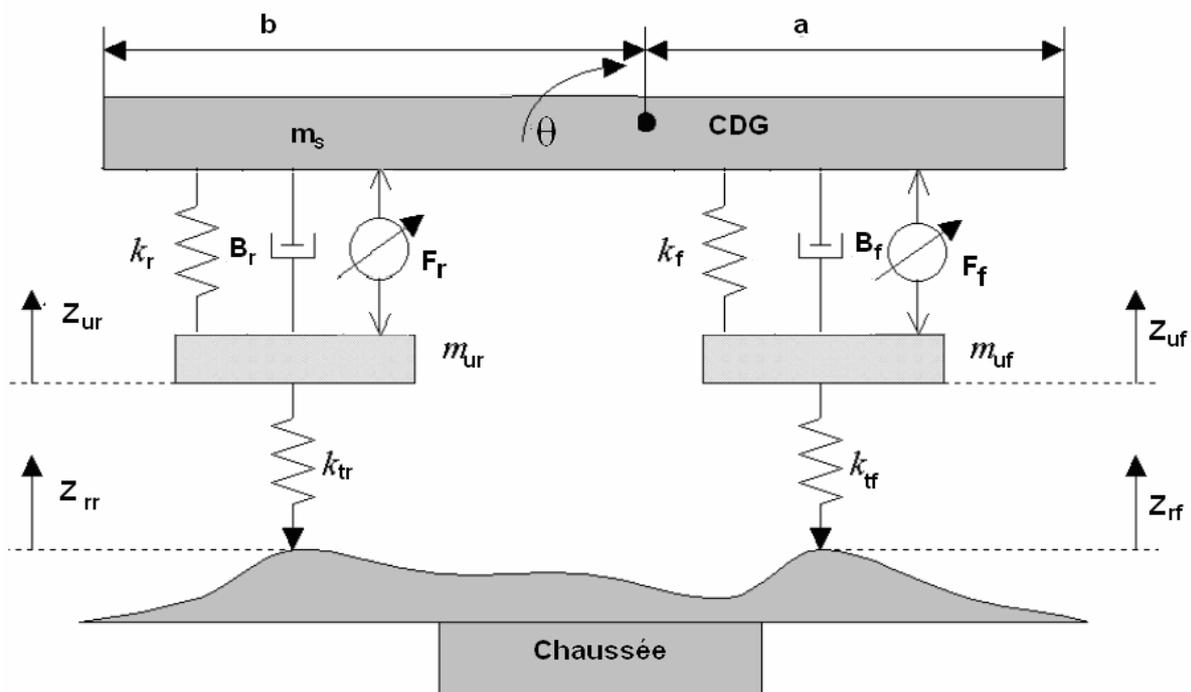


Figure 1.9 : Modèle non linéaire du demi véhicule (type bicyclette) [8]

A partir de la figure, on aura :

$$\text{Partie avant ; } Z_{sf} = Z - a.\sin(\theta) \quad (\text{I.25})$$

$$\text{Partie arrière ; } Z_{sr} = Z + b.\sin(\theta) \quad (\text{I.26})$$

Les forces équivalentes des deux cotés sont les suivantes ;

Partie avant ;

$$\begin{aligned} F_f &= -k_f.(Z_{sf} - Z_{uf}) - B_f.(\dot{Z}_{sf} - \dot{Z}_{uf}) + f_f \\ &= -k_f.(Z - a.\sin(\theta) - Z_{uf}) - B_f.(\dot{Z} - a\dot{\theta}.\cos(\theta) - \dot{Z}_{uf}) + f_f \end{aligned} \quad (\text{I.27})$$

Partie arrière ;

$$\begin{aligned} F_r &= -k_r.(Z_{sr} - Z_{ur}) - B_r.(\dot{Z}_{sr} - \dot{Z}_{ur}) + f_r \\ &= -k_r.(Z + b.\sin(\theta) - Z_{ur}) - B_r.(\dot{Z} + b\dot{\theta}.\cos(\theta) - \dot{Z}_{ur}) + f_r \end{aligned} \quad (\text{I.28})$$

En appliquant la deuxième lois de Newton, et en considérant la position d'équilibre statique comme origine des déplacements linéaires et angulaires du centre de gravité, on aura ;

$$\ddot{Z} = \frac{1}{m_s} \left\{ \begin{aligned} &-(k_f + k_r).Z - (B_f - B_r).\dot{Z} + (a.k_f - b.k_r).\sin(\theta) + (a.B_f - b.B_r).\dot{\theta}.\cos(\theta) \\ &+ k_f.Z_{uf} + k_r.Z_{ur} + B_f.\dot{Z}_{uf} + B_r.\dot{Z}_{ur} + f_f + f_r \end{aligned} \right\} \quad (\text{I.29})$$

Pour le mouvement angulaire, on aura :

$$\begin{aligned} J_y.\ddot{\theta} &= -F_f.a.\cos(\theta) + F_r.b.\cos(\theta) \\ &= \left(\begin{aligned} &a.k_f.(Z - a.\sin(\theta) - Z_{uf}).\cos(\theta) + a.B_f.(\dot{Z} - a\dot{\theta}.\cos(\theta) - \dot{Z}_{uf}).\cos(\theta) \\ &- b.k_r.(Z + b.\sin(\theta) - Z_{ur}).\cos(\theta) - b.B_r.(\dot{Z} + b\dot{\theta}.\cos(\theta) - \dot{Z}_{ur}).\cos(\theta) \\ &- a.\cos(\theta).f_f + b.\cos(\theta).f_r \end{aligned} \right) \end{aligned} \quad (\text{I.30})$$

Posons $J_y = m_s.r_y^2$ ou r_y est le rayon de giration.

$$\ddot{\theta} = \frac{1}{m_s.r_y^2} \left(\begin{aligned} &(a.k_f - b.k_r).Z.\cos(\theta) + (a.B_f - b.B_r).\dot{Z}.\cos(\theta) - (a^2.k_f + b^2.k_r).\sin(\theta).\cos(\theta) \\ &-(a^2.B_f + b^2.B_r).\dot{\theta}.\cos^2(\theta) - a.k_f.\cos(\theta).Z_{uf} + b.k_r.\cos(\theta).Z_{ur} \\ &- a.B_f.\cos(\theta).\dot{Z}_{uf} + b.B_r.\cos(\theta).\dot{Z}_{ur} - a.\cos(\theta).f_f + b.\cos(\theta).f_r \end{aligned} \right) \quad (\text{I.31})$$

Pour les roues on fait un calcul analogue permettant d'obtenir :

Roue avant ;

$$m_{uf}.\ddot{Z}_{uf} = B_f.(\dot{Z} - a\dot{\theta}.\cos(\theta) - \dot{Z}_{uf}) + k_f.(Z - a.\sin(\theta) - Z_{uf}) - f_f - k_{tr}.(Z_{uf} - Z_{rf}) \quad (\text{I.32})$$

Roue arrière ;

$$m_{ur}.\ddot{Z}_{ur} = B_r.(\dot{Z} + b\dot{\theta}.\cos(\theta) - \dot{Z}_{ur}) + k_r.(Z + b.\sin(\theta) - Z_{ur}) - f_r - k_{tr}.(Z_{ur} - Z_{rr}) \quad (\text{I.33})$$

En tenant compte des dynamiques non linéaires des actionneurs (on considère les mêmes actionneurs que le cas d'un quart de véhicule), et en choisissant les variables d'états suivantes,

$$\begin{aligned} X_1 = Z & & X_2 = \dot{Z} & & X_3 = \theta & & X_4 = \dot{\theta} & & X_5 = Z_{uf} & & X_6 = \dot{Z}_{uf} \\ X_7 = Z_{ur} & & X_8 = \dot{Z}_{ur} & & X_9 = P_{L1} & & X_{10} = P_{L2} & & X_{11} = X_{v1} & & X_{12} = X_{v2} \end{aligned}$$

Où :

X_{v1} et X_{v2} : Positions des deux pistons des deux actionneurs

P_{L1} et P_{L2} : Chutes de pressions dans les deux pistons des roues avant et arrière.

On aura :

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \left(\begin{aligned} &-\frac{k_f + k_r}{m_s} X_1 - \frac{B_f + B_r}{m_s} X_2 + \frac{ak_f - bk_r}{m_s} \sin(X_3) + \frac{aB_f - bB_r}{m_s} \cos(X_3) \cdot X_4 \\ &+ \frac{k_f}{m_s} X_5 + \frac{B_f}{m_s} X_6 + \frac{k_r}{m_s} X_7 + \frac{B_r}{m_s} X_8 + \frac{A_p \cdot (X_9 + X_{10})}{m_s} \end{aligned} \right) \end{aligned}$$

$$\begin{aligned} \dot{X}_3 &= X_4 \\ \dot{X}_4 &= \left(\begin{aligned} &\frac{a.k_f - bk_r}{m_s \cdot r_y^2} X_1 \cdot \cos(X_3) + \frac{a.B_f - bB_r}{m_s \cdot r_y^2} X_2 \cdot \cos(X_3) - \frac{a^2 k_f + b^2 k_r}{2 \cdot m_s \cdot r_y^2} \sin(2 \cdot X_3) \\ &- \frac{a^2 B_f + b^2 B_r}{m_s \cdot r_y^2} X_4 \cdot \cos^2(X_3) - \frac{aK_f}{m_s \cdot r_y^2} X_5 \cdot \cos(X_3) - \frac{aB_f}{m_s \cdot r_y^2} X_6 \cdot \cos(X_3) \\ &+ \frac{bk_r}{m_s \cdot r_y^2} X_7 \cdot \cos(X_3) + \frac{bB_r}{m_s \cdot r_y^2} X_8 \cdot \cos(X_3) - A_p \cdot \frac{aX_9 + bX_{10}}{m_s \cdot r_y^2} \cdot \cos(X_3) \end{aligned} \right) \end{aligned}$$

$$\begin{aligned} \dot{X}_5 &= X_6 \\ \dot{X}_6 &= \left(\begin{aligned} &\frac{k_f}{m_{uf}} X_1 + \frac{B_f}{m_{uf}} X_2 - \frac{ak_f}{m_{uf}} \sin(X_3) - \frac{aB_f}{m_{uf}} X_4 \cdot \cos(X_3) - \frac{k_f + k_{yf}}{m_{uf}} X_5 - \frac{B_f}{m_{uf}} X_6 \\ &- \frac{A_p \cdot X_9}{m_{uf}} + \frac{k_{yf}}{m_{uf}} Z_{rf} \end{aligned} \right) \end{aligned}$$

$$\begin{aligned} \dot{X}_7 &= X_8 \\ \dot{X}_8 &= \left(\begin{aligned} &\frac{k_r}{m_{ur}} X_1 + \frac{B_r}{m_{ur}} X_2 + \frac{bk_r}{m_{ur}} \sin(X_3) + \frac{bB_r}{m_{ur}} X_4 \cdot \cos(X_3) - \frac{k_r + k_{tr}}{m_{ur}} X_7 - \frac{B_r}{m_{ur}} X_8 \\ &- \frac{A_p \cdot X_{10}}{m_{ur}} + \frac{k_{tr}}{m_{ur}} Z_{rr} \end{aligned} \right) \end{aligned}$$

$$\dot{X}_9 = -\beta_f \cdot X_9 - \alpha_f \cdot A_p \cdot (X_2 - a \cdot X_4 \cdot \cos(X_3) - X_6) + \omega_f \cdot \gamma_f \cdot X_{11}$$

$$\dot{X}_{10} = -\beta_r \cdot X_{10} - \alpha_r \cdot A_p \cdot (X_2 + b \cdot X_4 \cdot \cos(X_4) - X_8) + \omega_r \cdot \gamma_r \cdot X_{12}$$

$$\dot{X}_{11} = \frac{1}{\tau} \cdot (u_f - X_{11})$$

$$\dot{X}_{12} = \frac{1}{\tau} \cdot (u_r - X_{12})$$

$$\text{avec } \omega_f = \text{sgn} \left[P_{sf} - \text{sgn}(X_{11}) \cdot X_9 \right] \cdot \sqrt{|P_{sf} - \text{sgn}(X_{11}) \cdot X_9|}$$

$$\omega_r = \text{sgn} \left[P_{sr} - \text{sgn}(X_{12}) \cdot X_{10} \right] \cdot \sqrt{|P_{sr} - \text{sgn}(X_{12}) \cdot X_{10}|}$$

$$U = \begin{bmatrix} u_f \\ u_r \end{bmatrix} \text{ la commande .}$$

$$d = \begin{bmatrix} Z_{rf} & Z_{rr} \end{bmatrix} \text{ profil de la route .}$$

I.5.2.4 Modèle véhicule complet non linéaire [5]:

On introduit des variables supplémentaires caractérisant la dynamique des 4 actionneurs,

x_{15} : chute de pression actionneur avant gauche

x_{16} : électrovanne avant gauche

x_{17} : chute de pression actionneur avant droit

x_{18} : électrovanne avant droite

x_{19} : chute de pression actionneur arrière gauche

x_{20} : électrovanne arrière gauche

x_{21} : chute de pression actionneur arrière droit

x_{22} : électrovanne arrière droite

Et on considère les angles de tangage et roulis ; ce qui donne le modèle suivant :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{(2k_{sf} + 2k_{sr})}{m_s} x_1 - \frac{(2B_{sf} + 2B_{sr})}{m_s} x_2 + \frac{(2ak_{sf} - 2bk_{sr})}{m_s} \sin(x_3) + \frac{(2aB_{sf} - 2bB_{sr})}{m_s} x_4 \cos(x_3) + \frac{k_{sf}}{m_s} x_7 + \frac{B_{sf}}{m_s} x_8 + \frac{k_{sf}}{m_s} x_9 + \frac{B_{sf}}{m_s} x_{10} + \frac{k_{sr}}{m_s} x_{11} + \frac{B_{sr}}{m_s} x_{12} + \frac{k_{sr}}{m_s} x_{13} + \frac{B_{sr}}{m_s} x_{14} + A_p \frac{x_{15} + x_{17} + x_{19} + x_{21}}{m_s}$$

$$\dot{x}_3 = x_4$$

$$\begin{aligned} \dot{x}_4 = & \frac{(2ak_{sf} - 2bk_{sr})}{I_{yy}} x_1 + \frac{(2aB_{sf} - 2bB_{sr})}{I_{yy}} x_2 - \frac{(2a^2k_{sf} + 2b^2k_{sr})}{I_{yy}} \sin(x_3) - \frac{(2a^2B_{sf} + 2b^2B_{sr})}{I_{yy}} x_4 \cos(x_3) - \\ & \frac{ak_{sf}}{I_{yy}} x_7 - \frac{aB_{sf}}{I_{yy}} x_8 - \frac{ak_{sf}}{I_{yy}} x_9 - \frac{aB_{sf}}{I_{yy}} x_{10} + \frac{bk_{sr}}{I_{yy}} x_{11} + \frac{bB_{sr}}{I_{yy}} x_{12} + \frac{bk_{sr}}{I_{yy}} x_{13} + \frac{bB_{sr}}{I_{yy}} x_{14} + \\ & A_p \frac{-a(x_{15} + x_{17}) + b(x_{19} + x_{21})}{I_{yy}} \end{aligned}$$

$$\dot{x}_5 = x_6$$

$$\begin{aligned} \dot{x}_6 = & -\frac{w^2(2k_{sf} + 2k_{sr})}{4I_{xx}} \sin(x_5) - \frac{w^2(2B_{sf} + 2B_{sr})}{4I_{xx}} x_6 \cos(x_5) + \frac{wk_{sf}}{2I_{xx}} x_7 + \frac{wB_{sf}}{2I_{xx}} x_8 - \frac{wk_{sf}}{2I_{xx}} x_9 - \\ & \frac{wB_{sf}}{2I_{xx}} x_{10} + \frac{wk_{sr}}{2I_{xx}} x_{11} + \frac{wB_{sr}}{2I_{xx}} x_{12} - \frac{wk_{sr}}{2I_{xx}} x_{13} - \frac{wB_{sr}}{2I_{xx}} x_{14} + A_p \frac{w(x_{15} - x_{17} + x_{19} - x_{21})}{2I_{xx}} \end{aligned}$$

$$\dot{x}_7 = x_8$$

$$\begin{aligned} \dot{x}_8 = & \frac{k_{sf}}{m_u} x_1 + \frac{B_{sf}}{m_u} x_2 - \frac{ak_{sf}}{m_u} \sin(x_3) - \frac{aB_{sf}}{m_u} x_4 \cos(x_3) + \frac{wk_{sf}}{2m_u} \sin(x_5) + \frac{wB_{sf}}{2m_u} x_6 \cos(x_5) - \frac{(k_{sf} + k_u)}{m_u} x_7 - \\ & \frac{B_{sf}}{m_u} x_8 + \frac{k_u}{m_u} z_{rfl} - A_p \frac{x_{15}}{m_u} \end{aligned}$$

$$\dot{x}_9 = x_{10}$$

$$\begin{aligned} \dot{x}_{10} = & \frac{k_{sf}}{m_u} x_1 + \frac{B_{sf}}{m_u} x_2 - \frac{ak_{sf}}{m_u} \sin(x_3) - \frac{aB_{sf}}{m_u} x_4 \cos(x_3) - \frac{wk_{sf}}{2m_u} \sin(x_5) - \frac{wB_{sf}}{2m_u} x_6 \cos(x_5) - \frac{(k_{sf} + k_u)}{m_u} x_9 - \\ & \frac{B_{sf}}{m_u} x_{10} + \frac{k_u}{m_u} z_{rfr} - A_p \frac{x_{17}}{m_u} \end{aligned}$$

$$\dot{x}_{11} = x_{12}$$

$$\begin{aligned} \dot{x}_{12} = & \frac{k_{sr}}{m_u} x_1 + \frac{B_{sr}}{m_u} x_2 + \frac{bk_{sr}}{m_u} \sin(x_3) + \frac{bB_{sr}}{m_u} x_4 \cos(x_3) + \frac{wk_{sr}}{2m_u} \sin(x_5) + \frac{wB_{sr}}{2m_u} x_6 \cos(x_5) - \frac{(k_{sr} + k_u)}{m_u} x_{11} - \\ & \frac{B_{sr}}{m_u} x_{12} + \frac{k_u}{m_u} z_{rrl} - A_p \frac{x_{19}}{m_u} \end{aligned}$$

$$\dot{x}_{13} = x_{14}$$

$$\begin{aligned} \dot{x}_{14} = & \frac{k_{sr}}{m_u} x_1 + \frac{B_{sr}}{m_u} x_2 + \frac{bk_{sr}}{m_u} \sin(x_3) + \frac{bB_{sr}}{m_u} x_4 \cos(x_3) - \frac{wk_{sr}}{2m_u} \sin(x_5) - \frac{wB_{sr}}{2m_u} x_6 \cos(x_5) - \frac{(k_{sr} + k_u)}{m_u} x_{13} - \\ & \frac{B_{sr}}{m_u} x_{14} + \frac{k_u}{m_u} z_{rrr} - A_p \frac{x_{21}}{m_u} \end{aligned}$$

$$\dot{x}_{15} = -\beta x_{15} - \alpha A_p (x_2 + \frac{w}{2} x_6 \cos(x_5) - ax_4 \cos(x_3) - x_8) + \gamma \omega_{15} x_{16}$$

$$\dot{x}_{16} = \frac{1}{\tau} (u_{fl} - x_{16})$$

$$\dot{x}_{17} = -\beta x_{17} - \alpha A_p \left(x_2 - \frac{w}{2} x_6 \cos(x_5) - ax_4 \cos(x_3) - x_{10} \right) + \gamma \omega_{17} x_{18}$$

$$\dot{x}_{18} = \frac{1}{\tau} (u_{fr} - x_{18})$$

$$\dot{x}_{19} = -\beta x_{19} - \alpha A_p \left(x_2 + \frac{w}{2} x_6 \cos(x_5) + bx_4 \cos(x_3) - x_{12} \right) + \gamma \omega_{19} x_{20}$$

$$\dot{x}_{20} = \frac{1}{\tau} (u_{rl} - x_{20})$$

$$\dot{x}_{21} = -\beta x_{21} - \alpha A_p \left(x_2 - \frac{w}{2} x_6 \cos(x_5) + bx_4 \cos(x_3) - x_{14} \right) + \gamma \omega_{21} x_{22}$$

$$\dot{x}_{22} = \frac{1}{\tau} (u_{rr} - x_{22})$$

$$\text{avec : } \omega_{15} = \text{sign}(P_s - \text{sign}(x_{16})x_{15}) \sqrt{|P_s - \text{sign}(x_{16})x_{15}|}$$

$$\omega_{17} = \text{sign}(P_s - \text{sign}(x_{18})x_{17}) \sqrt{|P_s - \text{sign}(x_{18})x_{17}|}$$

$$\omega_{19} = \text{sign}(P_s - \text{sign}(x_{20})x_{19}) \sqrt{|P_s - \text{sign}(x_{20})x_{19}|}$$

$$\omega_{21} = \text{sign}(P_s - \text{sign}(x_{22})x_{21}) \sqrt{|P_s - \text{sign}(x_{22})x_{21}|}$$

I.5.3 Simulation des modèles en boucle ouverte

I.5.3.1 Introduction

Dans cette partie nous allons simuler des modèles traités dans la partie précédente.

La simulation des modèles linéaires consiste à représenter le comportement de la suspension passive aux variations-types du profil de la chaussée (trottoir, dos d'âne), ce qui va être comparé aux résultats obtenus avec les modèles non linéaires afin de conclure sur la validité des approximations utilisées pour la linéarisation.

I.5.3.2 Modèle quart de véhicule linéaire:

Les valeurs des paramètres utilisés dans nos simulations sont :

$$M_s = 300\text{kg} \quad : \text{Masse de la caisse (un quart de la masse de la caisse).}$$

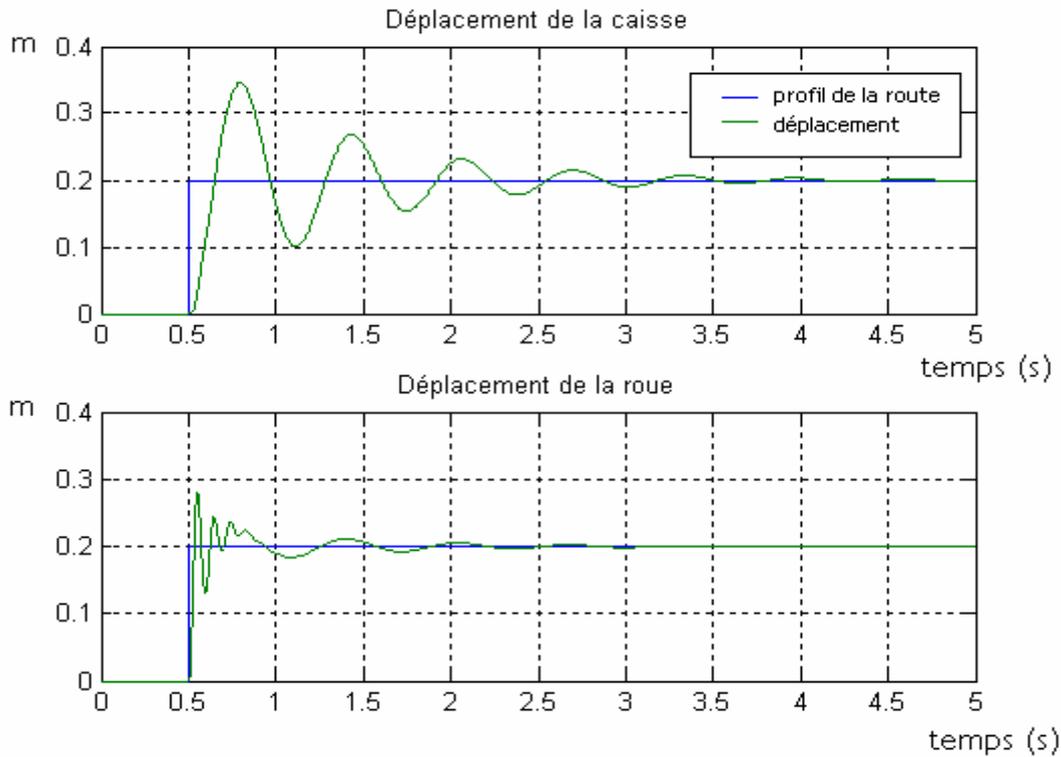
$$M_{us} = 50\text{kg} \quad : \text{Masse de la roue.}$$

$$K_s = 35000\text{N/m} \quad : \text{Constante de raideur du ressort caisse.}$$

$$K_u = 190\,000\text{N/m} \quad : \text{Constante de raideur du ressort pneu}$$

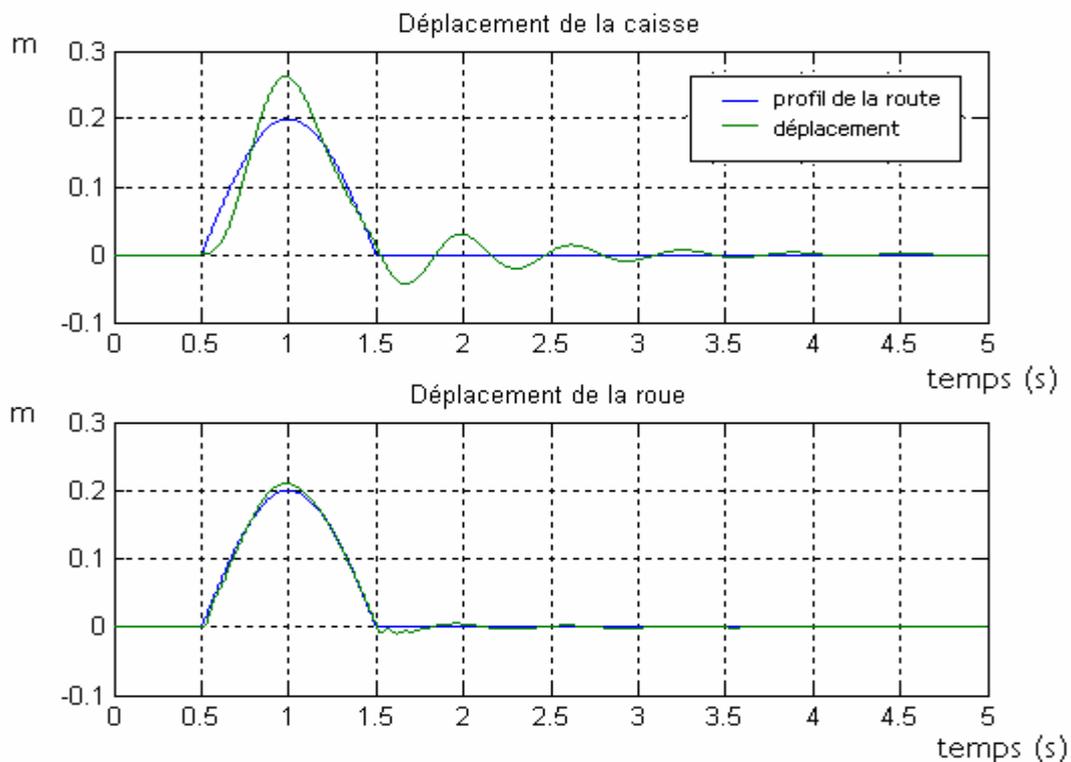
$$B_s = 1000\text{Ns/m} \quad : \text{L'amortissement de la suspension.}$$

I.5.3.2.1 montée sur un trottoir de 20 cm



Figure(1.10) : Montée sur trottoir de 20 cm / modèle ¼ de véhicule

I.5.3.2.2 passage sur une bosse (dos d'âne d'amplitude 20 cm / base =1m)



Figure(1.11) : Montée sur trottoir de 20 cm / modèle ¼ de véhicule

I.5.3.3 Modèle demi véhicule :

Les valeurs des paramètres utilisés sont:

$M_{uf}, M_{ur} = 59 \text{ kg}$ chacune : Masses des roues.

$K_{sf} = 35000 \text{ N/m}$: Constantes de raideur de ressort avant.

$K_{sr} = 38000 \text{ N/m}$: Constantes de raideur de ressort arrière.

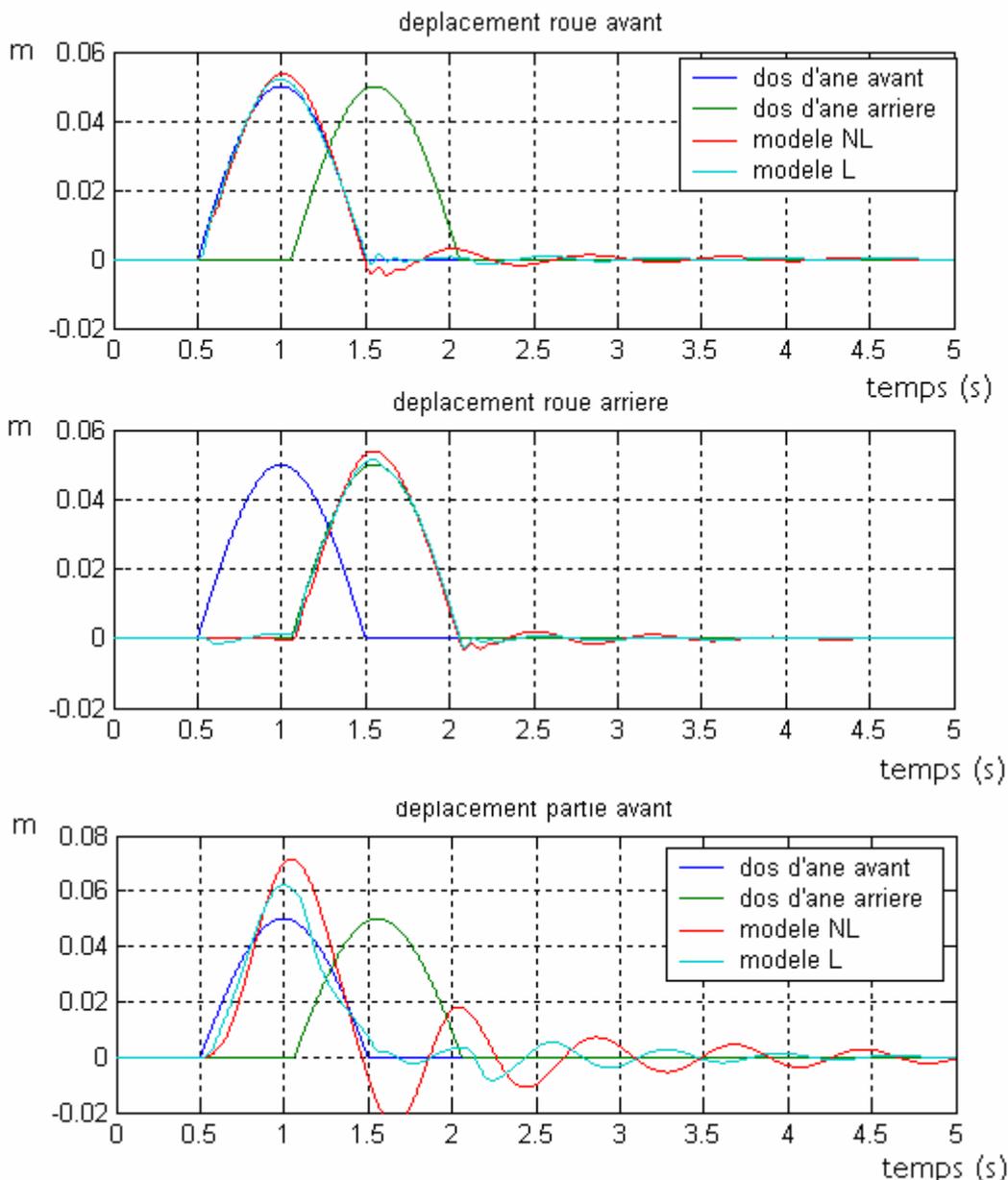
$K_{uf}, K_{ur} = 190\,000 \text{ N/m}$: Constantes de raideur des ressorts pneu.

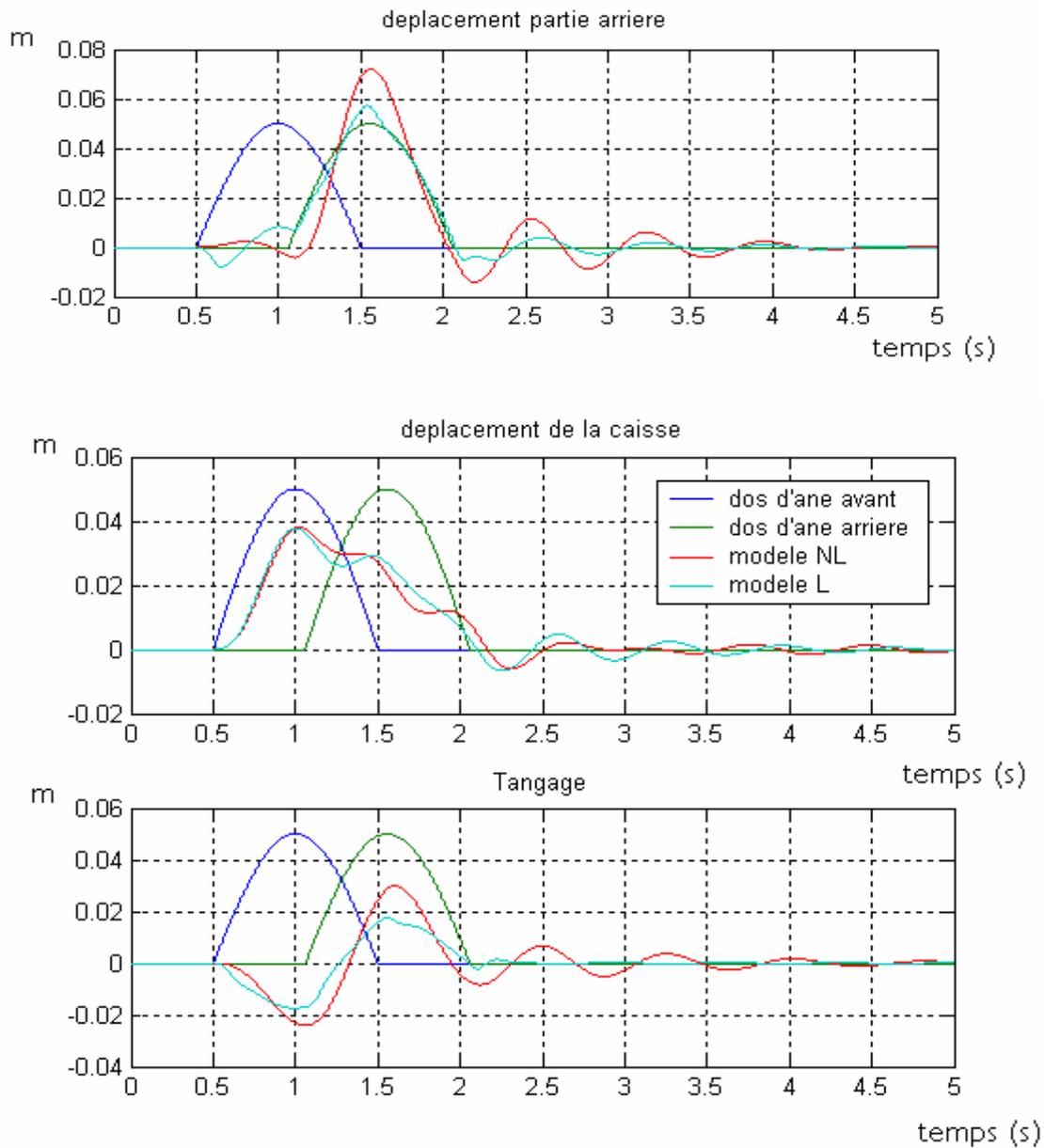
$B_{sf}, B_{sr} = 1000 \text{ Ns/m}$: Coefficients d'amortissement.

$M_s = 350 \times 2 = 700 \text{ Kg}$: Masse de notre châssis.

$a = 1.4 \text{ m}$ $b = 1.7 \text{ m}$: Distance entre les extrémités et le centre de gravité.

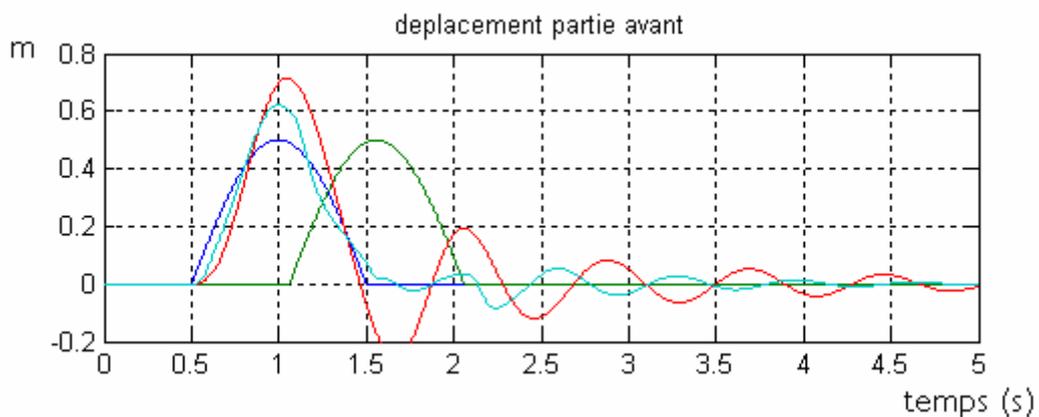
I.5.3.3.1 passage sur bosse à faible amplitude (5 cm) / vitesse=5 km/h

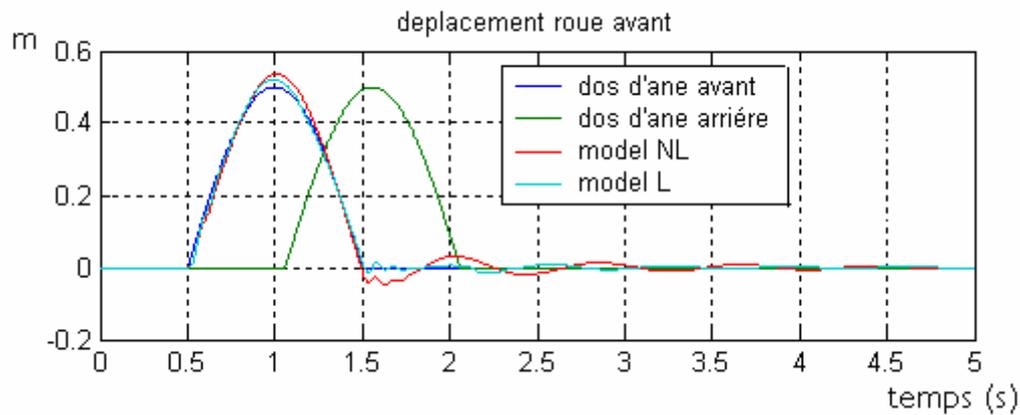
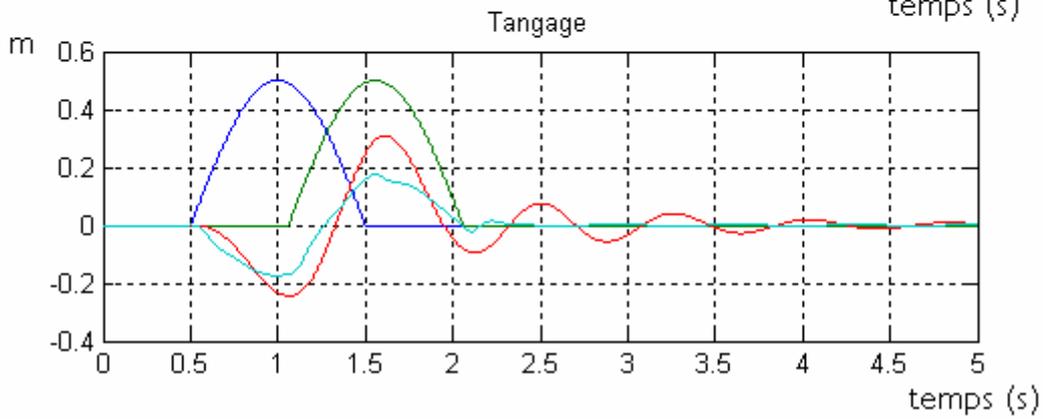
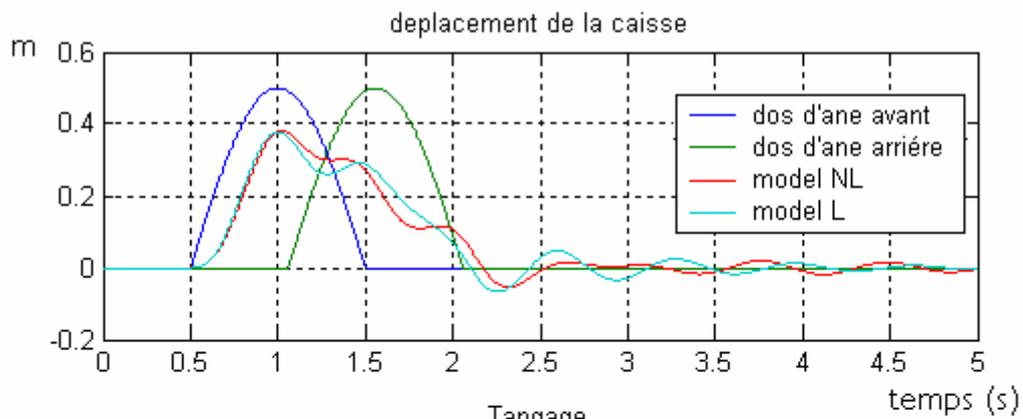
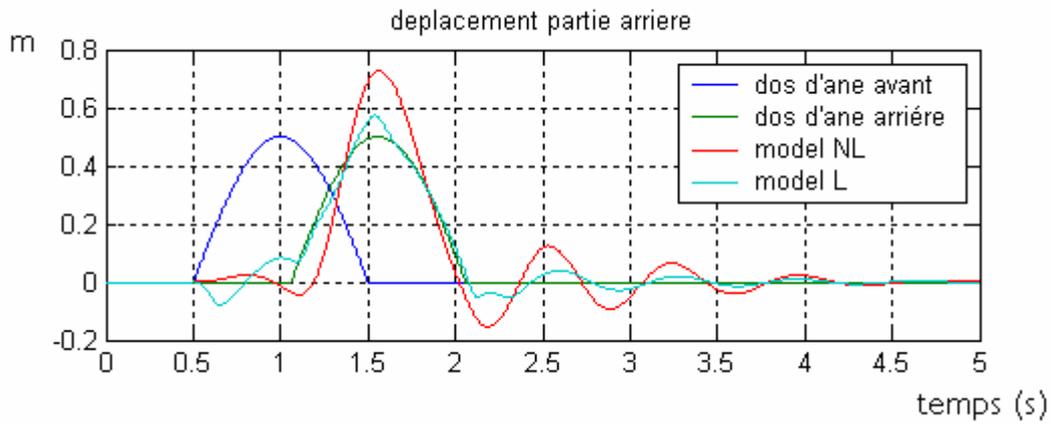


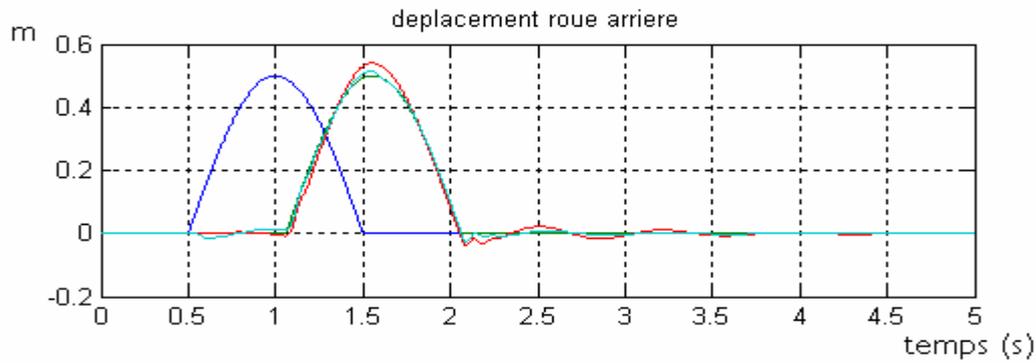


Figure(1.12) :Passage sur dos d’âne de 5 cm/ modèle demi véhicule

I.5.3.3.2 passage sur bosse à forte amplitude (50cm) / vitesse=5 km/h







Figure(1.13) :Passage sur dos d’âne de 50 cm/ modèle demi véhicule

I.5.3.4 Modèle véhicule complet linéaire:

Les valeurs des paramètres du modèle sont :

$M_u = 59$ kg chacune : Masses des roues.

$K_{sfl}, K_{sfr} = 35000$ N/m : Constantes de raideur de ressort avant.

$K_{srl}, K_{srr} = 38000$ N/m : Constantes de raideur de ressort arrière.

$K_{ufr}, K_{ufr}, K_{ur}, K_{urr} = 190\ 000$ N/m : Constantes de raideur des ressorts pneu.

$B_{sfl}, B_{sfr} = 1000$ Ns/m : Coefficients d’amortissement avant.

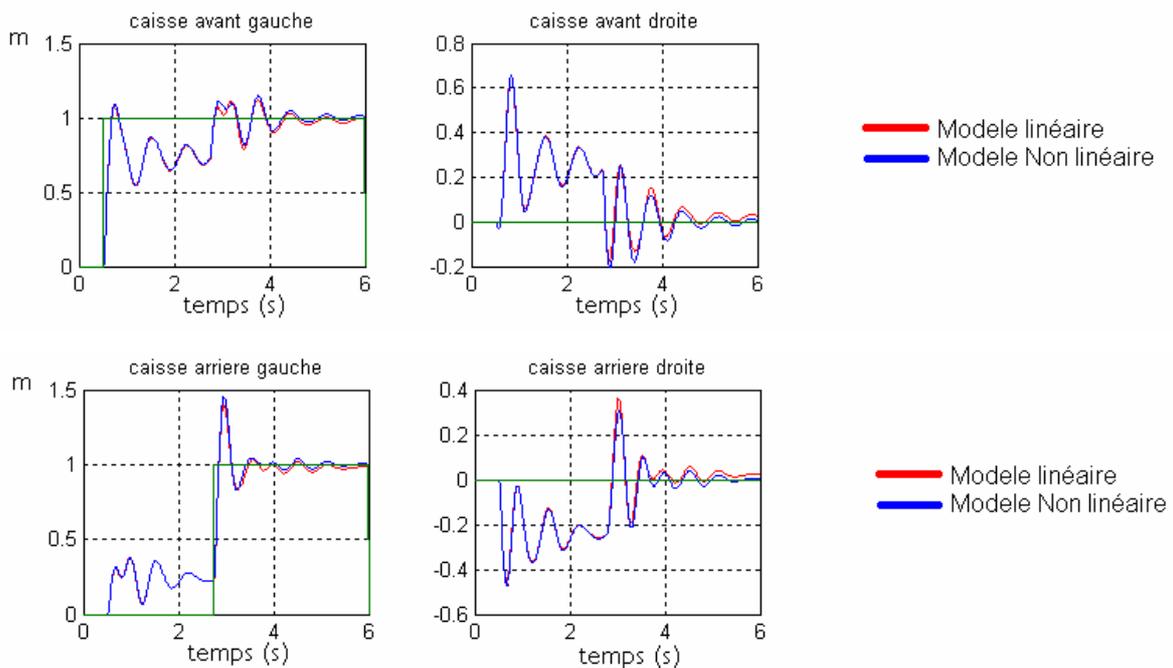
$B_{srl}, B_{srr} = 1100$ Ns/m : Coefficients d’amortissement arrière.

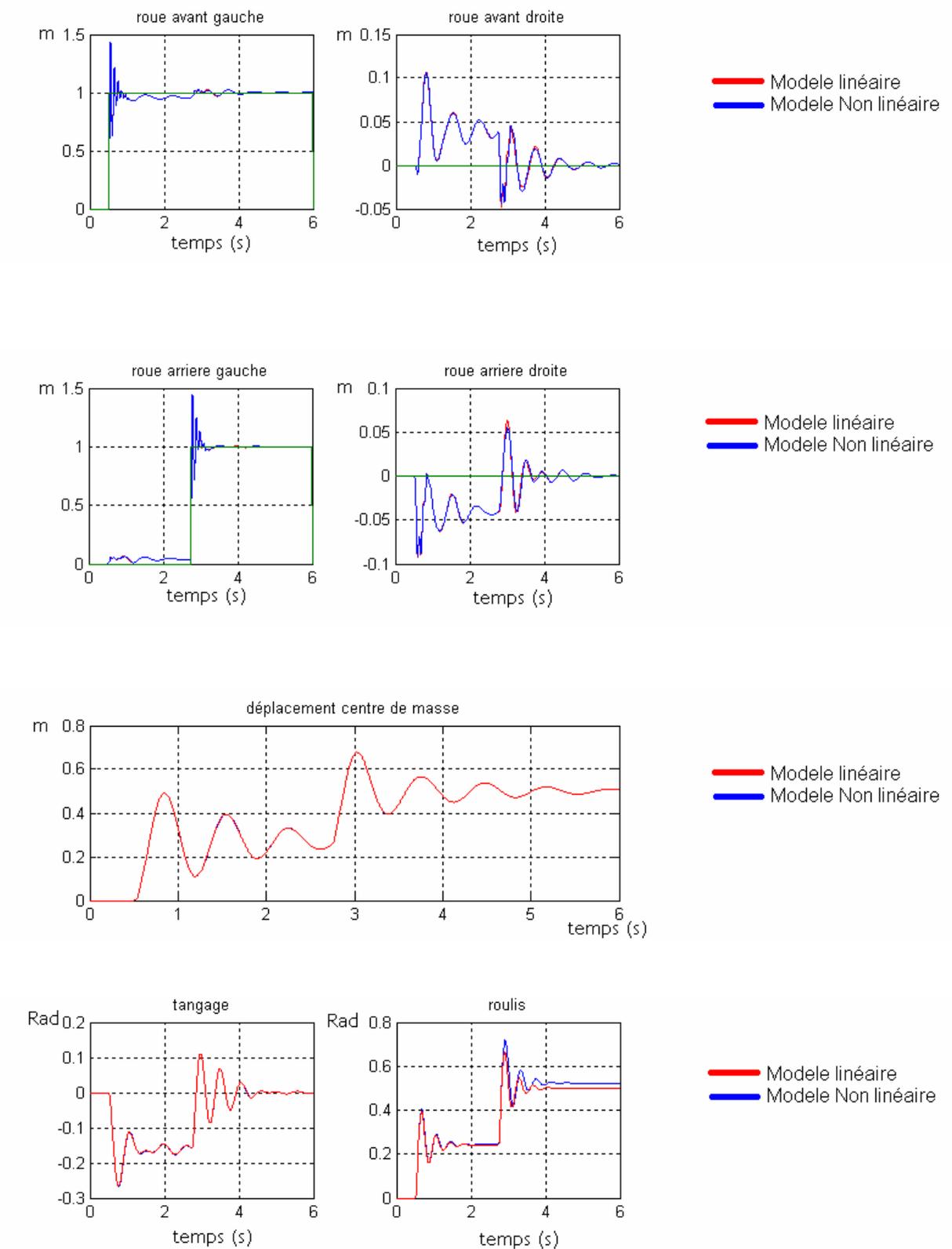
$M_s = 1500$ Kg : Masse de notre châssis.

$a = 1.4$ m $b = 1.7$ m : Distance entre les extrémités et le centre de gravité.

$w = 2$ m : largeur du véhicule.

I.5.3.4.1 réponse à un échelon d’amplitude 1m sur les roues gauches (vitesse 5 km/h):





Figure(1.14) : réponse à un échelon sur roues gauches/ modèle véhicule complet

I.5.3.4.2 réponse à un échelon d'amplitude 2 m sur toutes les roues (vitesse 5 km/h):

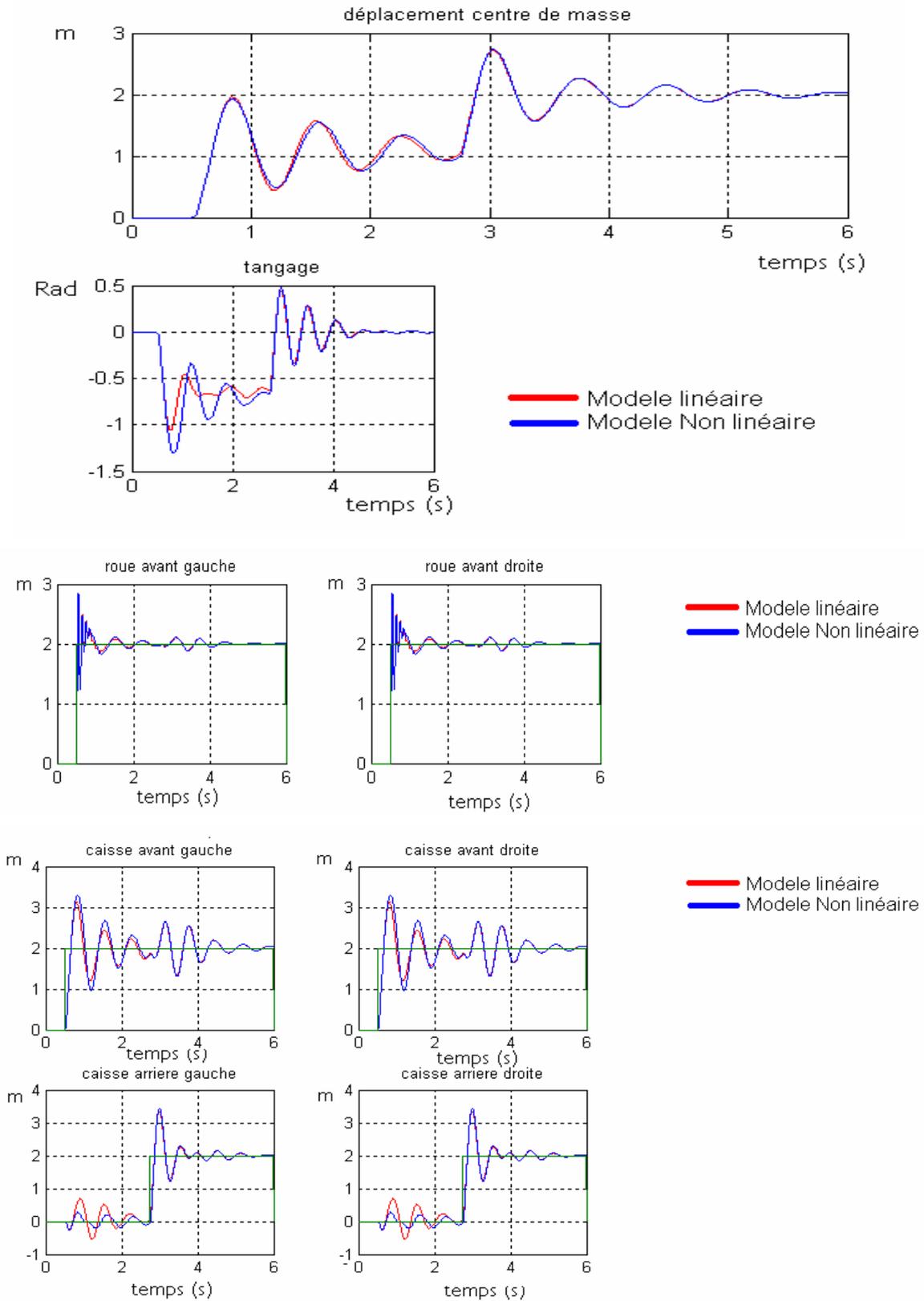


Figure (1.15) : réponse à un échelon de 2m (perturbation) / modèle véhicule complet

I.5.3.5 Modèle véhicule complet non linéaire:

I.5.3.5.1 Réponse à un échelon de commande sur tous les actionneurs avec montée sur trottoir:

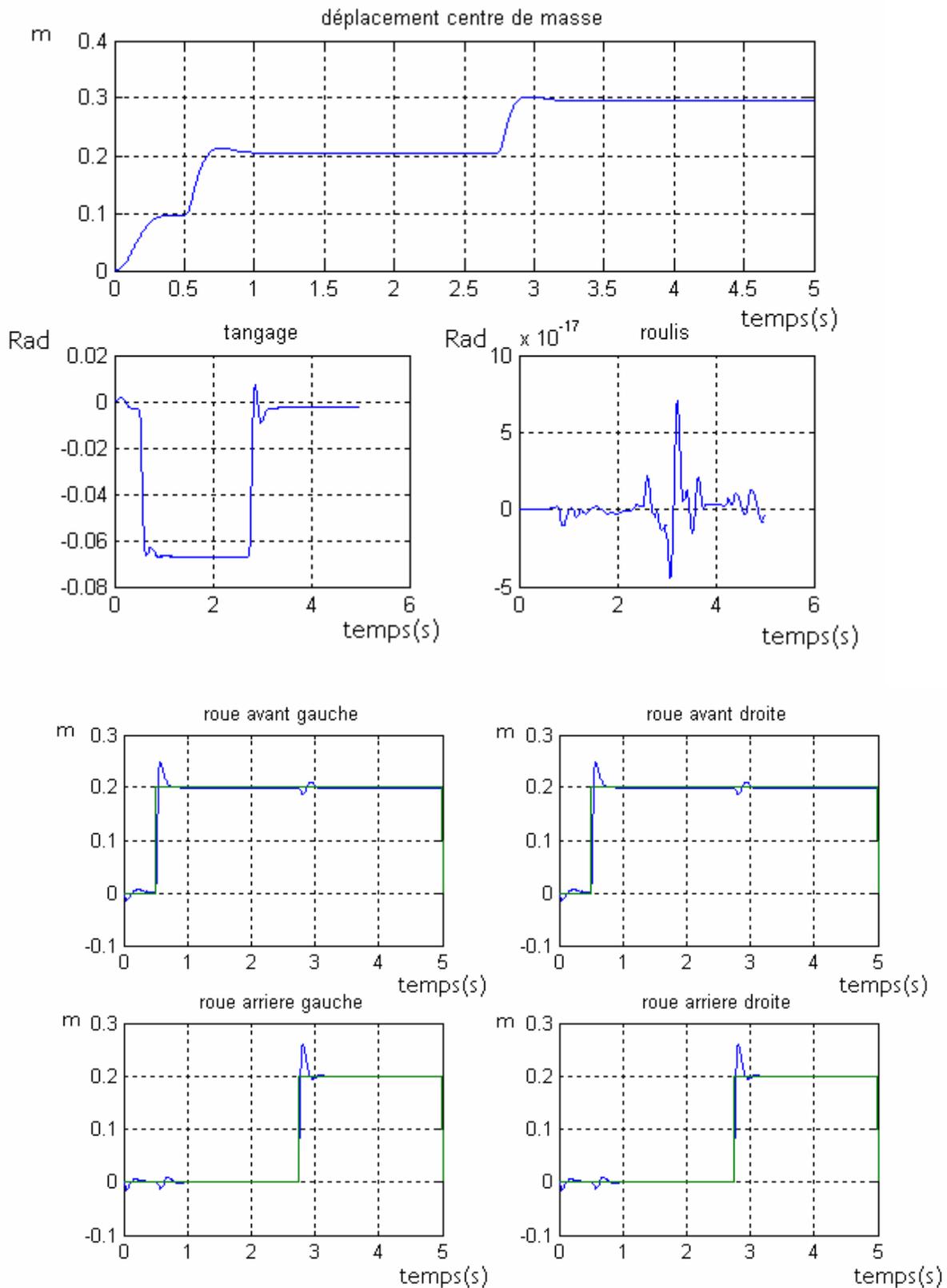


Figure (1.16) : réponse à un échelon de commande / modèle véhicule complet

I.6 Conclusions :

Dans cette partie nous avons étudié les différents modèles d'état, linéaires et non linéaires, de la suspension de véhicule et nous avons développé les équations relatives au modèle complet non linéaire.

Les simulations effectuées ont permis de comparer les modèles linéaires et non linéaires, afin de valider les approximations de la linéarisation.

D'après les résultats obtenus nous avons constaté que :

- * le modèle complet est plus représentatif que les autres modèles bien qu'il soit plus complexe, ceci est dû au fait qu'il prend en considération tous les déplacements du véhicule dans les 3 axes de l'espace ainsi que l'influence mutuelle du déplacement de chaque côté de la caisse sur les autres (interconnexion).
- * Les modèles non linéaires donnent des résultats plus précis que les modèles linéaires, mais pour des petites variations des différents angles (ce qui est le cas en pratique) l'approche linéaire donne des résultats satisfaisants, tout en étant plus simple en terme de complexité et de volume de calcul.

**Chapitre II :
Commande neurofloue**

II.1 Introduction

Dans cette partie on va s'intéresser à l'étude de la technique neurofloue qui figure parmi les techniques de l'intelligence artificielle ; l'utilisation de cette technique va nous permettre de concevoir un système de suspension modulaire ayant la faculté d'adaptation aux variations paramétriques de la suspension ; et ceci grâce à la combinaison de la logique floue aux réseaux de neurones artificiels.

On va étudier d'abord les réseaux de neurones ; l'apprentissage dans les différents types de réseaux ; ensuite on va introduire les notions du réglage flou. La fin de ce chapitre sera soldée par l'élaboration de régulateur hybride combinant les deux techniques et tirant profit de leurs avantages respectifs.

II.2 Notions sur les réseaux de neurones artificiels

II.2.1 Introduction

Un réseau de neurones (ou Artificial Neural Network en anglais) est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de neurones biologiques (humains ou animaux). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont classés d'une part dans la famille des applications statistiques, et d'autre part dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique ferme [12].

Le cerveau humain se compose d'environ 10^{12} neurones (mille milliards), avec 1000 à 10000 synapses (connexions) par neurone. Il est caractérisé par [17]:

- une architecture massivement parallèle
- un mode de calcul et une mémoire distribués
- une grande capacité d'apprentissage
- une capacité de généralisation
- une capacité d'adaptation
- une résistance aux pannes
- une faible consommation énergétique

II.2.2 Le neurone

La cellule de base composant le cerveau est appelée NEURONE ; Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de

chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone.

L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. La jonction entre deux neurones est appelée la synapse [15].

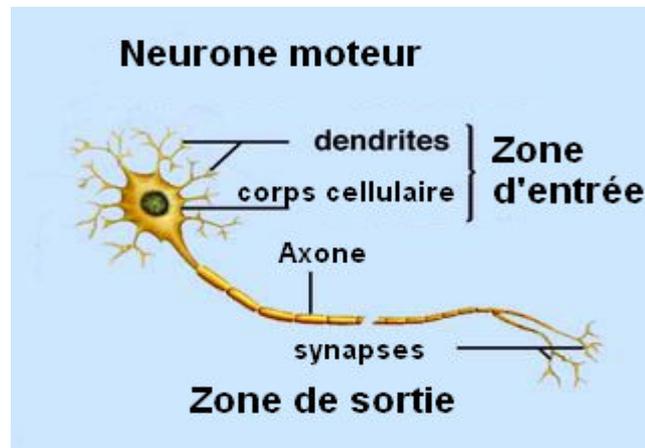


Figure 2.1 : neurone biologique [17]

La transmission de l'information se fait suivant deux modes :

- 1- à travers les axones ; par changement de potentiel ionique
- 2- à travers les synapses (transmission chimique)

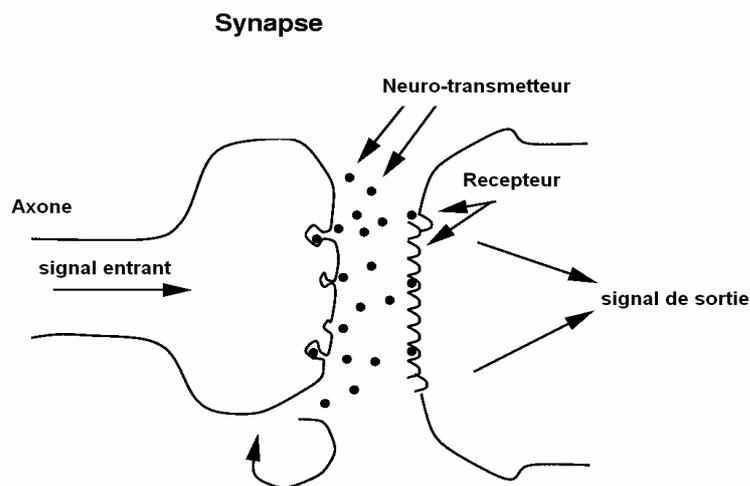


Figure 2.2 : synapse [17]

II.2.3 Historique [17]

Comme nous l'avons évoqué plus haut, les réseaux de neurones sont construits sur un paradigme biologique (neurone). Les neurologues *Warren Sturgis McCulloch* et *Walter*

Pitts menèrent les premiers travaux sur les réseaux de neurones à la suite de leur article fondateur : « *What the frog's eye tells to the frog's brain.* » Ils constituèrent un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent également théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

La fonction des réseaux de neurones formels à l'instar du modèle vivant est de résoudre des problèmes. À l'opposé des méthodes traditionnelles de résolution informatique, on ne doit pas construire un programme pas à pas. Les paramètres les plus importants de ce modèle sont les coefficients synaptiques. Ce sont eux qui construisent le modèle de résolution en fonction des informations données au réseau. Il faut donc trouver un mécanisme qui permette de les calculer à partir des grandeurs que l'on peut acquérir du problème. C'est le principe fondamental de l'apprentissage. Dans un modèle de réseaux de neurones formels, apprendre, c'est d'abord calculer les valeurs des coefficients synaptiques en fonction des exemples disponibles.

Les travaux de *McCulloch* et *Pitts* n'ont pas donné d'indication sur une méthode pour adapter les coefficients synaptiques. Cette question au cœur des réflexions sur l'apprentissage a connu un début de réponse grâce aux travaux du physiologiste américain *Donald Hebb* sur l'apprentissage en 1949 décrits dans son ouvrage « *The Organization of Behaviour* ». *Hebb* a proposé une règle simple qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des unités qu'ils relient. Cette règle aujourd'hui connue sous le nom de « règle de *Hebb* » est presque partout présente dans les modèles actuels, même les plus sophistiqués.

À partir de cet article, l'idée se sema au fil du temps dans les esprits, et elle germa dans l'esprit de *Franck Rosenblatt* en 1957 avec le modèle du **perceptron**. C'est le premier système artificiel capable d'apprendre par expérience, y compris lorsque son instructeur commet quelques erreurs (ce en quoi il diffère nettement d'un système d'apprentissage logique formel). D'autres travaux marquèrent également le domaine, comme ceux du physiologiste américain *Donald Hebb* sur l'apprentissage en 1949.

En 1969, un coup grave fut porté à la communauté scientifique gravitant autour des réseaux de neurones : *Marvin Lee Minsky* et *Seymour Papert* publièrent un ouvrage mettant en exergue quelques limitations théoriques du Perceptron, notamment l'impossibilité de traiter des problèmes non linéaires ou de connexité. Ils étendirent implicitement ces limitations à tous modèles de réseaux de neurones artificiels. Paraissant alors une impasse, la recherche sur les réseaux de neurones perdit une grande partie de ses financements publics, et le secteur

industriel s'en détourna aussi. Les fonds destinés à l'intelligence artificielle furent redirigés plutôt vers la logique formelle et la recherche piétina pendant dix ans. Cependant, les solides qualités de certains réseaux de neurones en matière adaptative, (e.g. *Adaline*), leur permettant de modéliser de façon évolutive des phénomènes eux-mêmes évolutifs les amèneront à être intégrés sous des formes plus ou moins explicites dans le corpus des systèmes adaptatifs, utilisés dans le domaine des télécommunications ou celui du contrôle de processus industriels.

En 1982, *John Joseph Hopfield*, physicien reconnu, donna un nouveau souffle au neuronal en publiant un article introduisant un nouveau modèle de réseau de neurones (complètement récurrent). Cet article eût du succès pour plusieurs raisons, dont la principale était de teinter la théorie des réseaux de neurones de la rigueur propre aux physiciens. Le neuronal redevint un sujet d'étude acceptable, bien que le modèle de *Hopfield* souffrait des principales limitations des modèles des années 1960, notamment l'impossibilité de traiter les problèmes non-linéaires.

À la même date, les approches algorithmiques de l'intelligence artificielle furent l'objet de désillusion, leurs applications ne répondant pas aux attentes. Cette désillusion motiva une réorientation des recherches en intelligence artificielle vers les réseaux de neurones (bien que ces réseaux concernent la perception artificielle plus que l'intelligence artificielle à proprement parler). La recherche fut relancée et l'industrie reprit quelque intérêt au neuronal (en particulier pour des applications comme le guidage de missiles de croisière). En 1985, c'est le système de rétro propagation du gradient de l'erreur qui est le sujet le plus débattu dans le domaine.

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non linéaires : le perceptron multicouche ne possède pas les défauts mis en évidence par *Minsky*. Proposé pour la première fois par *Werbos*, le Perceptron **Multi-Couche** apparaît en 1986 introduit par *Rumelhart*, et, simultanément, sous une appellation voisine, chez *Yann le Cun*. Ces systèmes reposent sur la rétro propagation du gradient de l'erreur dans des systèmes à plusieurs couches, chacune de type **Adaline** de *Bernard Widrow*, proche du Perceptron de *Rumelhart*.

Les réseaux de neurones ont par la suite connu un essor considérable, et ont fait partie des premiers systèmes à bénéficier de l'éclairage de la théorie de la régularisation statistique introduite par *Vladimir Vapnik* en Union Soviétique. Cette théorie, l'une des plus importantes du domaine des statistiques, permet d'anticiper, d'étudier et de réguler les phénomènes liés au sur-apprentissage. On peut ainsi réguler un système d'apprentissage pour qu'il arbitre au

mieux entre une modélisation pauvre (exemple : la moyenne) et une modélisation trop riche qui serait optimisée de façon illusoire sur un nombre d'exemples trop petit, et serait inopérant sur des exemples non encore appris, même proches des exemples appris. Le sur-apprentissage est une difficulté à laquelle doivent faire face tous les systèmes d'apprentissage par l'exemple, que ceux-ci utilisent des méthodes d'optimisation directe (e.g. régression linéaire), itératives (e.g. descente de gradient), ou itératives semi-directes (gradient conjugué, estimation-maximisation de Baum...) et que ceux-ci soient appliqués aux modèles statistiques classiques, aux modèles de Markov ou aux réseaux de neurones formels.

II.2.4 Utilisation [17], [15]

Grâce à leur capacité de généralisation, les réseaux de neurones sont généralement utilisés dans des problèmes de nature statistiques et perceptives, telles que la classification ou l'évaluation.

Il est à noter que le réseau de neurones ne fournit pas de règle exploitable par un humain. Le réseau reste une boîte noire qui fournit une réponse quand on lui présente une donnée, mais le réseau ne fournit pas de justification facile à lire sur la façon dont celle-ci est obtenue.

Les réseaux de neurones sont pratiquement utilisés, par exemple:

- pour la classification ; par exemple pour la classification d'espèces animales par espèce étant donnée une analyse ADN.
- reconnaissance de motif ; par exemple pour la Reconnaissance optique de caractères (OCR), et notamment par les banques pour vérifier le montant des chèques, par La Poste pour trier le courrier en fonction du code postal, etc. ; ou bien encore pour le déplacement automatisé de robots mobiles autonomes.
- approximation d'une fonction inconnue.
- modélisation accélérée d'une fonction connue mais très complexes à calculer avec exactitude ; par exemple certaines fonctions d'inversions utilisées pour décrypter les signaux de télédétection émis par les satellites et les transformer en données sur la surface de la mer.
- estimations boursières : apprentissage de la valeur d'une entreprise en fonction des indices disponibles : bénéfices, endettements à long et court terme, chiffre d'affaires,

carnet de commandes, indications techniques de conjoncture. Ce type d'application ne pose pas en général de problème d'apprentissage.

- tentatives de prédiction sur la périodicité des cours boursiers. Ce type de prédiction est très contesté pour deux raisons, l'une étant que le cours d'une action n'a jamais été périodique (le marché anticipe en effet largement les hausses comme les baisses prévisibles), et l'autre que c'est l'avenir prévisible d'une entreprise qui détermine essentiellement le cours de son action, et en aucun cas son passé.
- modélisation de l'apprentissage et amélioration des techniques de l'enseignement.

II.2.5 Le neurone formel

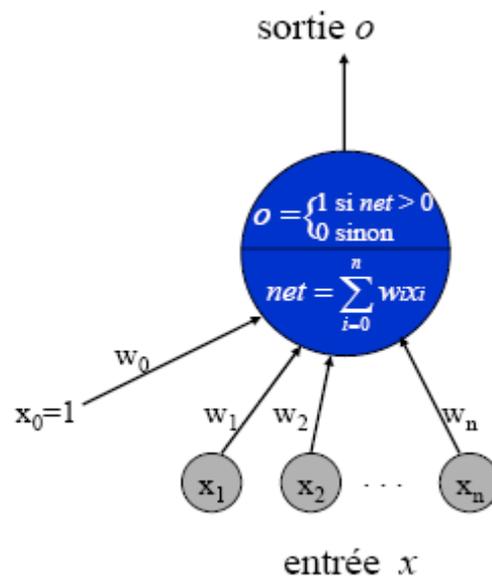


Figure 2.3 : Le modèle Mc Culloch et Pitts [13]

Cette figure montre la structure d'un neurone artificiel, appelé aussi PERCEPTRON. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associée un poids [15].

II.2.5.1 Fonctionnement [15]

On distingue deux phases. La première est habituellement le calcul de la somme pondérée des entrées selon l'expression suivante :

$$a = \sum w_i x_i - \theta$$

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones avals. Il existe de nombreuses formes possibles pour la fonction de transfert. Les plus courantes sont présentées sur la figure suivante. On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprise dans l'intervalle $[0, +1]$ (ou $[-1, +1]$).

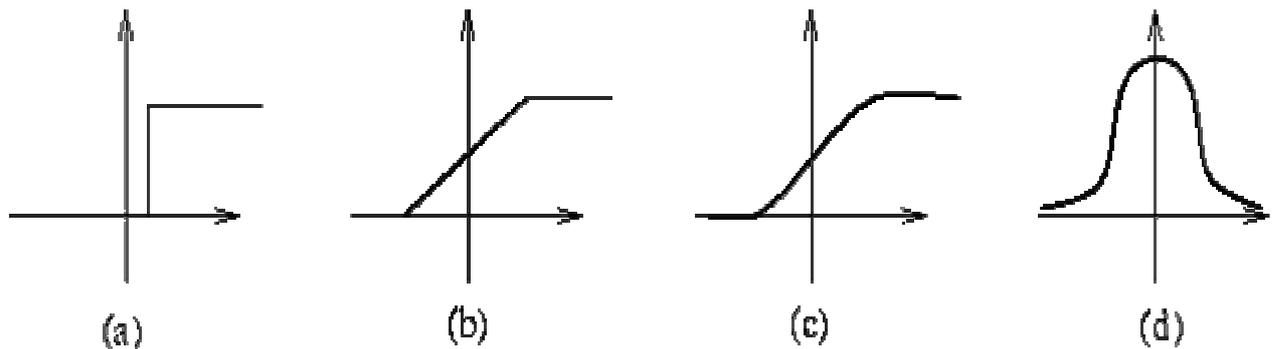


Figure 2.4 : Différents types de fonctions de transfert pour le neurone artificiel,

(a) : seuil (fonction de Heaviside)

(b) : linéaire par morceaux

(c) : sigmoïde $g(x) = (1 + e^{-\beta x})^{-1}$

(d) : gaussienne

Nous constatons que les équations décrivant le comportement des neurones artificiels n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons affaire à des modèles à temps discret, synchrones, dont le comportement des composants ne varie pas dans le temps.

II.2.6 Architecture des réseaux de neurones [14]

Un perceptron linéaire à seuil est bien adapté pour des échantillons linéairement séparables. Cependant, dans la plupart des problèmes réels, cette condition n'est pas réalisée. Un perceptron linéaire à seuil est constitué d'un seul neurone. On s'est très vite rendu compte qu'en combinant plusieurs neurones le pouvoir de calcul était augmenté. Par exemple, dans le cas des fonctions booléennes, il est facile de calculer le XOR en utilisant deux neurones linéaires à seuil. Cet exemple est présenté dans la figure suivante.

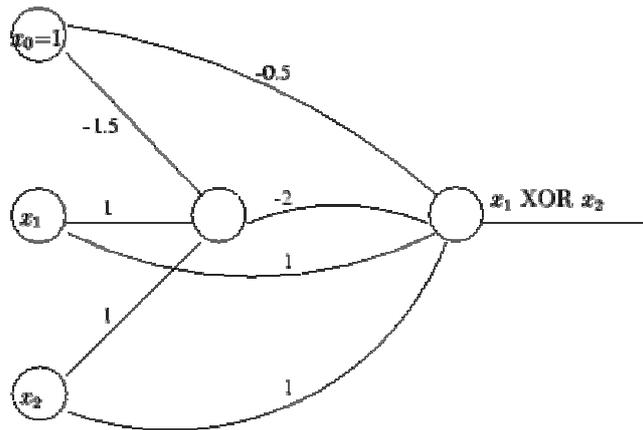


Figure 2.5 : Il suffit de rajouter un neurone intermédiaire entre la rétine et la cellule de décision pour pouvoir calculer le XOR [14]

La notion de perceptron multi-couches (PMC) a ainsi été définie. On considère une couche d'entrée qui correspond aux variables d'entrée, une couche de sorties, et un certain nombre de couches intermédiaires. Les liens n'existent qu'entre les cellules d'une couche avec les cellules de la couche suivante. Le XOR peut être calculé par un perceptron multicouches présenté dans la figure suivante en transformant légèrement le réseau présenté dans la figure précédente.

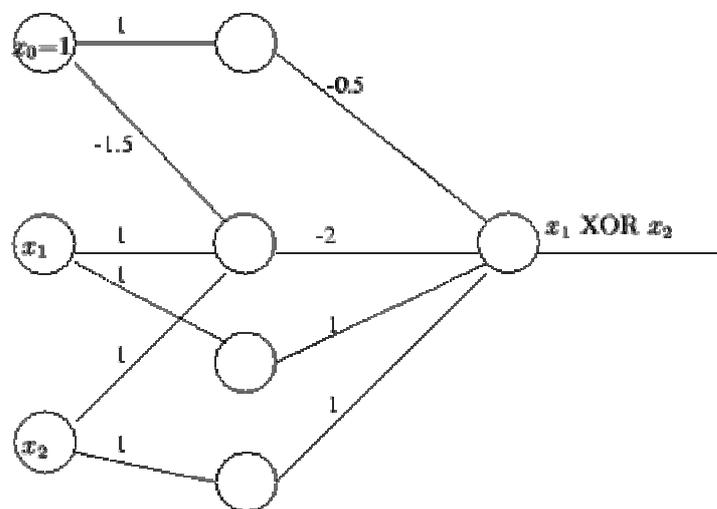


Figure 2.6 : PMC pour le XOR ; les liens avec poids nul ne sont pas représentés [14]

Définition [14]: Un réseau de neurones à couches cachées est défini par une architecture vérifiant les propriétés suivantes :

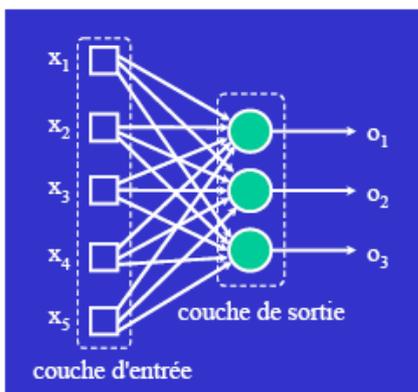
- les cellules sont réparties de façon exclusive dans des couches C_0, C_1, \dots, C_q ,

- la première couche C0 est la rétine composée des cellules d'entrée qui correspondent aux n variables d'entrée ; les couches C1, ..., Cq-1 sont les couches cachées ; la couche Cq est composée de la (ou les) cellule(s) de décision,
- Les entrées d'une cellule d'une couche Ci avec $i \geq 1$ sont toutes les cellules de la couche Ci-1 et aucune autre cellule.
- La dynamique du réseau est synchrone.

Le réseau présenté pour le calcul du XOR est un réseau à une couche cachée. L'architecture d'un réseau à couches cachées est sans rétroaction. Dans notre définition, nous avons supposé qu'une cellule avait pour entrée toutes les cellules de la couche précédente, ce peut être un sous-ensemble des cellules de la couche précédente. Ce qui est primordial dans la définition, c'est que les entrées appartiennent uniquement à la couche précédente, c'est-à-dire que la structure en couches est respectée et qu'il n'y a pas de rétroaction. Supposons que les cellules élémentaires soient des perceptrons linéaires à seuil, on parle alors de perceptrons multi-couches (PMC) linéaire à seuil. Soient n variables binaires, il est facile de montrer que le OU est calculable par un perceptron linéaire à seuil et que toute conjonction sur les littéraux définis à partir des n variables est calculable par un perceptron linéaire à seuil. Étant donné une fonction booléenne sur n variables, cette fonction peut être mise sous forme normale disjonctive, il suffit alors que chaque cellule de la couche cachée calcule une conjonction et que la cellule de sortie calcule la disjonction des résultats. Nous avons ainsi démontré que toute fonction booléenne peut être calculée par un PMC linéaire à seuil comprenant une seule couche cachée.

On distingue plusieurs structures de réseaux multicouches [13] :

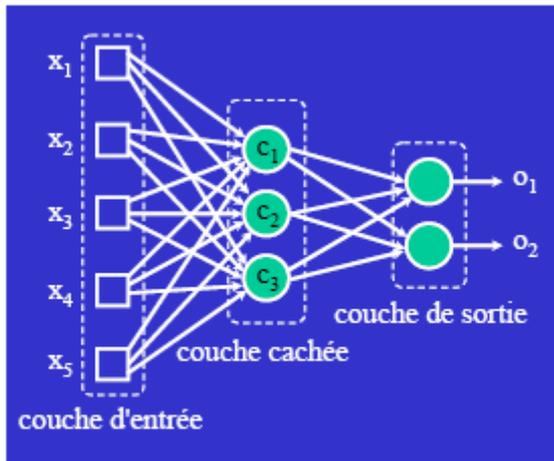
- Réseau acyclique ("feedforward") à une seule couche [13]



$$o_i = 1 \quad \text{si} \quad \sum_k w_{ik} \cdot x_k > 0$$

$$o_i = 0 \quad \text{sinon}$$

- Réseau acyclique multi-couches [13]



Couche cachée

$$c_j = 1 \quad \text{si} \quad \sum_k w_{jk} \cdot x_k > 0$$

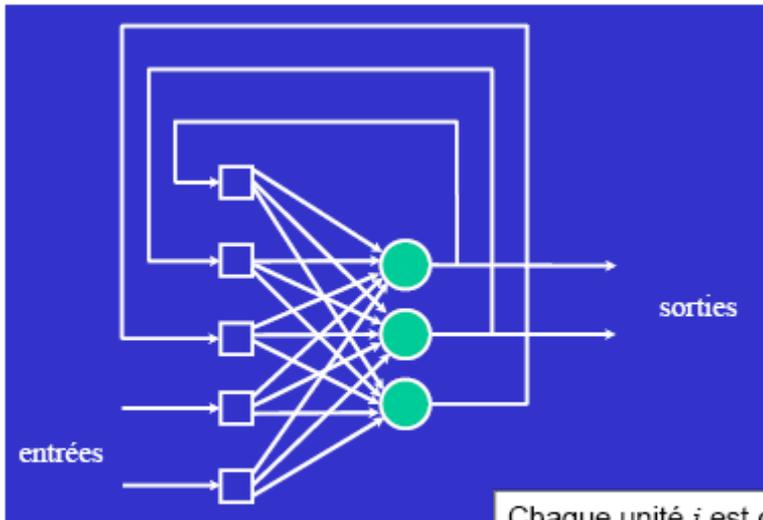
$$c_j = 0 \quad \text{sinon}$$

Couche de sortie

$$o_i = 1 \quad \text{si} \quad \sum_k w_{ik} \cdot c_k > 0$$

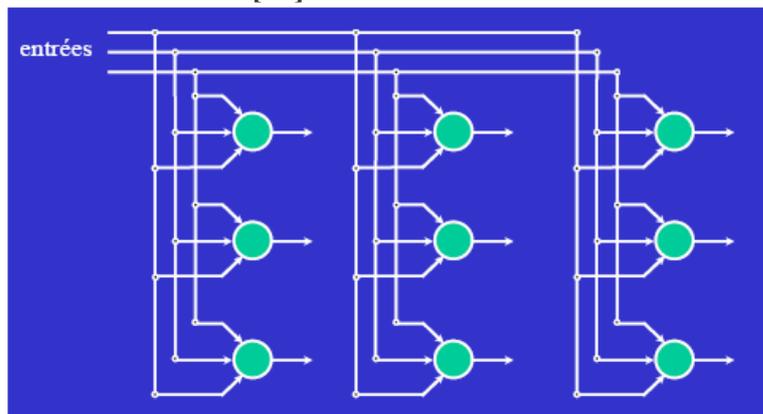
$$o_i = 0 \quad \text{sinon}$$

- Réseau récursif (réseau de Hopfield) [13]



Chaque unité i est connectée à chaque autre unité j par un poids w_{ij}
 les poids sont supposés symétriques: $w_{ij} = w_{ji}$

- Réseau "en treillis" [13]



Réseau en treillis 3x3 bi-dimensionnel

II.2.7 Notions sur l'apprentissage [15]

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux ; c'est la phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt. Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider à priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation.

Certains modèles de réseaux sont improprement dénommés à apprentissage permanent. Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, cependant on peut toujours distinguer une phase d'apprentissage (en fait de remise à jour du comportement) et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées. Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit **supervisé** ou **non supervisé**.

Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage **supervisé**, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage **non supervisé**. Remarquons cependant que les modèles à apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.

II.2.7.1 Types d'apprentissage [14]

II.2.7.1.1 apprentissage par correction d'erreurs

On initialise les poids du perceptron à des valeurs quelconques. A chaque fois que l'on présente un nouvel exemple, on ajuste les poids selon que le perceptron l'a correctement classé ou non. L'algorithme s'arrête lorsque tous les exemples ont été présentés sans modification d'aucun poids.

L'algorithme d'apprentissage par correction d'erreur du perceptron linéaire à seuil est [14]:

Entrée : un échantillon S de $R^n \times \{0,1\}$ ou $\{0,1\}^n \times \{0,1\}$
Initialisation aléatoire des poids w_i pour i entre 0 et n
Répéter
Prendre un exemple (x,c) dans S
Calculer la sortie o du perceptron pour l'entrée x
- - Mise à jour des poids - -
Pour i de 0 à n
 $w_i \leftarrow w_i + (c-o)x_i$
Fin pour
Fin Répéter
Sortie : Un perceptron P défini par (w_0, w_1, \dots, w_n)

La procédure d'apprentissage du perceptron est une procédure de *correction d'erreur* puisque les poids ne sont pas modifiés lorsque la sortie attendue c est égale à la sortie calculée o par le perceptron courant. Étudions les modifications sur les poids lorsque c diffère de o :

si $o=0$ et $c=1$, cela signifie que le perceptron n'a pas assez pris en compte les neurones actifs de l'entrée (c'est-à-dire les neurones ayant une entrée à 1) ; dans ce cas, $w_i \leftarrow w_i + x_i$; l'algorithme ajoute la valeur des entrées aux poids synaptiques (*renforcement*).

si $o=1$ et $c=0$, alors $w_i \leftarrow w_i - x_i$; l'algorithme retranche la valeur de la rétro aux poids synaptiques (*inhibition*).

Remarquons que, en phase de calcul, les constantes du perceptron sont les poids synaptiques alors que les variables sont les entrées. Tandis que, en phase d'apprentissage, ce sont les coefficients synaptiques qui sont variables alors que les entrées de l'échantillon S apparaissent comme des constantes.

II.2.7.1.2 Apprentissage par descente de gradient [14]

Plutôt que d'obtenir un perceptron qui classe correctement tous les exemples, il s'agira maintenant de calculer une erreur et d'essayer de minimiser cette erreur. Pour introduire cette notion d'erreur, on utilise des poids réels et on élimine la notion de seuil (ou d'entrée supplémentaire), ce qui signifie que la sortie sera égale au potentiel post-synaptique et sera donc réelle.

Définition [14] Un perceptron linéaire prend en entrée un vecteur x de n valeurs x_1, \dots, x_n et calcule une sortie o . Un perceptron est défini par la donnée d'un vecteur w de n constantes : les coefficients synaptiques w_1, \dots, w_n . La sortie o est définie par :

$$o = x.w = \sum w_i . x_i$$

L'erreur d'un perceptron P défini par $w=(w_1, \dots, w_n)$ sur un échantillon d'apprentissage S d'exemples (x^s, c^s) est définie en utilisant la fonction erreur quadratique par :

$$E(w) = 1/2 \sum_{x,c} (c - o)^2$$

où o est la sortie calculée par P sur l'entrée x. L'erreur mesure donc l'écart entre les sorties attendue et calculée sur l'échantillon complet. On remarque que $E(w) = 0$ si et seulement si le perceptron classe correctement l'échantillon complet.

On suppose S fixé, le problème est donc de déterminer un vecteur w qui minimise E(w). Une méthode qui permet de rechercher le minimum d'une fonction est d'utiliser la méthode du gradient.

Soit f une fonction d'une variable réelle à valeurs réelles, suffisamment dérivable dont on recherche un minimum. La méthode du gradient construit une suite x_n qui doit en principe s'approcher du minimum. Pour cela, on part d'une valeur quelconque x_0 et l'on construit la suite récurrente par : pour tout $n > 0$, $x_{n+1} = x_n + \Delta x_n$ avec $\Delta x_n = -\varepsilon f'(x_n)$ où ε est une valeur « bien » choisie [14]

Remarques :

1. le choix de ε est empirique,
2. si ε est trop petit, le nombre d'itérations peut être très élevé,
3. si ε est trop grand, les valeurs de la suite risquent d'osciller autour du minimum sans converger,
4. rien ne garantit que le minimum trouvé est un minimum global.

Voici à présent l'algorithme d'apprentissage par descente de gradient du perceptron linéaire :

Algorithme par descente de gradient [14]:
Entrée : un échantillon S de $R^n \times \{0,1\}$; ε
Initialisation aléatoire des poids w_i pour i entre 1 et n
Répéter
Pour tout i $\Delta w_i \leftarrow 0$ finPour
Pour tout exemple (x^s, c^s) de S
calculer la sortie o^s
Pour tout i $\Delta w_i \leftarrow \Delta w_i + \varepsilon (c^s - o^s) x_i^s$ finPour
finPour
Pour tout i $w_i \leftarrow w_i + \Delta w_i$ finPour
finRépéter
Sortie : Un perceptron P défini par (w_1, \dots, w_n)

La fonction erreur quadratique ne possède qu'un minimum (la surface est une paraboloïde).

L'algorithme précédent est assuré de converger, même si l'échantillon d'entrée n'est pas linéairement séparable, vers un minimum de la fonction erreur pour un η bien choisi suffisamment petit. Si η est trop grand, on risque d'osciller autour du minimum. Pour cette raison, une modification classique est de diminuer graduellement la valeur de η en fonction du nombre d'itérations. Le principal défaut est que la convergence peut être très lente et que chaque étape nécessite le calcul sur tout l'ensemble d'apprentissage.

II.2.7.1.2.1 Algorithme d'apprentissage de Widrow-Hoff [14]

Cet algorithme est une variante très utilisée de l'algorithme précédent. Au lieu de calculer les variations des poids en sommant sur tous les exemples de S , l'idée est de modifier les poids à chaque présentation d'exemple. La règle de modification des poids donnée devient :

$$\Delta w_i = \varepsilon (c - o) x_i$$

Cette règle est appelée règle delta, ou règle Adaline, ou encore règle de Widrow-Hoff d'après le nom de ses inventeurs. L'algorithme s'écrit alors :

Algorithme de Widrow-Hoff [14]:
Entrée : un échantillon S de $R^n \times \{0,1\}$; ε
Initialisation aléatoire des poids w_i pour i entre 1 et n
Répéter
Prendre un exemple (x,c) dans S
Calculer la sortie o du perceptron pour l'entrée x
-- Mise à jour des poids --
Pour i de 1 à n
 $w_i \leftarrow w_i + \varepsilon (c-o)x_i$
finpour
finRépéter
Sortie : Un perceptron P défini par (w_0, w_1, \dots, w_n)

En général, on parcourt l'échantillon dans un ordre prédéfini. Le critère d'arrêt généralement choisi tel que pour un passage complet de l'échantillon, toutes les modifications de poids soient en dessous d'un seuil prédéfini.

Au coefficient ε près dans la règle de modification des poids, on retrouve l'algorithme d'apprentissage par correction d'erreur. Pour l'algorithme de Widrow-Hoff, il y a correction chaque fois que la sortie totale (qui est un réel) est différente de la valeur attendue (égale à 0 ou 1). Ce n'est donc pas une méthode d'apprentissage par correction d'erreur puisqu'il y a modification du perceptron dans (presque) tous les cas. Rappelons également que l'algorithme par correction d'erreur produit en sortie un perceptron linéaire à seuil alors que l'algorithme

par descente de gradient produit un perceptron linéaire. L'avantage de l'algorithme de Widrow-Hoff par rapport à l'algorithme par correction d'erreur est que, même si l'échantillon d'entrée n'est pas linéairement séparable, l'algorithme va converger vers une solution << optimale >> (sous réserve du bon choix du paramètre ε). L'algorithme est, par conséquent, plus robuste au bruit.

L'algorithme de Widrow-Hoff s'écarte de l'algorithme du gradient sur un point important : on modifie les poids après présentation de *chaque* exemple en fonction de l'erreur locale et non de l'erreur globale. Rien ne prouve donc que la diminution de l'erreur en un point ne va pas être compensée par une augmentation de l'erreur pour les autres points. La justification empirique de cette manière de procéder est commune à toutes les méthodes adaptatives : le champ d'application des méthodes adaptatives est justement l'ensemble des problèmes pour lesquels des ajustements locaux vont finir par converger vers une solution globale. L'algorithme de Widrow-Hoff est très souvent utilisé en pratique et donne de bons résultats. La convergence est, en général, plus rapide que par la méthode du gradient. Il est fréquent pour cet algorithme de faire diminuer la valeur de ε en fonction du nombre d'itérations comme pour l'algorithme du gradient.

II.2.7.1.2.2 L'algorithme de rétropropagation du gradient [14]

La fonction sigmoïde de paramètre $k > 0$ est définie par :

$$\sigma(x) = \frac{e^x}{(1+e^x)} = \frac{1}{(1+e^{-x})}$$

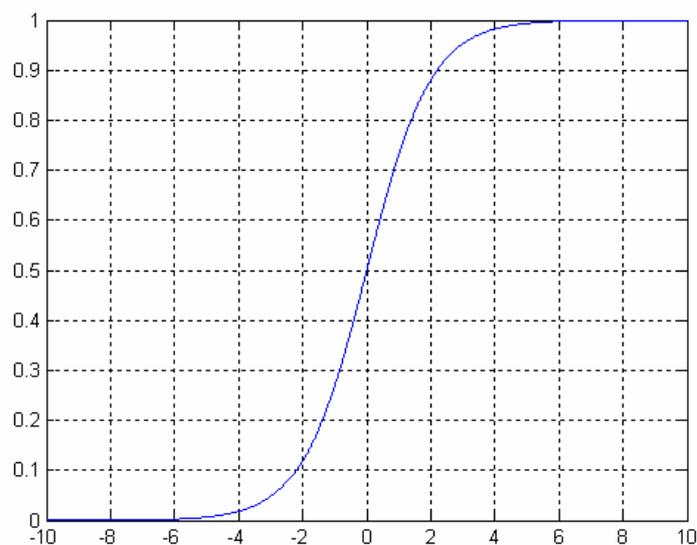


Figure 2.7 La fonction sigmoïde [14]

On peut remarquer que la dérivée de la fonction σ est simple à calculer :

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Il est essentiel que ce calcul soit simple car la dérivée de cette fonction sera utilisée dans la règle de mise à jour des poids par l'algorithme de rétro propagation du gradient.

Un perceptron multicouches (PMC) est un réseau de neurones à couches cachées avec les cellules sigmoïdes ainsi définies.

Le principe de l'algorithme est, comme dans le cas du perceptron linéaire, de minimiser une fonction d'erreur. Il s'agit ensuite de calculer la contribution à cette erreur de chacun des poids synaptiques. En effet, chacun des poids influe sur le neurone correspondant, mais, la modification pour ce neurone va influencer sur tous les neurones des couches suivantes. Ce problème est parfois désigné sous le nom de << Credit Assignment Problem >>.

Soit un PMC défini par une architecture à n entrées et à p sorties, soit w le vecteur des poids synaptiques associés à tous les liens du réseau. L'erreur du PMC sur un échantillon d'apprentissage S d'exemples (x, c) est définie par :

$$E(w) = 1/2 \sum_{x,c} \sum_{k=1}^p (c_k - o_k)^2$$

où o_k est la k -ième composante du vecteur de sortie o_s calculé par le PMC sur l'entrée x . L'erreur mesure donc l'écart entre les sorties attendue et calculée sur l'échantillon complet. On suppose S fixé, le problème est donc de déterminer un vecteur w qui minimise $E(w)$. Cependant, de la même façon que pour le perceptron avec la règle de Widrow-Hoff, plutôt que de chercher à minimiser l'erreur globale sur l'échantillon complet, on cherche à minimiser l'erreur sur chaque présentation individuelle d'exemple. L'erreur pour un exemple est :

$$E(w) = 1/2 \sum_{k=1}^p (c_k - o_k)^2$$

Pour écrire l'algorithme, nous allons simplifier quelques notations. Nous appelons la quantité, pour une cellule i de sortie, nous avons [14]:

$$\delta_i = o_i(1-o_i)(c_i-o_i)$$

pour une cellule i interne, nous avons :

$$\delta_i = o_i(1-o_i) \sum_{k \in \text{Succ}(i)} \delta_k w_{ki}$$

la modification du poids w_{ij} est alors définie par :

$$\Delta w_{ij} = \varepsilon x_{ij} \delta_i$$

On peut faire les remarques suivantes :

- la règle de modification des poids pour le perceptron linéaire est : $w_i \leftarrow w_i + \varepsilon (c-o)x_i$. Dans le cas du PMC, cette règle est : $w_{ij} \leftarrow w_{ij} + \varepsilon \delta_i x_{ij}$. Ces deux règles sont très similaires, le terme d'erreur $c-o$ est remplacé par un terme plus compliqué δ_i ,
- pour une cellule i de sortie, la quantité δ_i correspond à l'erreur usuelle c_i-o_i multipliée par la dérivée de la fonction sigmoïde,
- pour une cellule i interne, le calcul de δ_i dépend de la somme pondérée des erreurs des cellules de la couche suivante,
- après présentation de l'entrée x et calcul de la sortie o , le calcul des erreurs Δ_i sera effectué de la couche de sortie vers la couche d'entrée.

Il ne reste qu'à écrire l'algorithme.

Algorithme de rétropropagation du gradient[14] :

Entrée : un échantillon S de $R^n \times R^p$; ε
 un PMC avec une couche d'entrée C_0 , $q-1$ couches cachées C_1, \dots, C_{q-1} ,
 une couche de sortie C_q , n cellules.
 Initialisation aléatoire des poids w_i dans $[-0.5,0.5]$ pour i entre 1 et n

Répéter

Prendre un exemple $(x^{\rightarrow}, c^{\rightarrow})$ de S et calculer o^{\rightarrow}

-- calcul des δ_i par rétropropagation

Pour toute cellule de sortie i $\delta_i \leftarrow o_i(1-o_i)(c_i-o_i)$ **finPour**

Pour chaque couche de $q-1$ à 1

Pour chaque cellule i de la couche courante

$$\delta_i = o_i(1-o_i) \sum_{k \in \text{Succ}(i)} \delta_k w_{ki}$$

finPour

finPour

-- mise à jour des poids

Pour tout poids $w_{ij} \leftarrow w_{ij} + \varepsilon \delta_i x_{ij}$ **finPour**

finRépéter

Sortie : Un PMC défini par la structure initiale choisie et les w_{ij}

Remarques

L'algorithme de rétropropagation du gradient est une extension de l'algorithme de Widrow-Hoff. En effet, dans les deux cas, les poids sont mis à jour à chaque présentation d'exemple et donc on tend à minimiser l'erreur calculée pour chaque exemple et pas l'erreur globale.

La méthode donne de bons résultats pratiques. Dans la plupart des cas, on rencontre peu de problèmes dus aux minima locaux, peut être grâce au fait que l'on minimise une erreur locale. Cependant, les problèmes de minima locaux existent. Pour améliorer l'algorithme vis à vis de ce problème, mais aussi pour essayer d'améliorer la vitesse de convergence, une variante couramment utilisée consiste à pondérer la modification des poids en fonction du nombre

d'itérations déjà effectué. Plus formellement, on fixe une constante alpha $[0,1[$ appelée *moment* (momentum) ; soit t un compteur du nombre d'itérations de la boucle principale, la règle de modification des poids devient :

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}(t)$$
$$\Delta w_{ij}(t) = \varepsilon \delta_i x_{ij} + \alpha \Delta w_{ij}(t-1)$$

L'intérêt de cette règle est de prendre en compte les modifications antérieures des poids dans le but d'éviter des oscillations perpétuelles.

Le critère d'arrêt n'est pas précisé dans l'algorithme. Ce critère peut être : << arrêter dès que l'erreur estimée passe sous un seuil prédéfini >>. On retrouve alors le problème toujours présent en apprentissage de la sur-spécialisation. En effet, le critère d'arrêt dépend de l'erreur observée mesurée sur l'ensemble d'apprentissage et non de l'erreur réelle. Les méthodes pour palier à ce problème sont identiques à celles utilisées pour les arbres de décision. Par exemple, utiliser un ensemble test, quand c'est possible, pour estimer l'erreur réelle. Ou encore, utiliser des techniques d'élagage qui tentent de diminuer la taille du réseau.

Le mode de présentation des exemples est également absent de notre algorithme. En règle générale, on choisit une présentation équitable.

Le choix de l'architecture initiale du réseau reste un problème difficile. Ce choix peut être fait par l'expérience. Des méthodes dites << autoconstructives >> existent : il s'agit d'ajouter des cellules au cours de l'apprentissage pour que l'apprentissage se fasse bien. Mais ces méthodes rencontrent souvent le problème de la sur-spécialisation. L'architecture peut aussi être choisie à l'aide de méthodes basées sur les algorithmes génétiques.

Le modèle et l'architecture étant choisie, un problème est également de choisir les bonnes valeurs pour les paramètres . Pour cela, on découpe l'ensemble d'apprentissage en un ensemble d'apprentissage, un ensemble de validation et un ensemble test. Lors de la phase d'apprentissage, on arrête périodiquement l'apprentissage, on estime l'erreur réelle sur l'ensemble de validation, on met à jour les paramètres en fonction de la variation de cette erreur estimée. L'ensemble test sert à estimer l'erreur réelle à la fin de l'apprentissage.

L'algorithme de rétropropagation du gradient a pu être étendu à certaines classes de réseaux récurrents pour lesquels il y a rétroaction. Ces réseaux sont très utiles pour la prédiction de séries temporelles.

Enfin, terminons par une dernière remarque sur la puissance de représentation des réseaux multi-couches. Nous avons vu que toute fonction booléenne peut être calculée par un PMC linéaire à seuil comprenant une seule couche cachée dont on rappelle qu'elle peut être de taille

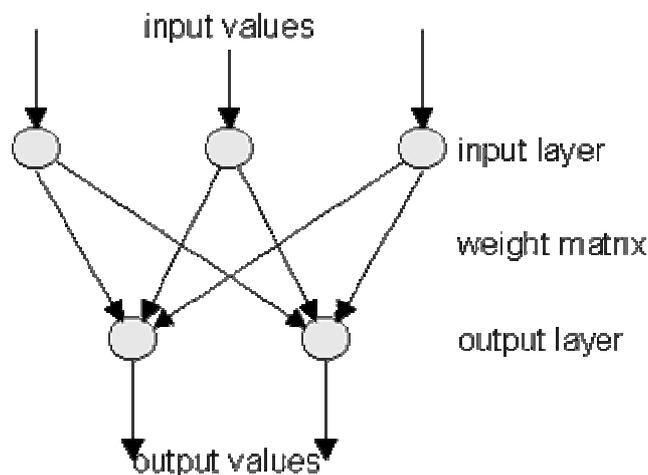
exponentielle. Ce résultat a été généralisé par Hornik en 1989 au cas des fonctions réelles et des PMC : *la plupart des fonctions numériques peuvent être approximées avec une précision arbitraire par des réseaux à une seule couche cachée*. Mais comme dans le cas booléen, cette couche cachée peut être démesurément grande et le théorème de Hornik est essentiellement un résultat *théorique* sur l'expressivité des réseaux multi-couches.

II.2.7.2 Application des algorithmes d'apprentissage sur les RNA [18]:

II.2.7.2.1 Perceptron (F. Rosenblatt, 1958)

C'est un réseau de neurones très simple avec deux couches de neurones acceptant seulement des valeurs d'entrées et de sorties binaires. Le procédé d'apprentissage est supervisé et le réseau est capable de résoudre des opérations logiques : simple somme, AND ou OR. La méthode d'apprentissage est supervisée. Ce qui veut dire que les sorties du réseau sont comparées avec des sorties cibles (patrons) désirées.

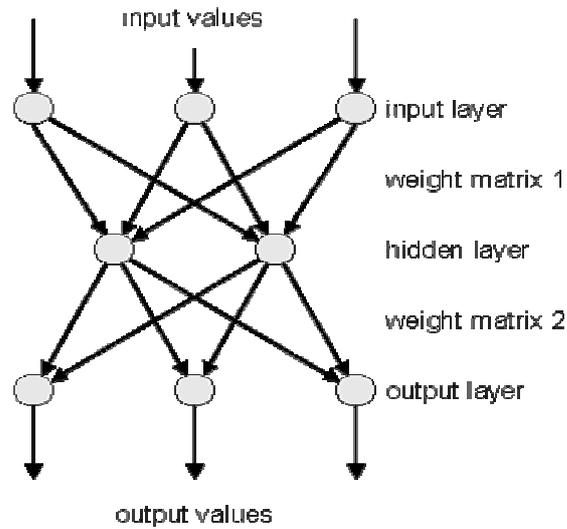
Il est aussi utilisé pour la classification de patrons. Les opérations logiques plus compliquées (comme un problème XOR) ne peuvent être résolues par un Perceptron. L'algorithme d'apprentissage est la règle d'apprentissage de Hebb qui affecte le changement des poids en multipliant l'entrée d'un neurone par sa sortie et le taux d'apprentissage du réseau.



II.2.7.2.2 Perceptron à niveaux multiples (Minsky et S. Papert, 1969) [18]

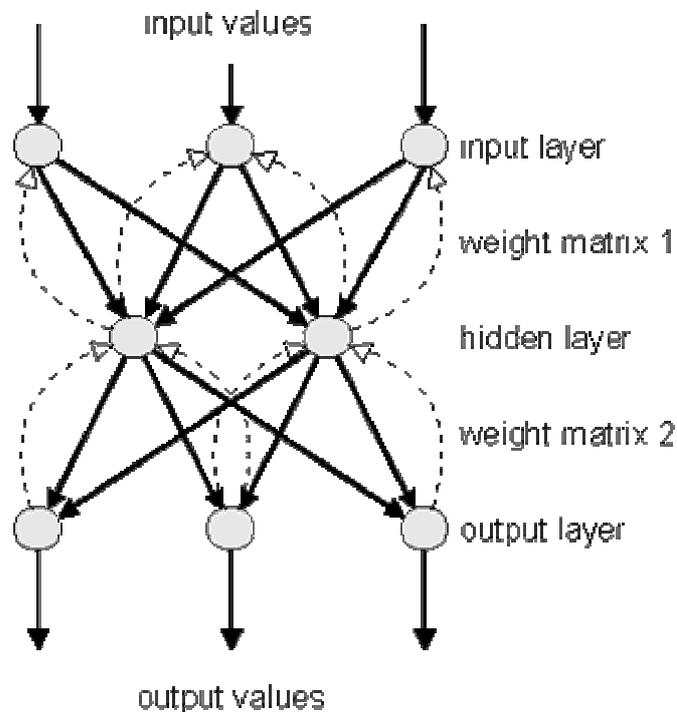
C'est une extension du Perceptron qui a un ou plusieurs sous-couches cachées entre ces couches d'entrées et de sortie. Du à sa structure étendue, un Perceptron à niveau multiples est capable de résoudre toute opération logique incluant le problème XOR. La méthode d'apprentissage est supervisée. L'algorithme d'apprentissage est la règle d'apprentissage delta

qui affecte le changement des poids en multipliant l'entrée d'un neurone avec la différence de la sortie et de la sortie désirée et le taux d'apprentissage du réseau



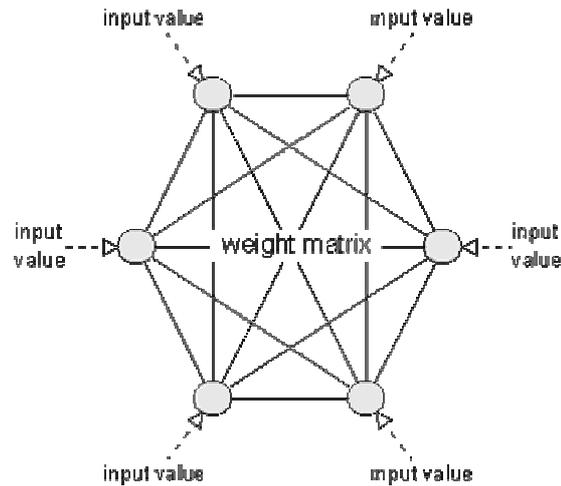
II.2.7.2.3 Rétro propagation (G.E. Hinton, E. Rumelhart et R.J. Williams, 1986) [18]

Ce type d'apprentissage est l'un des plus puissant . Il y a la même structure que le Perceptron à niveau multiples et utilise l'algorithme de rétro propagation de l'erreur. La méthode d'apprentissage est supervisée.



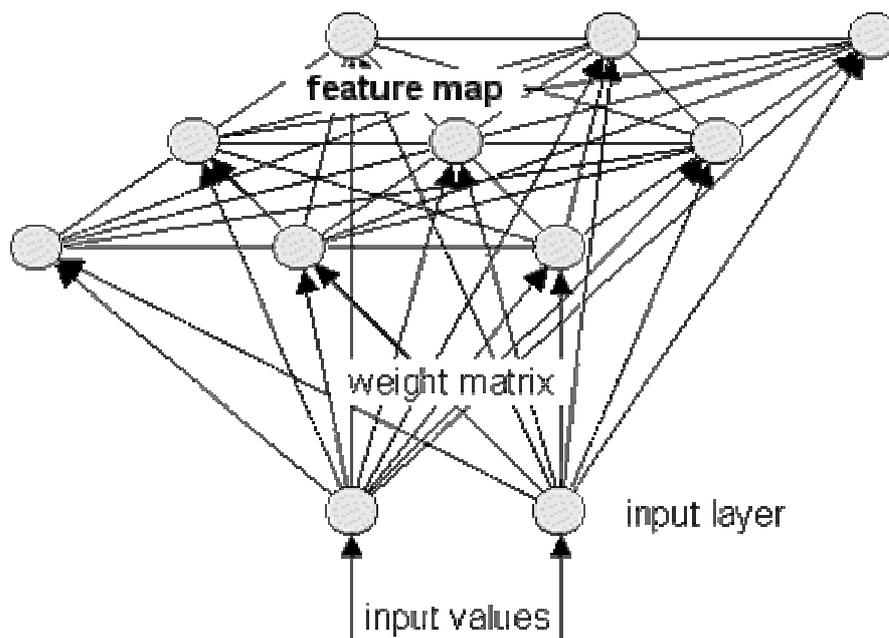
II.2.7.2.4 Hopfield (J.J. Hopfield, 1982) [18]

Ce type de réseau est entièrement connecté. Il n'y a aucune différenciation entre les neurones d'entrées et de sorties. L'application principale d'un réseau Hopfield est l'entrepôt et la reconnaissance des patrons, e.g. l'optimisation des images. La méthode d'apprentissage est non supervisée. L'algorithme d'apprentissage employé est la règle d'apprentissage "simulated annealing" ou parfois delta.



II.2.7.2.5 Kohonen (Teuvo Kohonen - University of Helsinki, 1982) [18]

C'est probablement le type de réseau de neurones le plus proche du cerveau humain. Le noyau de ce réseau est la carte où des neurones s'organisent eux-mêmes par rapport à certaines données à l'entrée. Ce type de réseau utilise l'anticipation (feed forward) ainsi que la rétroaction (feedback). Il est utilisé dans le codage d'informations.



II.2.7.2.6 Différences temporelles [18]

Ce type est l'un des rares algorithmes de réseaux de neurones modélisant des processus dépendant du temps (non stationnaires) . La différence principale entre l'algorithme de différences temporelles et de rétro propagation est que ce dernier est capable d'apprendre des fonctions orientées but, où la sortie éventuelle de chaque action du réseau n'est pas connu avant plusieurs étapes . Le type d'entraînement est basé sur la notion de récompense/pénalité, et apprentissage se fait par renforcement.

II.2.8 Conclusion

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont des approximateurs universels parcimonieux.

La propriété d'approximation peut être énoncée de la manière suivante : toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire. Cette propriété de parcimonie est précieuse dans les applications industrielles.

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux, cependant il existe plusieurs algorithmes utilisés pour faire l'apprentissage des réseaux multicouches, les performances des algorithmes d'apprentissage sont discutables suivant les applications en perspectives, dans la partie suivante on va énoncer les bases de la logique floue ; qui va être combiné avec les RNA dans la troisième partie de ce chapitre.

II.3 Bases de la logique floue et du réglage flou

II.3.1 Introduction [12]

La logique floue est une forme de logique utilisée dans quelques systèmes experts et d'autres applications d'intelligence artificielle dans lesquels les variables peuvent avoir des degrés d'exactitude ou de fausseté représentée par une gamme des valeurs entre 1 (vrai) et 0 (faux). Avec la logique floue, les résultats d'une opération peuvent être exprimés comme probabilité plutôt que comme certitude. Par exemple, en plus d'être vrais ou faux, des résultats pourraient avoir des significations telles que probablement vraies, et probablement fausses.

Elle s'appuie sur la théorie mathématique des sous-ensembles flous. Cette théorie, introduite par Pr L.Zadeh en 1965, est une extension de la théorie des ensembles classiques pour la prise en compte d'ensembles définis de façon imprécise. C'est une théorie formelle et mathématique dans le sens où , en partant du concept de fonction d'appartenance pour modéliser la définition d'un sous-ensemble d'un univers donné, on élabore un modèle complet de propriétés et de définitions formelles. On a aussi montré que cette théorie des sous-ensembles flous se réduit effectivement à la théorie des sous-ensembles classiques dans le cas où les fonctions d'appartenance considérées prennent des valeurs binaires ($\{0,1\}$).

Elle présente aussi l'intérêt d'être plus facile et meilleur marché à implémenter qu'une logique probabiliste, bien que cette dernière seule soit stricto sensu cohérente (voir Théorème de Cox-Jaynes).

II.3.2 Bref historique

Voici quelques repères historiques permettant de situer dans le temps le développement de la logique floue et ses applications au réglage:

- 1965 Le Prof. L. A. Zadeh de l'Université de Berkeley (Californie) pose les bases théoriques de la logique floue.
- 1973 L. A. Zadeh propose d'appliquer la logique floue aux problèmes de réglage.
- 1974 Première application du réglage par la logique floue appliquée à une turbine à vapeur. Suivie en 1980 par une application sur un four à ciment et en 1983 sur un épurateur d'eau.
- 1985 Premiers produits industriels (Japon) utilisant le principe de la logique floue appliqué à des problèmes de réglage et de commande. Développement de processeurs dédiés à des applications de réglage par la logique floue.

II.3.3 Principes [19]

II.3.3.1 Ensemble flou

Définition 1 [19]: Soit X un ensemble. Un sous-ensemble flou A de X est défini par une fonction d'appartenance sur x à valeurs dans l'intervalle [0,1].

Remarque : la fonction d'appartenance peut être fixée arbitrairement. Un des problèmes des applications pratiques est de la définir, on fait généralement appel à des données statistiques ou à l'avis d'un expert.

La notion de sous-ensemble flou englobe celle de sous-ensemble classique pour laquelle f_a (fonction d'appartenance) est la fonction indicatrice .

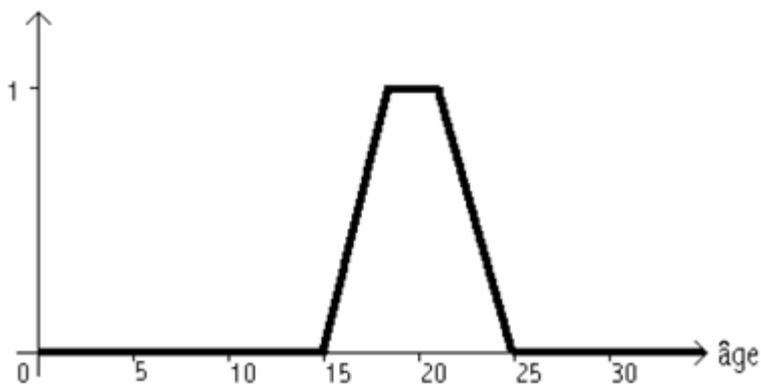
Définition 2 [19]: si A et B sont deux ensembles, tels que A est inclus dans B, nous appelons "fonction indicatrice de A (relativement à B), la fonction 1_a définie dans $\{0,1\}$, et telle que :

$$1_a(x) = 1 \quad \text{si } x \text{ est dans } A$$

$$1_a(x) = 0 \quad \text{si } x \text{ n'est pas dans } A$$

Les fonctions indicatrices sont souvent des intermédiaires techniques très pratiques.

Exemple : une fonction caractéristique possible pour définir le sous-ensemble flou "avoir une vingtaine d'années"



Les notions suivantes sont caractéristiques de A :

Définitions :

D1. Support de A : $supp(A) = \{x \in X, f_A(x) \neq 0\}$

D2. Hauteur de A : $h(A) = \sup_{x \in X} f_A(x)$

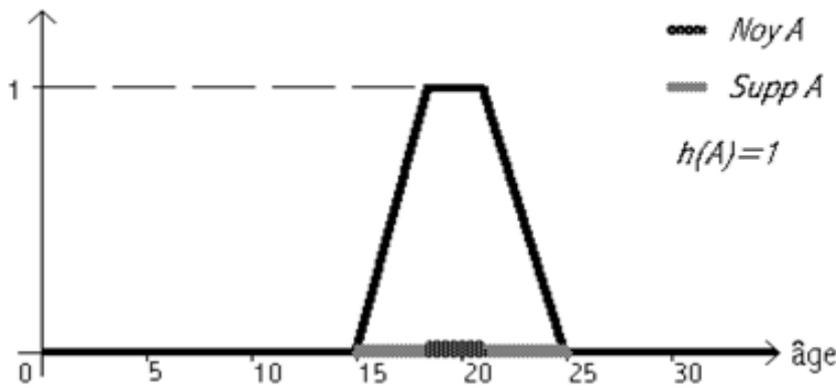
D3. A est dit normalisé si $h(A) = 1$

Remarque Les sous-ensembles flous considérés seront tous supposés normalisés, de hauteur égale à 1.

D4. Noyau de A : $noy(A) = \{x \in X, f_A(x) = 0\}$

D5. Cardinalité de A : $|A| = \sum_{x \in X} f_A(x)$

Exemple : avec le cas de la figure précédente :



D6. Si A et B sont deux sous-ensembles flous de l'ensemble X, nous disons que :

. A est "plus spécifique" que B si $noy(A) = noy(B)$ et $supp(A) \subset supp(B)$

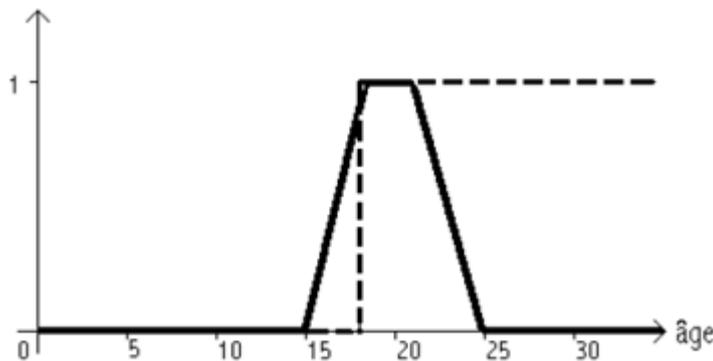
D7. Il y a égalité entre deux sous-ensembles flous si et seulement si $\forall x \in X, f_A(x) = f_B(x)$

D8. Il y a inclusion si et seulement si $\forall x \in X, f_A(x) \leq f_B(x)$

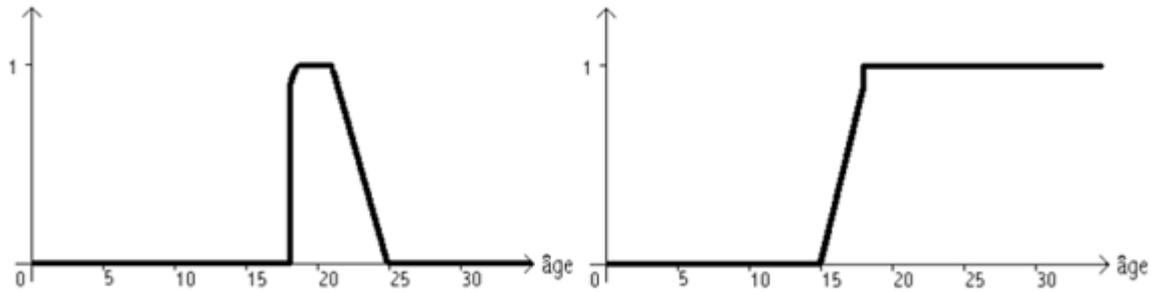
D9. L'intersection $A \cap B$ est définie par $f_{A \cap B} : x \rightarrow \min(f_A(x), f_B(x))$

D10. L'union $A \cup B$ est définie par $f_{A \cup B} : x \rightarrow \max(f_A(x), f_B(x))$

Exemple : reprenons le cas déjà envisagé. Nous considérons les personnes ayant une "vingtaine d'années", et celles "ayant la majorité" (en pointillés sur la figure : nous considérons celle-ci comme un sous ensemble non-flou) :



Selon les définitions de l'intersection ("ET logique" ou multiplication logique selon l'algèbre de Boole) et de l'union ("OU logique" ou addition logique selon l'algèbre de Boole), nous pouvons caractériser les sous-ensembles flous correspondant aux personnes "ayant une vingtaine d'années et la majorité" (à droite dans la figure ci-dessous) ainsi que celui des personnes "ayant une vingtaine d'années ou la majorité" (à gauche dans la figure ci-dessous) :



II.3.3.2 Opérations sur les ensembles flous :

Les opérations d'union, d'intersection et de complémentation dans les ensembles flous sont définies à l'aide de leur fonction d'appartenance.

La disjonction floue, 'OU' ou union :

La fonction d'appartenance $\mu_{A \cup B}$ est définie, pour tout $u \in U$, par :

Mamdani : $\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$

Sugeno : $\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u) \cdot \mu_B(u)$

La conjonction floue, 'ET' ou intersection :

La fonction d'appartenance $\mu_{A \cap B}$ est définie, pour tout $u \in U$, par :

Mamdani : $\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$

Sugeno : $\mu_{A \cap B}(u) = \mu_A(u) \cdot \mu_B(u)$

La complémentation, négation ou inverse

La fonction d'appartenance $\mu_{\bar{A}}$ du complément d'un ensemble A est définie, pour tout $u \in U$, dans les deux cas par

$$\mu_{\bar{A}} = 1 - \mu_A(u)$$

Produit cartésien

Si F_1, F_2, \dots, F_n sont des ensembles flous dans U_1, U_2, \dots, U_n respectivement, le produit cartésien de F_1, F_2, \dots, F_n est un ensemble flou dans l'espace produit $U_1 \times U_2 \times \dots \times U_n$ ayant pour fonction d'appartenance :

$$\mu_{F_1, \dots, F_n}(u_1, u_2, \dots, u_n) = \min\{\mu_{F_1}(u_1), \mu_{F_2}(u_2), \dots, \mu_{F_n}(u_n)\}$$

ou bien,

$$\mu_{F_1, \dots, F_n}(u_1, u_2, \dots, u_n) = \prod_{i=1}^n \mu_{F_i}(u_i)$$

Relation floue

Une relation floue d'ordre n est un ensemble flou dans $U_1 \times U_2 \times \dots \times U_n$ et est exprimé comme

$$R_{U_1 * U_2 * \dots * U_n} = \{(u_1, u_2, \dots, u_n), \mu_R(u_1, u_2, \dots, u_n) \mid (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n\}$$

II.3.3.3 Règles d'inférence

Une règle d'inférence est une règle qui relie les variables floues d'entrée d'un système aux variables floues de sortie de ce système.

Elle se présente ainsi:

Si condition 1 et/ou condition 2 ... et/ou condition n Alors action sur les sorties.

Avec deux entrées et une sortie de commande, on aura par exemple : \rightarrow Si a est A et/ou b est B Alors u est C.

Où a, b, et u sont des variables linguistiques représentant deux variables d'état du système et une variable de commande (sortie du régulateur) ; A, B et C sont les valeurs linguistiques des différentes variables.

Pour définir ces règles, on fait appel à une bonne connaissance du comportement du système (c'est le rôle de l'expert de trouver des règles adaptées au but recherché). Ces règles sont par la suite mémorisées dans l'organe de commande.

On peut représenter nos règles sous forme d'un tableau. Par exemple, si on a deux entrées et une sortie, ce tableau est de dimension deux.

Opérateur	Degré d'appartenance du résultat
Et	Minimum
	Produit
Ou	Maximum
	Valeur moyenne
Non	Complément à 1

Les opérations "minimum" et "maximum" présentent l'avantage de la simplicité lors du calcul ; par contre, elles privilégient l'une des deux variables. Les opérations de "produit" et "valeur moyenne" sont plus difficiles à calculer mais elles permettent d'obtenir un résultat qui tient compte des valeurs des deux variables.

II.3.3.4 Structure d'un régulateur flou [1]

La résolution d'un problème par la logique floue comprend trois étapes :

- 1- la quantification 'floue' des entrées du système
- 2- l'établissement des règles liant les sorties aux entrées à base des règles floues
- 3- la combinaison de ces règles pour la génération des sorties floues

La conception d'un régulateur flou doit donc passer par plusieurs étapes illustrées sur la figure de la page suivante :

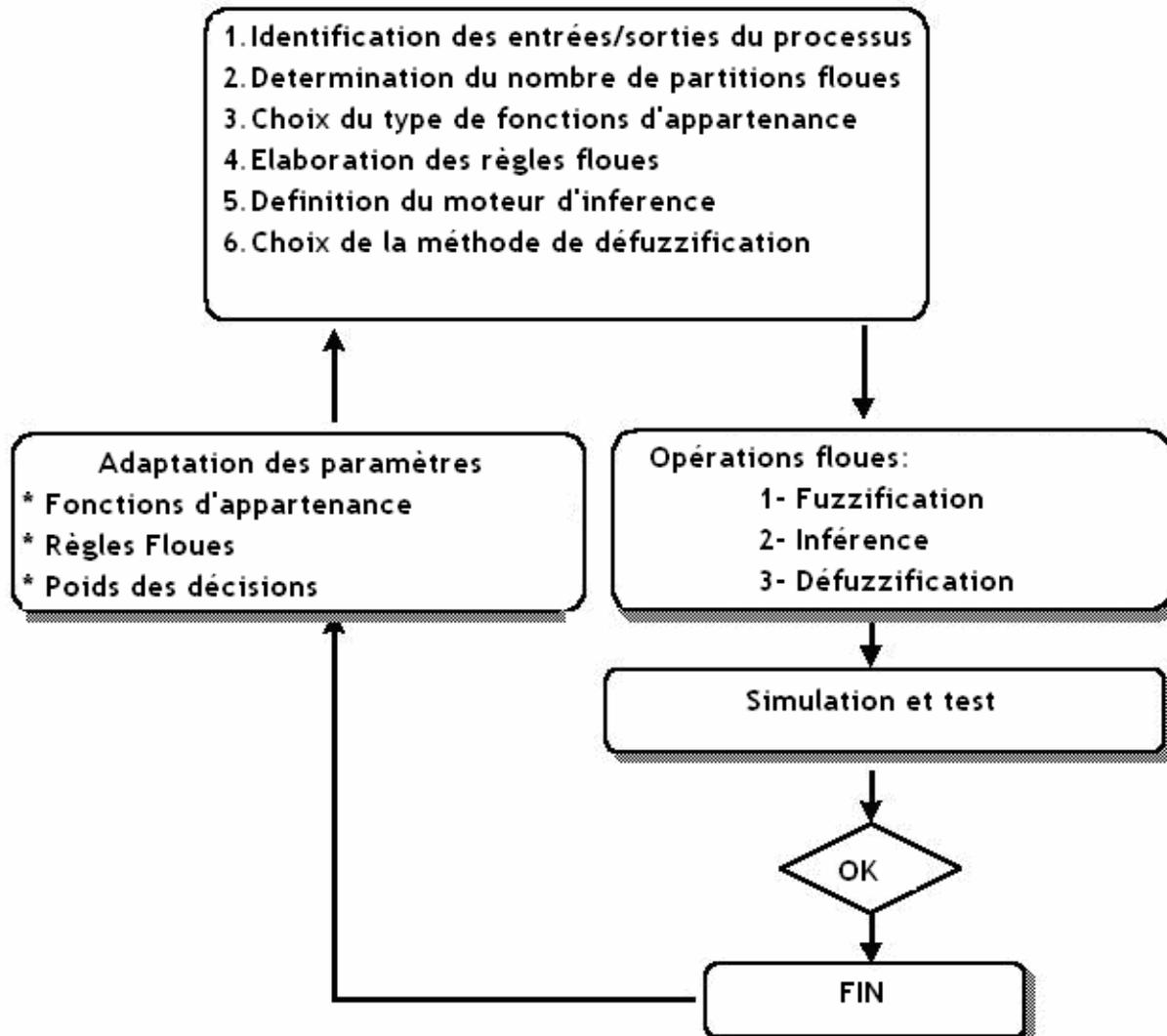


Figure 2.8: conception d'un régulateur Flou

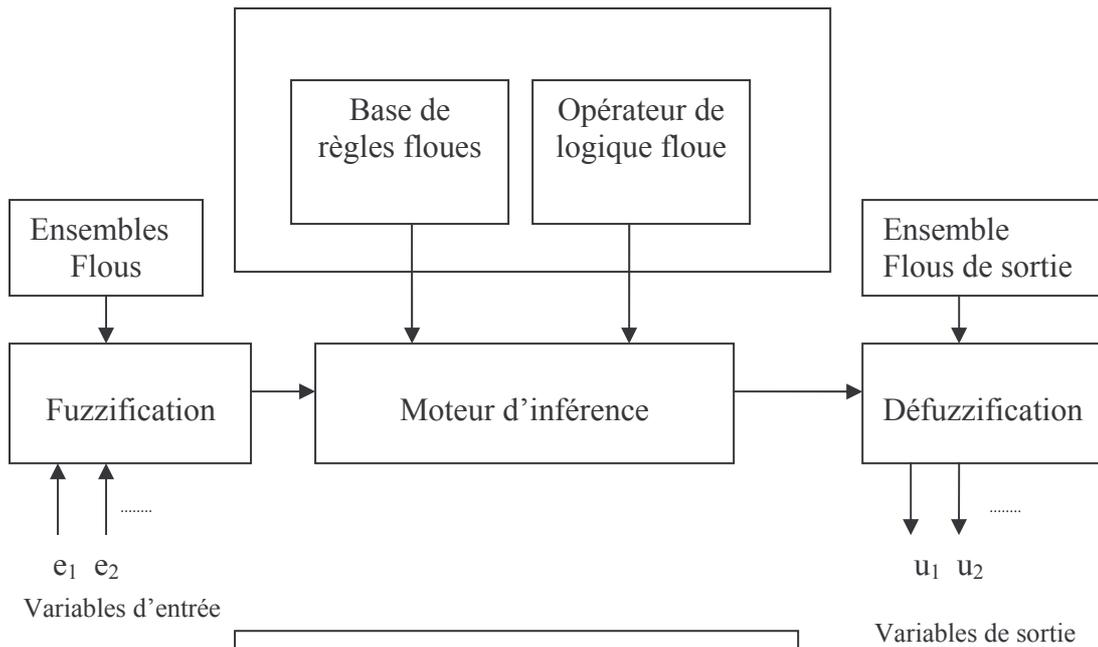


Figure 2.9 :
Structure d'un régulateur flou [20]

II.3.3.4.1 Principe

- 1- Procéder à la partition en sous ensembles flous des différents univers de discours que le système impose, et transformer les variables réelles en variables floues (fuzzification)
- 2- déterminer la base de règle qui va caractériser le fonctionnement désiré du système.
- 3- utiliser les variables floues dans un mécanisme d'inférence qui crée et détermine les variables floues de sortie, en utilisant des opérations sur les fonctions d'appartenance
- 4- Effectuer la défuzzification qui consiste à extraire une valeur réelle de sortie de la fonction d'appartenance du sous-ensemble flou de sortie établi par le mécanisme d'inférence

II.3.3.4.1.1 Interface de fuzzification

Un opérateur de fuzzification convertit une valeur numérique en une valeur floue (degré d'appartenance)

Le sous-ensemble flou U_0 doit être représenté par un fait imprécis. On utilise alors une méthode de fuzzification qui associe à la variable mesurée u_0 une fonction d'appartenance :

$$\mu_{u_0}(u) = \max \left\{ 0; \frac{|u - u_0|}{\varepsilon} \right\}$$

dont la représentation est sous forme de triangle de base ε fonction de l'importance relative des erreurs de mesures. Plus elle sont importantes plus la mesure de la variable u_0 devient imprécise, et donc plus la base du triangle doit s'élargir

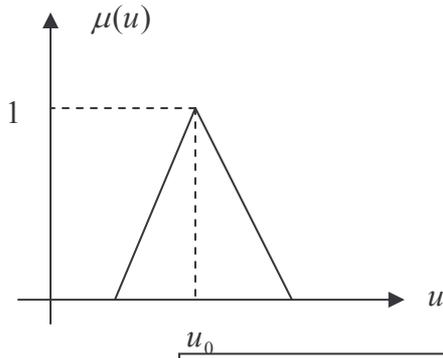


Figure 2.10: Fonction d'appartenance triangulaire

Ce triangle isocèle est utilisée dans le cas d'un fait incertain tel que u est à peu près égale à u_0 . dans le cas d'une affirmation de u est à peu près compris entre u_1 et u_2 , la forme de la fonction d'appartenance est trapézoïdale

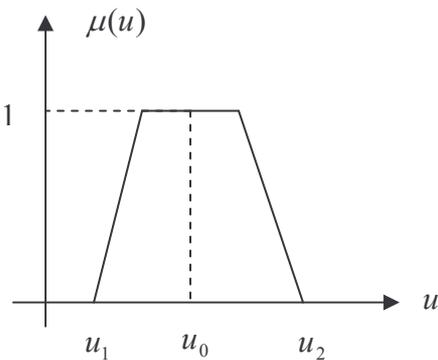


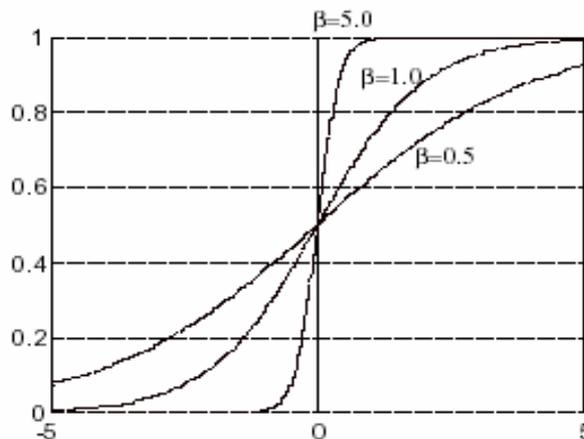
Figure 2.11 : Fonction d'appartenance trapézoïdale [20]

la forme sigmoïde et couramment la plus utilisée et est donnée par

$$\mu(u) = \frac{1}{1 + \exp(-\beta u)}$$

$$\varphi(u) = \frac{1}{1 + \exp(-\beta u)}$$

Figure 2.12: Fonction d'appartenance sigmoïdale



Partition floue

La partition floue d'un univers de discours U consiste à définir n sous-ensembles flous F_i de façon à recouvrir U . C'est-à-dire que pour tout élément u de U , il faut assurer une appartenance minimale ε à l'union des F_i :

$$\bigcup F_i \supseteq U_\varepsilon = \{u \in U; \mu_{U_\varepsilon}(u) = \varepsilon\}$$

En terme de fonction d'appartenance :

$$\forall u \in U, \mu_{F_1}(u) \cup \mu_{F_2}(u) \cup \dots \cup \mu_{F_n}(u) \geq \varepsilon$$

L'ensemble de sous-ensembles flous à définir dans une partition d'un univers de discours est fixé par l'expert.

Les ensembles flous (termes linguistiques) ont habituellement un sens tel :

<i>NG</i>	Négatif grand
<i>NM</i>	Négatif moyen
<i>NP</i>	Négatif petit
<i>EZ</i>	Environ zéro
<i>PP</i>	Positif petit
<i>PM</i>	Positif moyen
<i>PG</i>	Positif grand

Les prédicats « petit », « moyen », « grand » ou « négatif », « positif », « proche de zéro » servent à la régulation, les autres servent à la poursuite.

En contrôle classique cinq prédicats peuvent suffire, dans le cas extrême on peut aller jusqu'à sept.

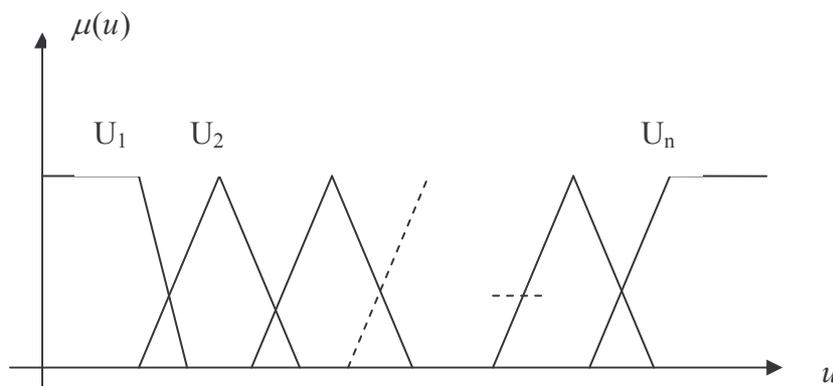


Figure 2.13
Univers de discours U

Base de règles [20]

Les relations entre classes d'évènement possibles en entrée et des commandes correspondantes sont caractérisées par une base de règle « Si...Alors... ».

Par conséquent, si l'on considère n univers de discours U_i pour les prémisses de règles floues et si pour chaque univers U_i on définit une partition en m_i sous-ensembles flous, le nombre maximum de règles r_{\max} est de

$$r_{\max} = \prod_{i=1}^n m_i$$

Le nombre de règles définies par l'expert peut être inférieur à r_{\max} , c'est le cas s'il existe des configurations impossibles à obtenir

Le nombre de sous-ensembles flous, définissant la partition de l'univers de discours de la commande, n'est pas forcément égale au nombre de règles. Des configurations différentes peuvent aboutir aux mêmes conclusions

L'implication floue

L'implication floue la plus utilisée en commande floue est celle définie par Mamdani, donnée par

$$\forall u_1 \in U_1, \forall u_2 \in U_2, R(u_1, u_2) = \min(\mu_1(u_1), \mu_2(u_2))$$

Le raisonnement approximatif [20]

Concept de base : la représentation de propositions par des formules affectant des ensembles flous comme valeurs aux variables.

Soient deux variables $x \in X$ et $y \in Y$, et une relation de cause à effet entre x et y , $y = f(x)$

Alors, on peut effectuer l'inférence :

Prémisse $y = f(x)$

Fait $x = x'$

Conséquence $y = f(x')$

La plupart du temps, on ne connaît le lien de cause à effet f entre x et y qu'en certaines valeurs x particulières. On a une base de règle :

$$R_i : \text{si } x = x_i \text{ alors } y = y_i, \quad i = \overline{1, n}$$

Il faut alors, connaissant $x' \in X$, trouver $y' \in Y$ correspondant à x' , conformément à la base de règles.

Le problème de base du raisonnement approximatif est de trouver la fonction d'appartenance de la conséquence C d'une base de règles $R_i, i = \overline{1, n}$, et le fait « x est x' »

II.3.3.4.1.2 Mécanisme d'inférence

C'est l'application du raisonnement approximatif. Il permet de calculer le sous-ensemble flou $\mu(u_0)$ relatif à la commande du système, à partir de la base de règles fournie par l'expert et

l'entrée fuzzifiée u_0 telle que $u_0 = [u_{0,1}, u_{0,2}, \dots, u_{0,n}]^T$

Ces règles floues lient les sorties aux entrées par des conditions linguistiques. La sortie est obtenue par déduction floue.

Dans ce contexte, on distingue deux sortes de règles d'inférence :

Le « Modus Ponens Généralisé » (MPG), utilisé dans le raisonnement par chaînage avant (déduction successive de faits) et est particulièrement utile dans la commande plane

Le « Modus Tollens Généralisé » (MTG), utilisé en chaînage d'inférence arrière (vérification d'hypothèses) plutôt utilisé en diagnostic

Dans les inférences ou implications floues interviennent les opérateurs ET et OU . L'opérateur ET s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur OU lie les différentes règles.

Afin de traiter numériquement une inférence floue, plusieurs méthodes existent. Les plus importantes sont :

Méthode d'inférence min-max

Méthode d'inférence max-prod

Le tableau suivant résume ces deux principales méthodes

Min-max	Max-prod
$ET \Leftrightarrow \min$	$ET \Leftrightarrow \min$
$OU \Leftrightarrow \max$	$OU \Leftrightarrow \max$
$Alors \Leftrightarrow \min$	$Alors \Leftrightarrow prod(\times)$
Combinaison des règles activées $OU \Leftrightarrow \max$	

Qualitativement, ces règles donnent sensiblement les mêmes résultats

La plus utilisée est la première méthode qu'est de Mamdani :

- Détermination de la conséquence $\rightarrow \min$.
- Agrégation des règles $\rightarrow \max$

Remarque :

Afin de résumer les règles floues, une « table de décision » ou « table d'inférence », anti-diagonale a été adoptée. Elle a pour forme :

sortie inférée	entrées floues				
Entrées floues					

II.3.3.4.1.3 Interface de défuzzification [20]

Les méthodes d'inférences fournissent une fonction d'appartenance résultante pour la variable de sortie. C'est une information floue qu'il convient de transformer en une grandeur de commande précise : c'est l'étape de défuzzification.

Il existe plusieurs techniques dont nous citons :

- Méthode du maximum
- Méthode du centre de gravité
- Méthode de la moyenne pondérée

- Méthode des surfaces
- Méthode des hauteurs

Les trois premières citées ci-dessus, sont les plus utilisées.

Méthode du maximum elle consiste à ne considérer pour chaque sortie que la règle présentant le maximum de validité.

Méthode de la moyenne pondérée: elle considère comme valeur de sortie la moyenne des valeurs préconisées par chaque règle, pondérées par leur degré respectif de validité.

Méthode du centre de gravité C'est la plus performante. Elle consiste à tracer sur le même diagramme, les différentes zones correspondantes à chacune des règles, et à calculer le centre de gravité de la zone considérée.

Si $\mu_{res}(u)$ est la fonction d'appartenance résultante, l'abscisse du centre de gravité u_0 peut être déterminé à l'aide de la relation générale suivante :

Dans le cas d'un univers de discours continu :

$$u_0^* = \frac{\int_a^b u \mu_{res}(u) du}{\int_a^b \mu_{res}(u) du} \quad \text{sur l'intervalle } [a, b]$$

Dans le cas d'un univers de discours discret

$$u_0^* = \frac{\sum_{i=1}^n \mu_u(w_i) \cdot w_i \cdot S_i}{\sum_{i=1}^n \mu_u(w_i) \cdot S_i}$$

n : nombre des niveaux de quantification de la sortie du contrôleur

S_i : surface de la $i^{\text{ème}}$ fonction d'appartenance u de la variable de sortie

w_i : valeur de la commande locale pour laquelle la fonction d'appartenance atteint la valeur maximale $\mu_u(w_i)$

Dans le cas où les fonctions d'appartenance sont de forme symétrique et de distribution uniforme l'équation devient :

$$u_0^* = \frac{\sum_{i=1}^n \mu_u(w_i) \cdot w_i}{\sum_{i=1}^n \mu_u(w_i)}$$

II.3.4 Conclusion

Sans grand risque de se tromper on peut affirmer que la logique floue va prendre de plus en plus de place dans le domaine du réglage. Cette affirmation est motivée par le fait que ce principe ne nécessite pas de modèle du système à régler et qu'il utilise les connaissances de l'opérateur (expert).

Dans le développement, l'approche digitale est, évidemment, la plus souple à mettre en oeuvre. Son coût, son encombrement et sa consommation peuvent devenir un handicap pour certaines applications en production. Pour ces applications l'approche analogique apporte une solution intéressante.

Dans ce qui va suivre ; on va combiner les avantages des deux techniques qu'on a étudiées dans ce chapitre ; pour concevoir une commande hybride, tirant profit des caractéristiques du raisonnement approximatif de la logique floue ; et la capacité d'apprentissage des RNA.

II.4 Systèmes hybrides neuroflous

II.4.1 Introduction [21]

Les réseaux de neurones et la logique floue sont deux concepts totalement inspirés du raisonnement humain. Bien que ces deux modèles numériques soient différents du point de vue structure, ils possèdent de nombreux points communs. En effet, l'utilisation de ces deux techniques ne nécessite pas de modèle mathématique bien défini car la résolution se base sur les valeurs numériques des E/S pour les RNA et sur la logique du système dans le cas de la logique floue. De plus, les résultats sont généralement incertains. En revanche ces deux techniques peuvent accomplir des tâches complémentaires. La logique floue fournit des connaissances avec un certain degré d'incertitude ; et les RNA peuvent modéliser et reproduire l'apprentissage du cerveau humain, d'où l'idée de combiner ces deux techniques afin de créer un système artificiel intelligent caractérisé par :

- Combinaison des traitements parallèles et des capacités d'apprentissage des réseaux de neurones avec les raisonnements qualitatifs et les capacités d'explication des systèmes flous : Les RNA deviennent plus transparents, les systèmes flous acquièrent la capacité d'apprendre.

- RNA fonctionnellement équivalent à un modèle d'inférence flou ; on peut l'entraîner à développer des règles floues SI-ALORS et à trouver les fonctions d'appartenance de variables d'entrées/sorties en partant d'un ensemble de données représentatives.
- On peut aussi y inclure les connaissances d'un expert.
- Structure similaire à RNA multi couches: 1 couche d'entrée, 1 couche de sortie et 3 couches cachées pour les fonctions d'appartenance et les règles.

II.4.2 Principe de fonctionnement des réseaux neuroflous [16] :

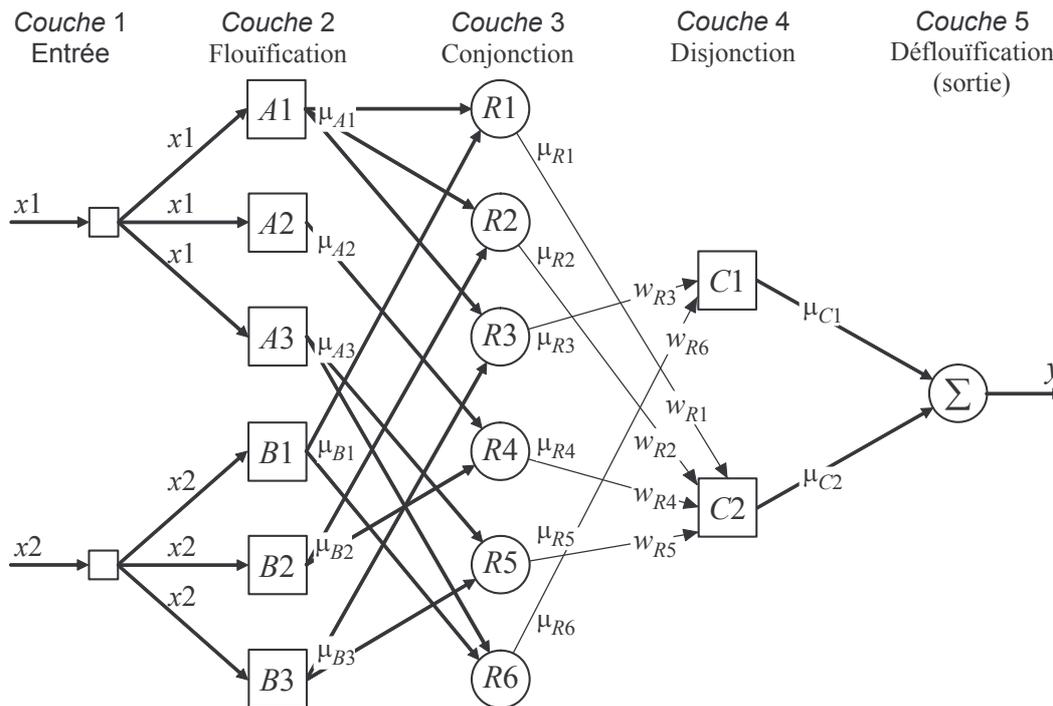


Figure 2.14 : Architecture d'un système neuroflou [16]

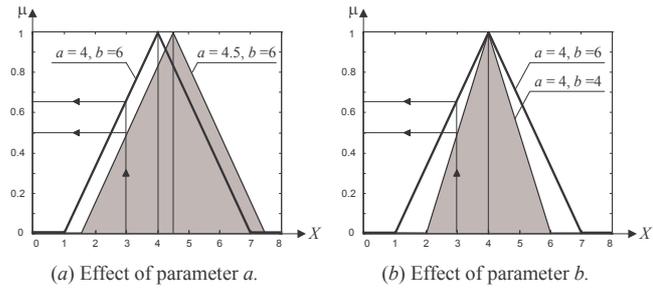
Couche 1 : Agit comme un tampon qui transmet simplement les données d'entrée à la couche de fuzzification. On a :

$$y_i^{(1)} = x_i^{(1)}$$

Couche 2 : Les neurones de cette couche réalisent chacun un ensemble flou qui servira dans les antécédents des règles. Chaque neurone reçoit une valeur d'entrée précise et génère son degré d'appartenance à l'ensemble flou du neurone.

Dans le cas d'ensembles flous triangulaires, on peut utiliser des fonctions d'appartenance de même forme qui sont définies par deux paramètres $\{a, b\}$:

$$y_i^{(2)} = \begin{cases} 0, & \text{if } x_i^{(2)} \leq a - \frac{b}{2} \\ 1 - \frac{2|x_i^{(2)} - a|}{b}, & \text{if } a - \frac{b}{2} < x_i^{(2)} < a + \frac{b}{2} \\ 0, & \text{if } x_i^{(2)} \geq a + \frac{b}{2} \end{cases}$$



Couche 3 : Elle réalise les conjonctions des antécédents de règles floues. Chaque neurone dans cette couche reçoit comme entrée les degrés d'appartenance définis dans la couche 2. L'intersection floue est réalisée avec l'opérateur produit :

$$y_i^{(3)} = x_{1i}^{(3)} \times x_{2i}^{(3)} \times \dots \times x_{ki}^{(3)} \qquad y_{R1}^{(3)} = \mu_{A1} \times \mu_{B1} = \mu_{R1}$$

Couche 4 : Elle réalise les disjonctions des antécédents de règles floues. Chaque neurone dans cette couche reçoit comme entrée les degrés d'appartenance définis dans la couche 3. L'union floue est réalisé à l'aide de l'opérateur ou probabiliste :

$$y_i^{(4)} = x_{1i}^{(4)} \oplus x_{2i}^{(4)} \oplus \dots \oplus x_{ki}^{(4)} \qquad y_{C1}^{(4)} = \mu_{R3} \oplus \mu_{R6} = \mu_{C1}$$

μ_{C1} représente la force combinée des conjonctions implémentées par les neurones R3 et R6.

Couche 5 : Chaque neurone dans cette couche prend les ensembles flous précédent, écrêtés par la force des règles correspondantes, et les combine en un seul ensemble flou. Ce dernier est alors défuzzifiée par une méthode standard.

II.4.2.1 Structure ANFIS (Adaptive NeuroFuzzy Inference System) [16], [7]

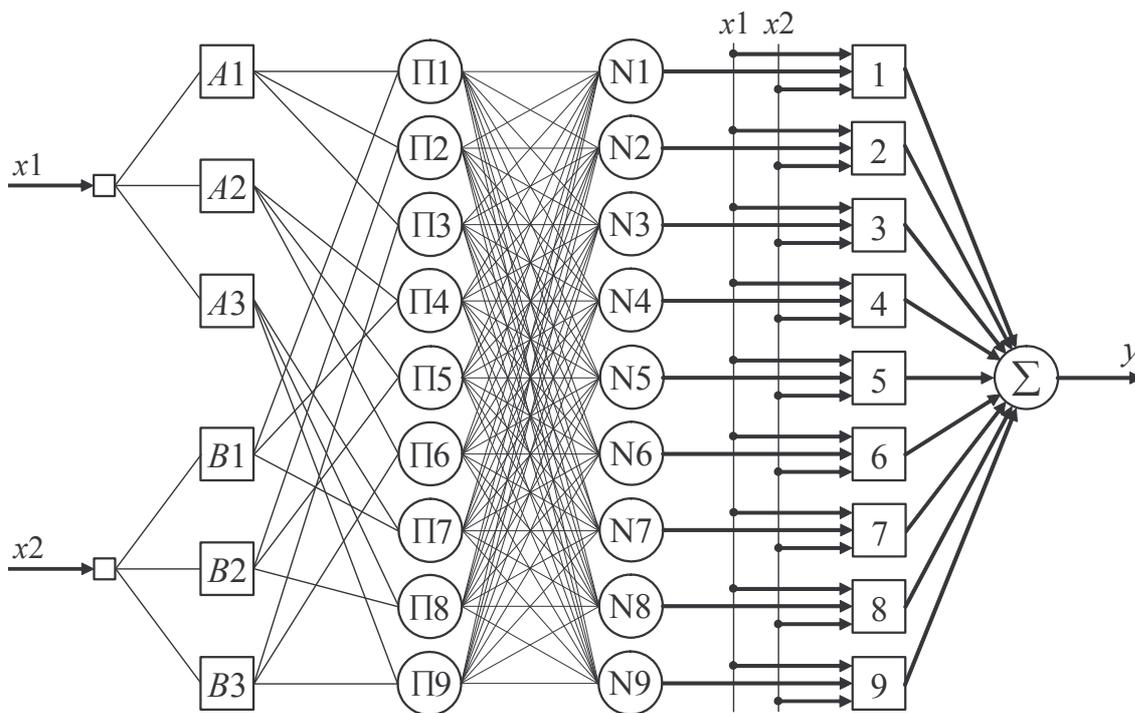


Figure 2.15 : ARCHITECTURE ANFIS [16]

C'est un modèle de génération automatique de règles floues basé sur le modèle d'inférence de Sugeno :

IF x_1 is A_1 AND x_2 is A_2 . . . AND x_m is A_m

THEN $y = f(x_1, x_2, \dots, x_m)$

où x_1, x_2, \dots, x_m sont des variables d'entrée et A_1, A_2, \dots, A_m sont des ensembles flous.

Lorsque $y =$ constante, on obtient un modèle de Sugeno d'ordre zéro. Le conséquent d'une règle est un singleton.

Lorsque y est une combinaison linéaire des entrées :

$$y = k_0 + k_1 x_1 + k_2 x_2 + \dots + k_m x_m$$

on obtient un modèle de Sugeno de premier ordre.

II.4.2.2 Principe de fonctionnement du réseau ANFIS [7]:

Couche 1 : Transmet simplement les données d'entrée à la couche de fuzzification.

Couche 2 : Les neurones de cette couche réalisent les ensembles flous qui serviront dans les antécédents des règles. Dans le modèle de Jang, les fonctions d'appartenance sont des gaussiennes.

Couche 3 : Chaque neurone dans cette couche correspond à une règle floue Sugeno. il reçoit les sorties des neurones de fuzzification et calcule son activation. La conjonction des antécédents est réalisée avec l'opérateur produit. Ainsi, la sortie du neurone i de la couche 3 est donnée par

$$y_i^{(3)} = \prod_{j=1}^k x_{ji}^{(3)} \quad \text{et} \quad y_{\prod 1}^{(3)} = \mu_{A_1} \times \mu_{B_1} = \mu_1$$

où μ_1 représente le degré de vérité de Règle 1.

Couche 4 : Chaque neurone calcule le degré de vérité normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final. Ainsi la sortie du neurone i de la couche 4 est :

$$y_i^{(4)} = \frac{x_{ii}^{(4)}}{\sum_{j=1}^n x_{ji}^{(4)}} = \frac{\mu_i}{\sum_{j=1}^n \mu_j} = \bar{\mu}_i \quad y_{N1}^{(4)} = \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3 + \mu_4} = \bar{\mu}_1$$

Couche 5 : Chaque neurone i de cette couche est relié à un neurone de normalisation correspondant et aux entrées initiales du réseau. Il calcule est le conséquent pondéré de la règle sous jacent comme étant

$$y_i^{(5)} = x_i^{(5)} [k_{i0} + k_{i1} x_1 + k_{i2} x_2] = \bar{\mu}_i [k_{i0} + k_{i1} x_1 + k_{i2} x_2]$$

Où les X_i sont les entrées, et k_{i0} , k_{i1} et k_{i2} sont des paramètres du conséquent de la règle i .

Couche 6 : Comprend un seul neurone qui fournit la sortie de ANFIS en calculant la somme des sorties de tous les neurones de défuzzification.

$$y = \sum_{i=1}^n x_i^{(6)} = \sum_{i=1}^n \bar{\mu}_i [k_{i0} + k_{i1} x_1 + k_{i2} x_2]$$

II.4.2.3 Entraînement d'un réseau ANFIS [16]

On utilise un algorithme à deux temps où on estime d'abord les paramètres des conséquents par une technique de moindres carrés et ensuite les poids du réseau par une descente de gradient.

Chaque époque d'entraînement comprend une passe avant et une passe arrière. Durant la passe avant, les patrons d'entrée servent à déterminer les sorties des neurones couche par couche, permettant de déterminer les valeurs des paramètres de conséquents en bout de compte.

Durant la passe arrière, l'algorithme de rétro propagation d'erreur est appliqué pour régler les poids des différentes couches.

Partant de P paires d'apprentissage, on peut former P équations linéaires en fonction des paramètres des conséquents :

$$\begin{cases} y_d(1) = \bar{\mu}_1(1)f_1(1) + \bar{\mu}_2(1)f_2(1) + \dots + \bar{\mu}_n(1)f_n(1) \\ y_d(2) = \bar{\mu}_1(2)f_1(2) + \bar{\mu}_2(2)f_2(2) + \dots + \bar{\mu}_n(2)f_n(2) \\ \vdots \\ y_d(p) = \bar{\mu}_1(p)f_1(p) + \bar{\mu}_2(p)f_2(p) + \dots + \bar{\mu}_n(p)f_n(p) \\ \vdots \\ y_d(P) = \bar{\mu}_1(P)f_1(P) + \bar{\mu}_2(P)f_2(P) + \dots + \bar{\mu}_n(P)f_n(P) \end{cases}$$

Où $\bar{\mu}_i$ est la valeur moyenne de μ_i , et $f_i()$ est la fonction de sortie dont on veut déterminer les paramètres.

On peut écrire l'équation précédente sous la forme $y_d = A k$,

où y_d est un vecteur désiré de dimension $P \times 1$:

$$y_d = \begin{bmatrix} y_d(1) \\ y_d(2) \\ \vdots \\ y_d(p) \\ \vdots \\ y_d(P) \end{bmatrix} \quad A = \begin{bmatrix} \bar{\mu}_1(1) & \bar{\mu}_1(1)x_1(1) & \dots & \bar{\mu}_1(1)x_m(1) & \dots & \bar{\mu}_n(1) & \bar{\mu}_n(1)x_1(1) & \dots & \bar{\mu}_n(1)x_m(1) \\ \bar{\mu}_1(2) & \bar{\mu}_1(2)x_1(2) & \dots & \bar{\mu}_1(2)x_m(2) & \dots & \bar{\mu}_n(2) & \bar{\mu}_n(2)x_1(2) & \dots & \bar{\mu}_n(2)x_m(2) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ \bar{\mu}_1(p) & \bar{\mu}_1(p)x_1(p) & \dots & \bar{\mu}_1(p)x_m(p) & \dots & \bar{\mu}_n(p) & \bar{\mu}_n(p)x_1(p) & \dots & \bar{\mu}_n(p)x_m(p) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots & \vdots & \dots & \vdots \\ \bar{\mu}_1(P) & \bar{\mu}_1(P)x_1(P) & \dots & \bar{\mu}_1(P)x_m(P) & \dots & \bar{\mu}_n(P) & \bar{\mu}_n(P)x_1(P) & \dots & \bar{\mu}_n(P)x_m(P) \end{bmatrix}$$

et k est le vecteur des paramètres de conséquent inconnus de dimension $n(1+m) \times 1$:

$$k = [k_{10} \quad k_{11} \quad k_{12} \quad \& k_{1m} \quad k_{20} \quad k_{21} \quad k_{22} \quad \& k_{2m} \quad \& k_{n0} \quad k_{n1} \quad k_{n2} \quad \& k_{nm}]^T$$

On a donc :

$$k = A^{-1}y_d$$

Une fois le vecteur k déterminé, le vecteur de sortie du réseau y peut être calculé ainsi que le vecteur d'erreur associé, e :

$$e = y_d - y$$

Lors de la passe arrière, l'algorithme de rétro propagation d'erreur est appliqué pour mettre à jour les poids des antécédents des règles.

Dans l'algorithme ANFIS de Jang, on optimise aussi bien les paramètres de antécédents que ceux des conséquents. Durant la passe avant, les paramètres des conséquents sont adaptés alors que les paramètres des antécédents sont maintenus constants ; durant la passe arrière, les rôles sont échangés.

II.4.3 Application de la commande Neuro-Floue au système de suspension de véhicule [7], [2], [3]:

II.4.3.1 Variables de commandes

Avant de passer en revue les régulateurs synthétisés, il y a lieu de choisir une variable à réguler qui nous permettrait d'atteindre les objectifs recherchés. La variable choisie est la suivante :

$$e = x_1 - \bar{x}_3$$

Où x_1 est le déplacement du châssis et \bar{x}_3 est une version filtrée du déplacement de la roue x_3 :

$$\bar{x}_3 = \frac{\varepsilon}{s + \varepsilon} x_3.$$

Le choix de la valeur de ε affecte la régulation :

- Pour de petites valeurs d' ε , on obtient un filtrage de x_3 par un filtre passe-bas, donc la valeur régulée e est essentiellement égale à x_1 (déplacement du châssis) car les perturbations introduites par la chaussée, et transmises par la roue à la caisse, sont rejetées par le filtre. Ceci nous permet d'obtenir une suspension confortable ; au détriment de la tenue de route.
- Si la valeur d' ε est grande, des fréquences de plus en plus grandes peuvent passer à travers le filtre. La valeur régulée se rapproche de la déflexion de la suspension, et ceci va nous permettre d'améliorer la tenue de route en imposant à la caisse une trajectoire sensiblement identique à celle de la roue (et donc la chaussée), ce qui va réduire la perturbation enjondrée par l'inertie de la caisse sur la roue.

Pour combiner les deux cas, il faut concevoir une logique permettant de changer le paramètre ε , et donc de mettre l'accent sur un des deux objectifs : confort ou bien limiter la déflexion suivant plusieurs critères :

- Etat de la chaussée
- Vitesse du véhicule
- Type du véhicule, et sa fonction

II.4.3.2 Architecture du régulateur neuroflou utilisé

II.4.3.2.1 Modèle quart de véhicule (mono roue)

Le régulateur utilisé est de type ANFIS, ce choix est motivé par :

- algorithme d'apprentissage simplifié du fait que les poids se trouvent au niveau de la dernière couche
- vitesse de convergence supérieure aux structures standard

La sortie du réseau est nette (SUGENO) ; sa structure est analogue à celle de la figure (2.15) avec :

- 7 ensembles flous gaussiens caractérisés par leur moyennes et écart-type, pour chaque variable de régulation (erreur et dynamique de l'erreur).
- 49 règles floues

Les ensembles flous des entrées sont donnés par les figures suivantes :

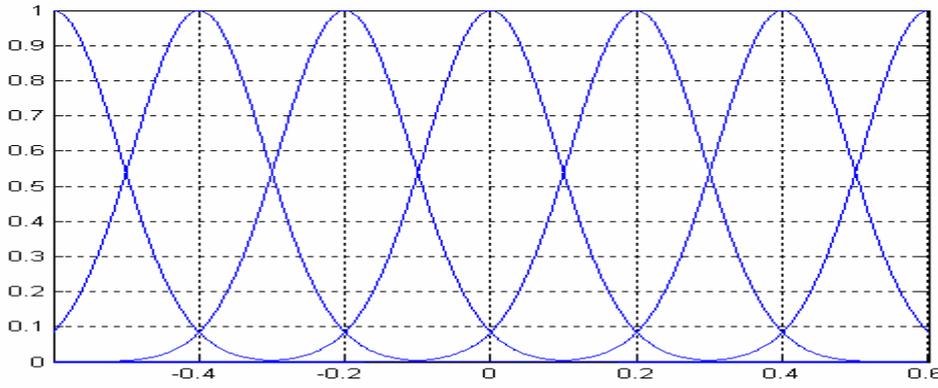


Figure 2.16 Ensembles flous de l'erreur

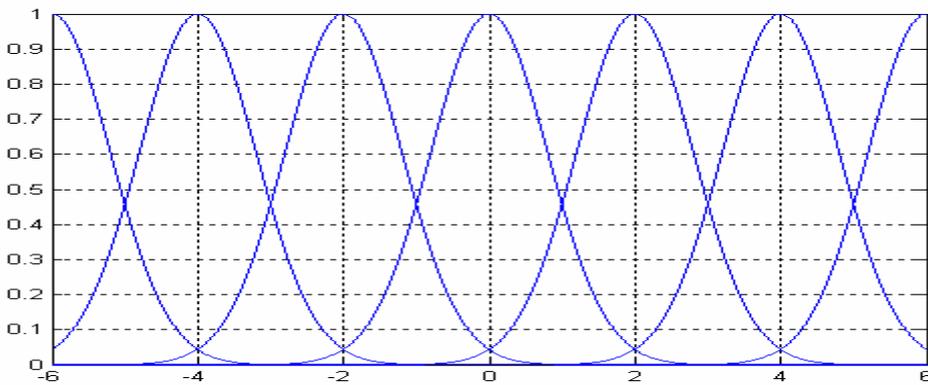


Figure 2.17 Ensembles flous de la dynamique de l'erreur

Remarques :

- * Le choix de l'univers de discours est basé sur des simulations en boucle ouverte (domaine de variation des différentes grandeurs d'entrée)
- * Nous avons choisit le nombre de partitions floues (7) après avoir testé les configurations plus simples (3,5) ce qui a donné des résultats non satisfaisants (mauvaises performances en poursuite de trajectoire avec présence du phénomène du sur-apprentissage)

Le schéma de commande est illustré dans la figure suivante :

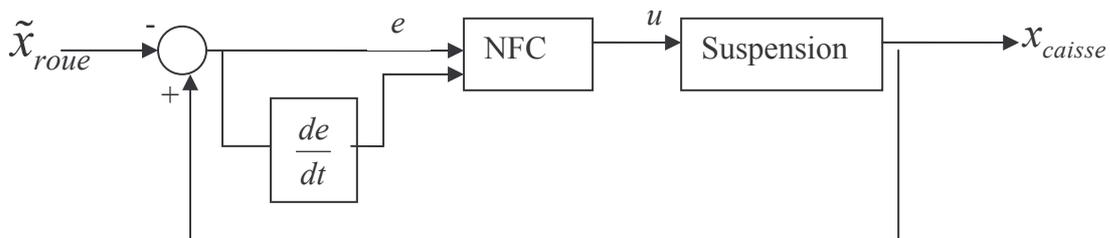


Figure 2.18: Contrôleur neuroflou [20]

II.4.3.2.1.1 Fonctionnement du NFC [20],[16]

On adopte les notations suivantes :

$O_{i,j}$: sortie du $j^{\text{ème}}$ noeud de la couche i

μ_A : fonction d'appartenance à l'ensemble A

$m_i; \sigma_i$ moyenne et écart-type des fonctions d'appartenance gaussiennes

p, q, s : paramètres variables de la dernière couche caractérisant les décisions

Le régulateur possède une architecture SUGENO et possède 5 couches qui effectuent les tâches suivantes :

Couche 1 : couche d'entrée :

$$O_{1,1} = e$$

$$O_{1,2} = \frac{de}{dt}$$

Couche 2 : couche de fuzzification

$$O_{2,i} = \mu_{ei}(e) \quad i = 1:7$$

$$O_{2,i+7} = \mu_{dei}(de) \quad i = 1:7$$

$$\text{vec} : \mu(x) = \exp \left\{ - \left[\frac{x - m}{\sigma} \right]^2 \right\}$$

Couche 3: couche de règles (type produit)

$$O_{3,i} = w_i = \mu_{e(1:7)}(e) \times \mu_{de(1:7)}(de) \quad i = 1:49$$

Couche 4 couche de normalisation des prémisses

$$\bar{w}_i = \frac{w_i}{\sum_{j=1:49} w_j} \quad i = 1:49$$

Couche 5 couche de décisions

$$O_5 = \sum_{i=1:49} \bar{w}_i (p_i e + q_i \frac{de}{dt} + s_i)$$

II.4.3.2.1.2 Apprentissage du NFC [7]

Durant la phase d'apprentissage on va utiliser l'algorithme du rétro propagation du gradient minimisant l'erreur suivante :

$$E = \sum_{p=1:P} E_p$$

Avec :

$$E_p = \frac{1}{2}(T_p - O_{5,p})^2$$

P : nombre d'exemples d'apprentissage

T_p : sortie désirée

O_{5,p} : sortie du réseau

La dérivée de l'erreur par rapport au paramètres p,q,s est donnée par :

$$\frac{\partial^+ E_p}{\partial x} = \frac{\partial E_p}{\partial O_{5,p}} \frac{\partial O_{5,p}}{\partial x} \quad \text{avec} \quad \frac{\partial E_p}{\partial O_{5,p}} = -(T_p - O_{5,p})$$

Avec x=p,q ou s.

Et on a :

$$\begin{cases} \frac{\partial O_{5,p}}{\partial p_i} = \bar{w}_i e \\ \frac{\partial O_{5,p}}{\partial q_i} = \bar{w}_i \frac{de}{dt} \\ \frac{\partial O_{5,p}}{\partial s_i} = \bar{w}_i \end{cases} \quad i = 1:49$$

D'où :

$$\begin{cases} \frac{\partial^+ E_p}{\partial p_i} = -\bar{w}_i e (T_p - O_{5,p}) \\ \frac{\partial^+ E_p}{\partial q_i} = -\bar{w}_i \frac{de}{dt} (T_p - O_{5,p}) \\ \frac{\partial^+ E_p}{\partial s_i} = -\bar{w}_i (T_p - O_{5,p}) \end{cases} \quad i = 1:49$$

La dérivée globale s'obtient par

$$\frac{\partial^+ E}{\partial x_i} = \sum_{p:P} \frac{\partial^+ E_p}{\partial x_i}$$

D'où les algorithmes d'apprentissage :

$$\begin{cases} p_i(t+1) = p_i(t) - \eta \frac{\partial^+ E}{\partial p_i} \\ q_i(t+1) = q_i(t) - \eta \frac{\partial^+ E}{\partial q_i} \\ s_i(t+1) = s_i(t) - \eta \frac{\partial^+ E}{\partial s_i} \end{cases} \quad i = 1:49$$

II.4.3.2.2 Modèle demi véhicule (Bicyclette)

Dans ce cas on a synthétisé une commande décentralisée constituée de deux régulateurs identiques au cas précédent (quart de véhicule) ; et on considère le modèle $\frac{1}{2}$ comme une interconnexion de deux modèle $\frac{1}{4}$.

La structure des régulateurs reste donc la même ; la seule différence est que pendant la phase d'apprentissage, les perturbations sont introduite au niveau des deux roues simultanément, ce qui permet au système de prendre en considération l'effet de l'interconnexion.

II.4.3.2.3 Modèle véhicule complet (3d 7 ddl)

Dans ce cas on a synthétisé une commande constituée de quatre régulateurs identiques à celui synthétisé pour le modèle quart de véhicule ; et on considère le modèle complet comme une interconnexion de quatre modèle $\frac{1}{4}$. L'apprentissage se déroule d'une façon identique au modèle demi véhicule (introduction des perturbations sur toutes les roues).

II.4.4 Critères de performance

(Confort - tenue de route - énergie de commande) [2],[3]

II.4.4.1 Critères pour le quart de véhicule

La seule entrée de perturbation considérée est le profil vertical de la route.

Les principaux critères concernent le confort, l'adhérence ainsi que l'énergie de la commande.

Le confort est directement lié au déplacement de la caisse, par contre la sécurité (tenue de route) est caractérisée par l'écart entre la roue et la chaussée.

Soit T la durée de la simulation, on peut alors calculer les critères :

- confort $J_c = \min \sum \ddot{x}_c = \min \sum \dot{x}_2$ (voir modele) ou $\min \sum x_1$
- tenue de route: $J_s = \min \sum |x_w - r| = \min \sum |x_3 - z_r|$
- énergie de la commande : $J_u = \min \sum |u|$

II.4.4.2 Critères pour le modèle demi véhicule

- En ce qui concerne le confort, on cherche à minimiser l'accélération de notre caisse à l'avant comme à l'arrière mais aussi à minimiser le tangage, et pour cela on minimise $\sum |\ddot{\theta}|$ ou bien $\sum |\theta|$.
- En ce qui concerne la tenue de route, notre critère peut s'exprimer par : $\min \sum |x_{w1} - r_1|$ et $\min \sum |x_{w2} - r_2|$ (soit la somme des deux).
- Pour les critères de commande, on doit minimiser : $\sum |u_f| + \sum |u_r|$

II.4.4.3 Critères pour le modèle véhicule complet

- Pour optimiser le confort, on doit minimiser en plus de l'accélération de notre caisse d'une façon globale, le tangage et le roulis, pour cela on minimise $\sum |\ddot{\phi}|$ ou bien $\sum |\phi|$
- En ce qui concerne la tenue de route, notre critère peut s'exprimer par : $\min \sum |x_{rl} - r_{rl}| + \sum |x_{fr} - r_{fr}| + \sum |x_{rl} - r_{rl}| + \sum |x_{rr} - r_{rr}|$
- Pour les critères de commande, on doit minimiser : $\sum |u_{fl}| + \sum |u_{fr}| + \sum |u_{rl}| + \sum |u_{rr}|$

Chapitre II :
Simulations & Résultats

III.1 Introduction

Dans cette partie nous allons présenter les résultats des simulations effectuées sur les modèles énoncés dans le premier chapitre ; commandés par des régulateurs neuroflous analogues à ceux présentés dans le chapitre précédent ; bien entendu ; les régulateurs ont subi un apprentissage de durée variable afin de déterminer leurs paramètres ; la durée de l'apprentissage dépend en partie de la complexité du modèle utilisé, ensuite nous utiliseront deux types de perturbations courantes dans le domaine de la suspension pour tester l'efficacité des NFC :

- montée sur trottoir de 30 cm
- dos d'âne de 20 cm

Remarquons que l'amplitude des perturbations est importante ; elle représente au fait les cas extrêmes que peut subir le système de suspension.

Durant la conception des régulateurs neuroflous, trois configurations sont considérées :

- Configuration « confort optimal » où $\varepsilon = 2$; elle correspond à un filtrage poussé des perturbations de la chaussée.
- Configuration « tenue de route optimale » où $\varepsilon = 20$.
- Configuration hybride à mi-chemin entre les deux configurations précédentes ; où $\varepsilon = 5$

Ainsi, pour chaque configuration un apprentissage approprié est effectué, ce qui permet d'obtenir un jeu de paramètres spécifique ; remarquons toutefois que pour $\varepsilon = 2$, la durée de l'apprentissage ainsi que la fréquence de simulation augmentent sensiblement.

A la fin de chaque simulation ; les critères de performance (voir chapitre I) sont calculés afin de pouvoir comparer les configurations d'une façon globale.

Pour le modèle complet (3d 7ddl), un test d'interconnexion est effectué, il consiste à introduire une perturbation sur une seule roue et observer le comportement du système, du point de vue rejet de perturbations provenant des autres roues (sous-systèmes).

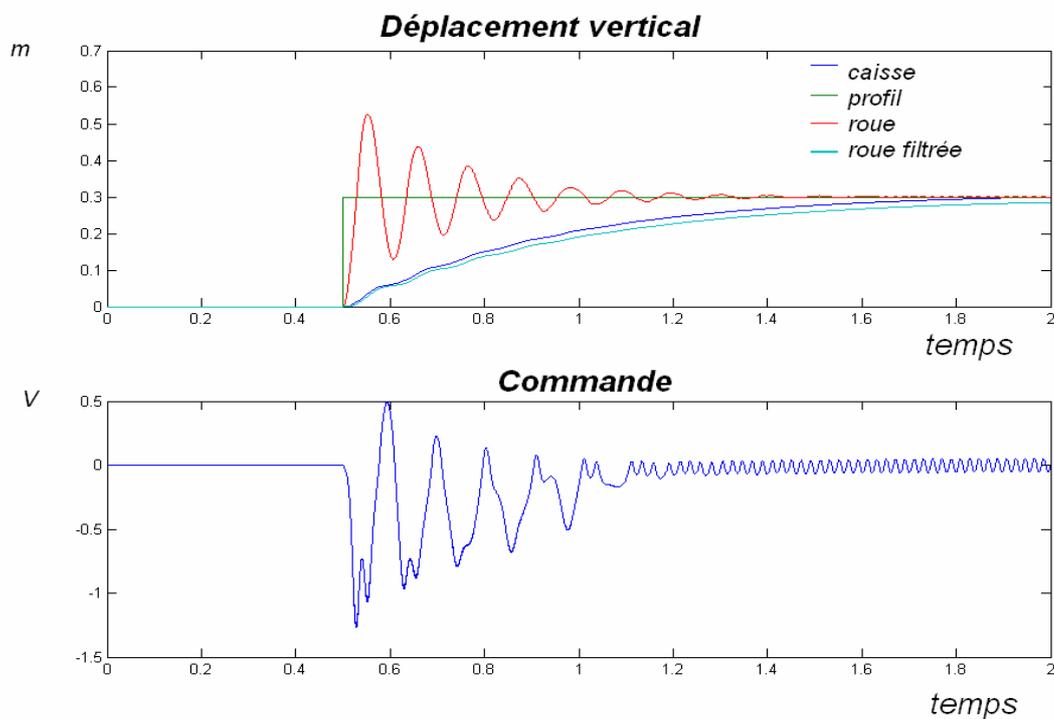
Les perturbations sont introduites à partir de l'instant $t=0.5$ s ; pour les modèles demi et 3d, la vitesse intervient dans le calcul du profil de la perturbation sur les roues arrière, en supposant que la vitesse du véhicule reste constante pendant le passage du véhicule sur le profil.

III.2 Simulations

III.2.1 Modèle quart de véhicule

III.2.1.1 configuration « confort optimal » $\varepsilon = 2$

i. Montée sur trottoir



ii. Passage sur dos d'âne

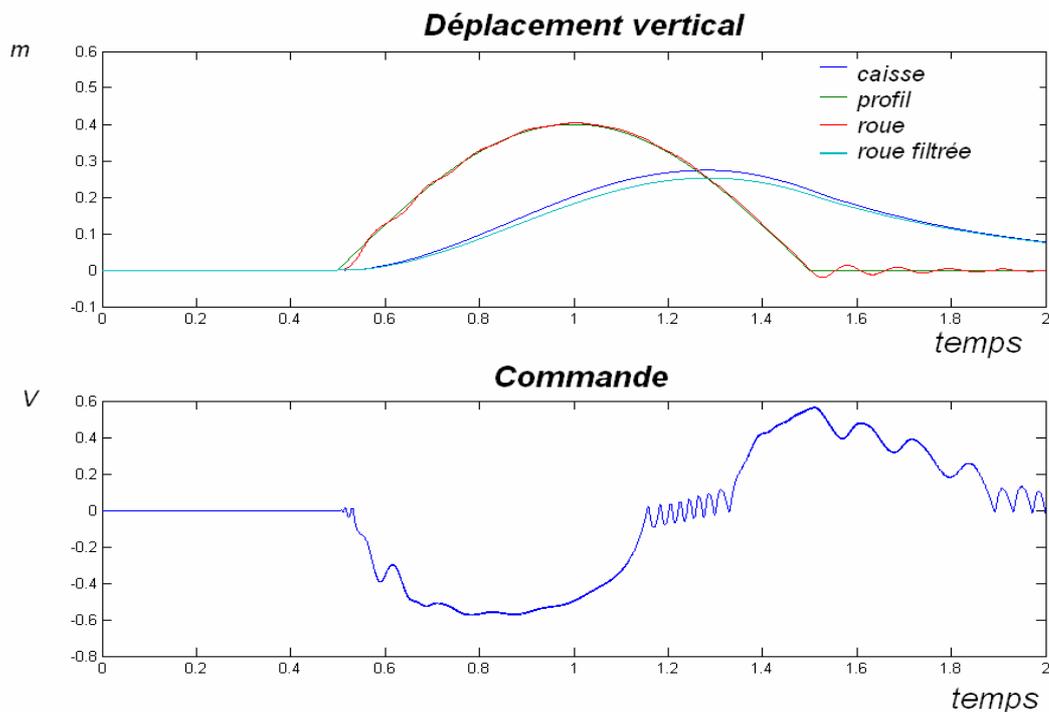
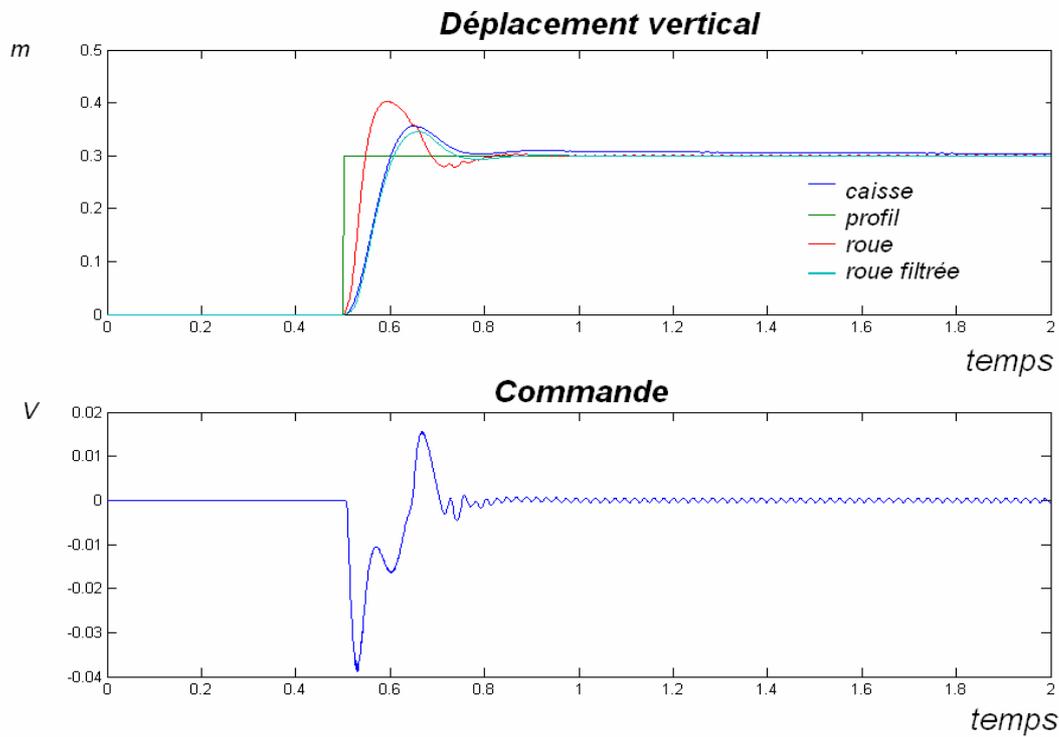


Figure III.1 : Réponse du modèle quart de véhicule /configuration « confort optimal »

III.2.1.2 configuration « tenue de route optimal » $\varepsilon = 20$

i. Montée sur trottoir



ii. Passage sur dos d'âne

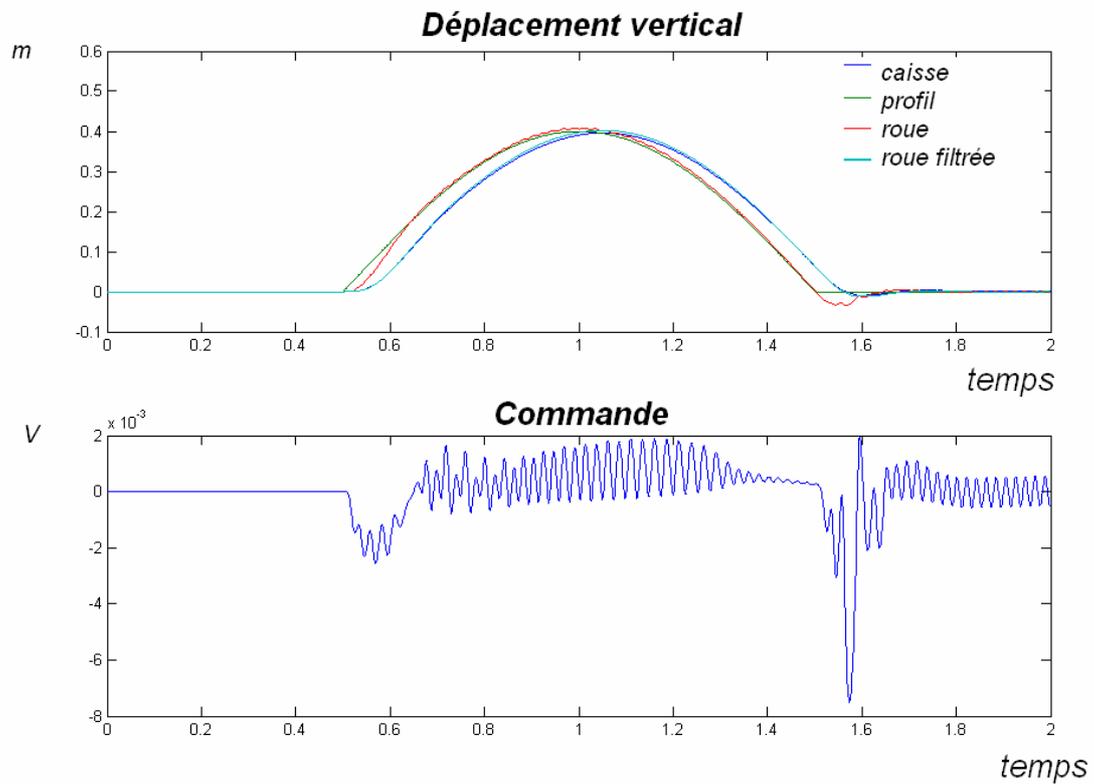
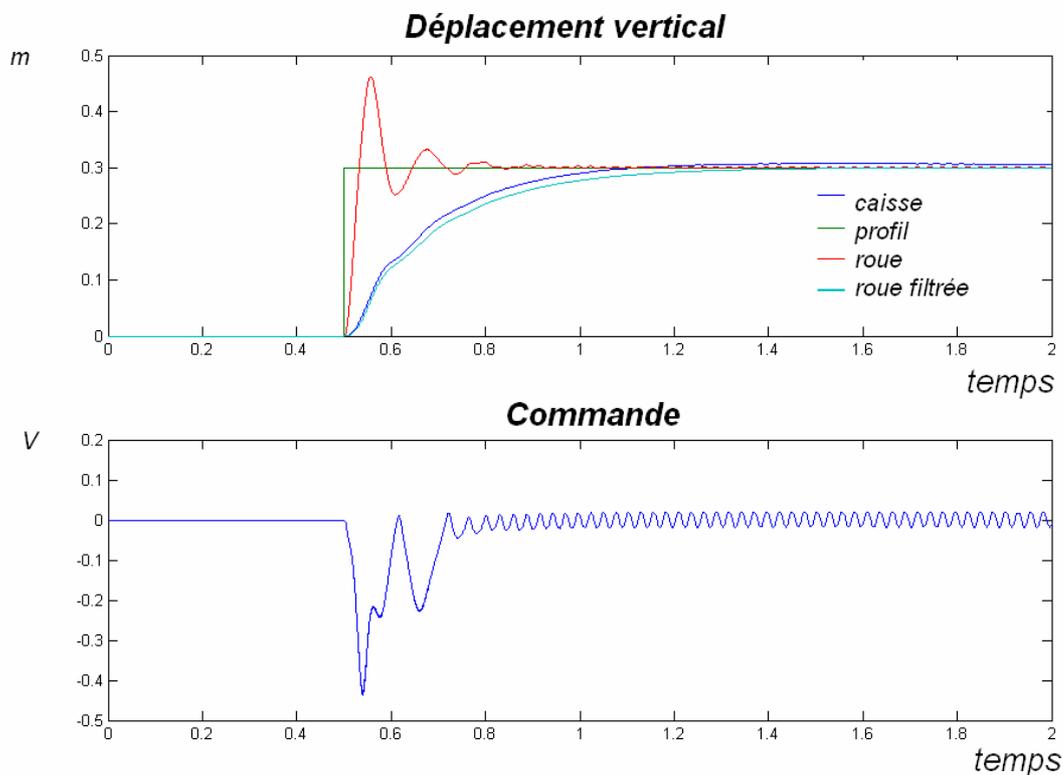


Figure III.2 : Réponse du modèle quart de véhicule /config. « tenue de route optimal »

III.2.1.3 configuration « hybride » $\varepsilon = 5$

i. Montée sur trottoir



ii. Passage sur dos d'âne

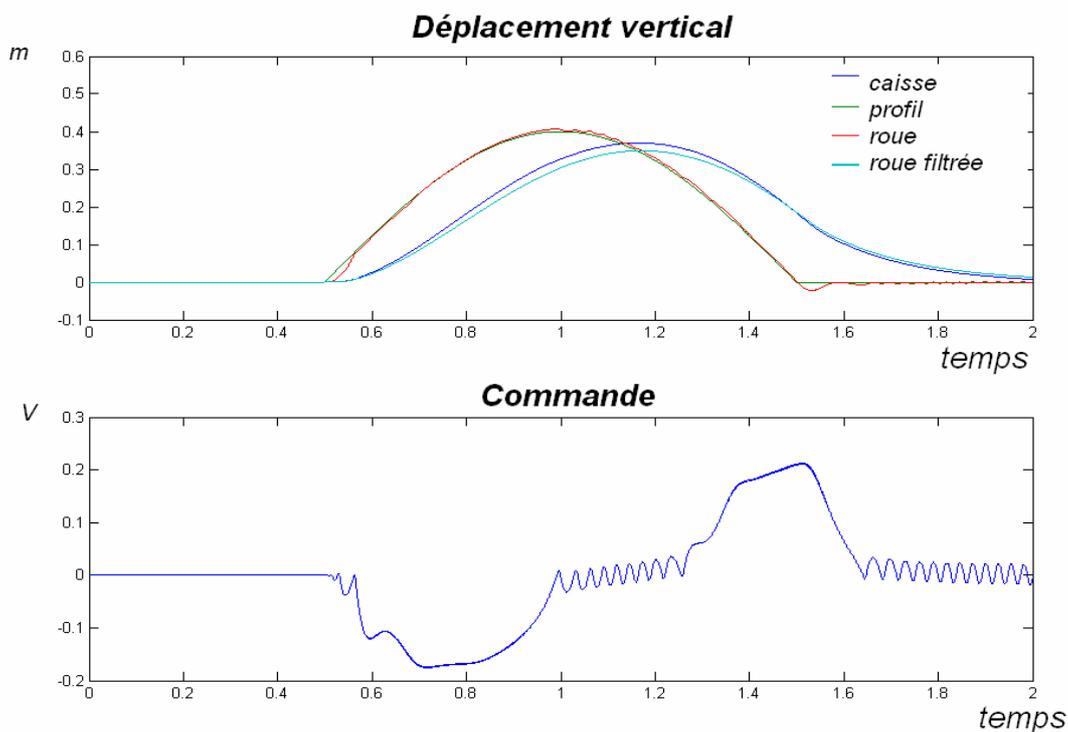


Figure III.3 : Réponse du modèle quart de véhicule /config. « hybride »

III.2.1.4 Réponse de la suspension passive aux perturbations :

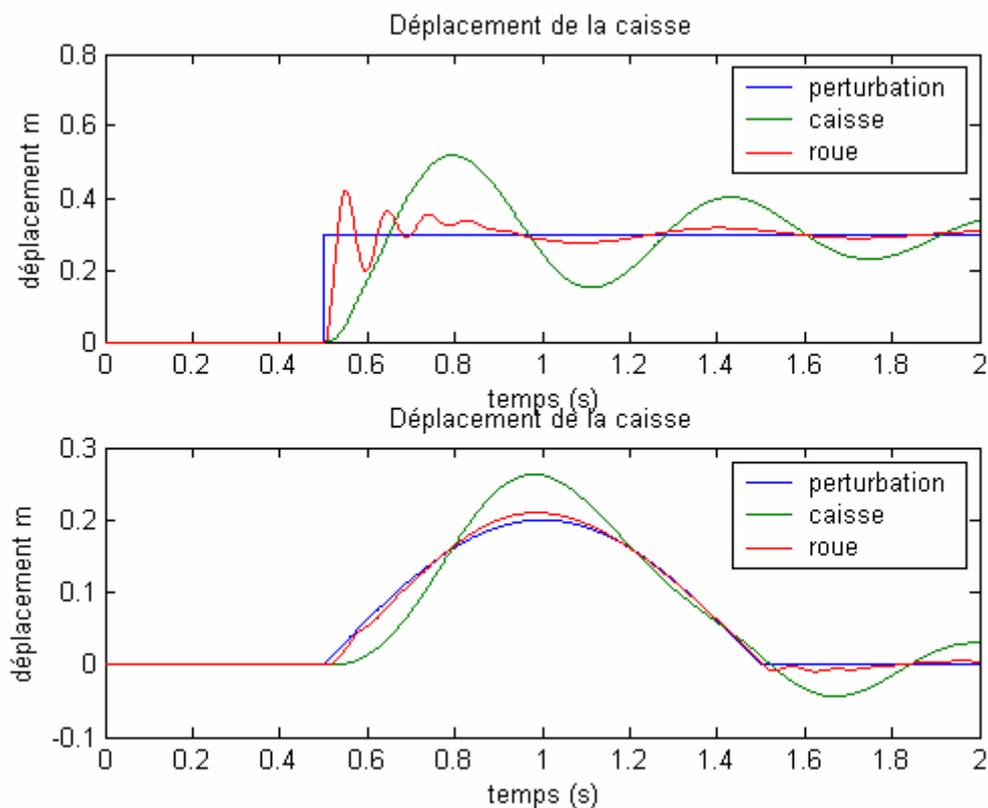


Figure III.4 : Réponse du modèle quart de véhicule/ suspension passive

III.2.1.5 Comparaison des valeurs des critères de performance

Les critères sont calculés en utilisant la perturbation du type trottoir.

Configuration \ Critère	$\varepsilon = 2$	$\varepsilon = 5$	$\varepsilon = 20$	passive
Tenue de Route	0.03932	0.01999	0.01638	0.03641
Confort	0.32959	0.40512	0.44509	1.34883
Energie de commande	0.85248	0.16886	0.00865	X

III.2.1.6 Remarques

- Comme prévu théoriquement les critères confirment l'existence du compromis tenue de route/ confort ; on remarque aussi que l'énergie de commande croît d'une façon exponentielle lorsqu'on favorise le confort ($\varepsilon = 2$).
- La suspension active donne des résultats meilleurs sur tous les plans et pour toutes les configurations testées par rapport à la suspension passive ; elle nécessite toutefois un apport d'énergie.

III.2.2 modèle demi véhicule – vitesse 10 km/h

III.2.2.1 configuration « confort optimal » $\varepsilon = 2$

i. Montée sur trottoir

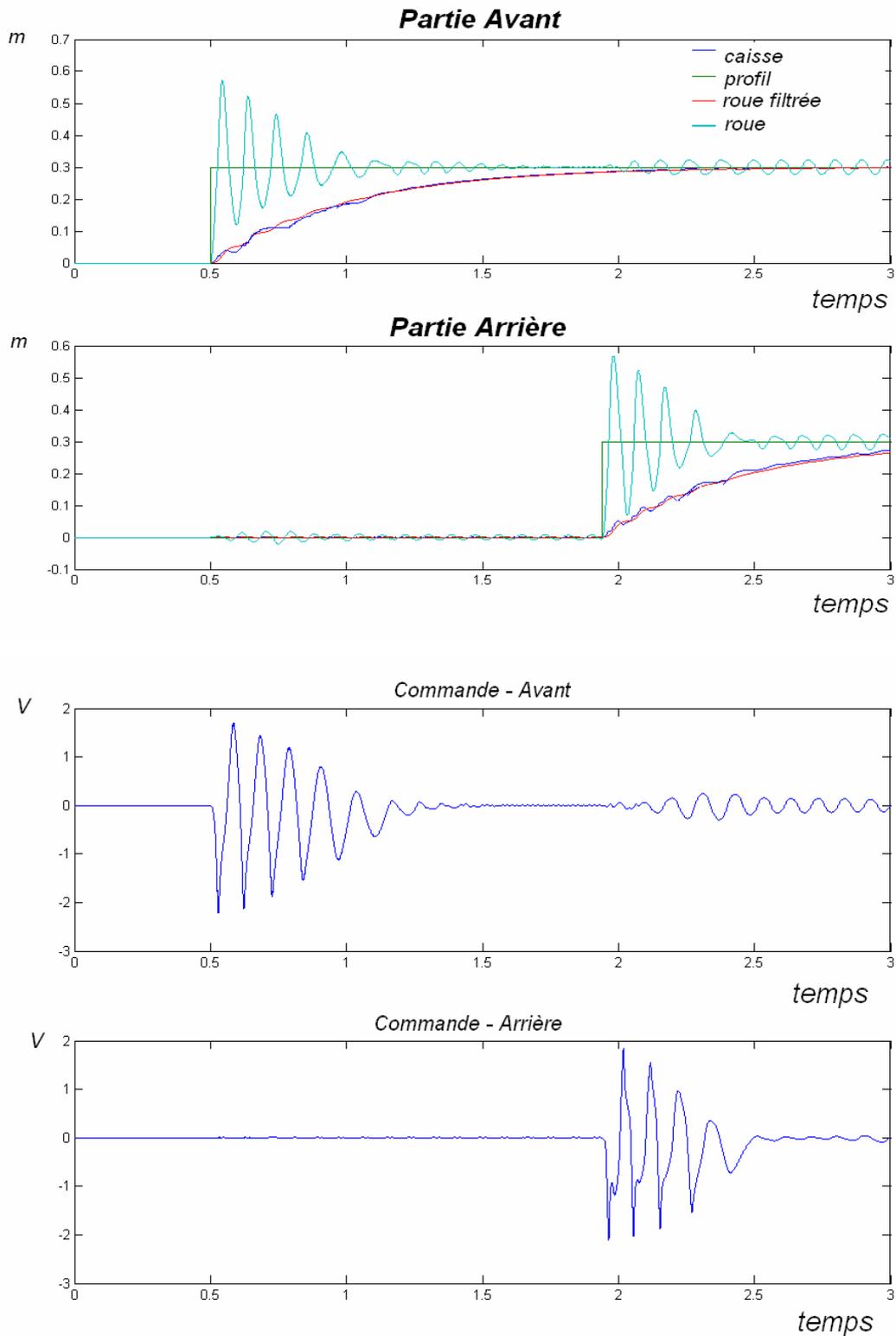


Figure III.5 : Réponse du modèle demi véhicule (montée sur trottoir)/ $\varepsilon = 2$

ii. Passage sur dos d'âne

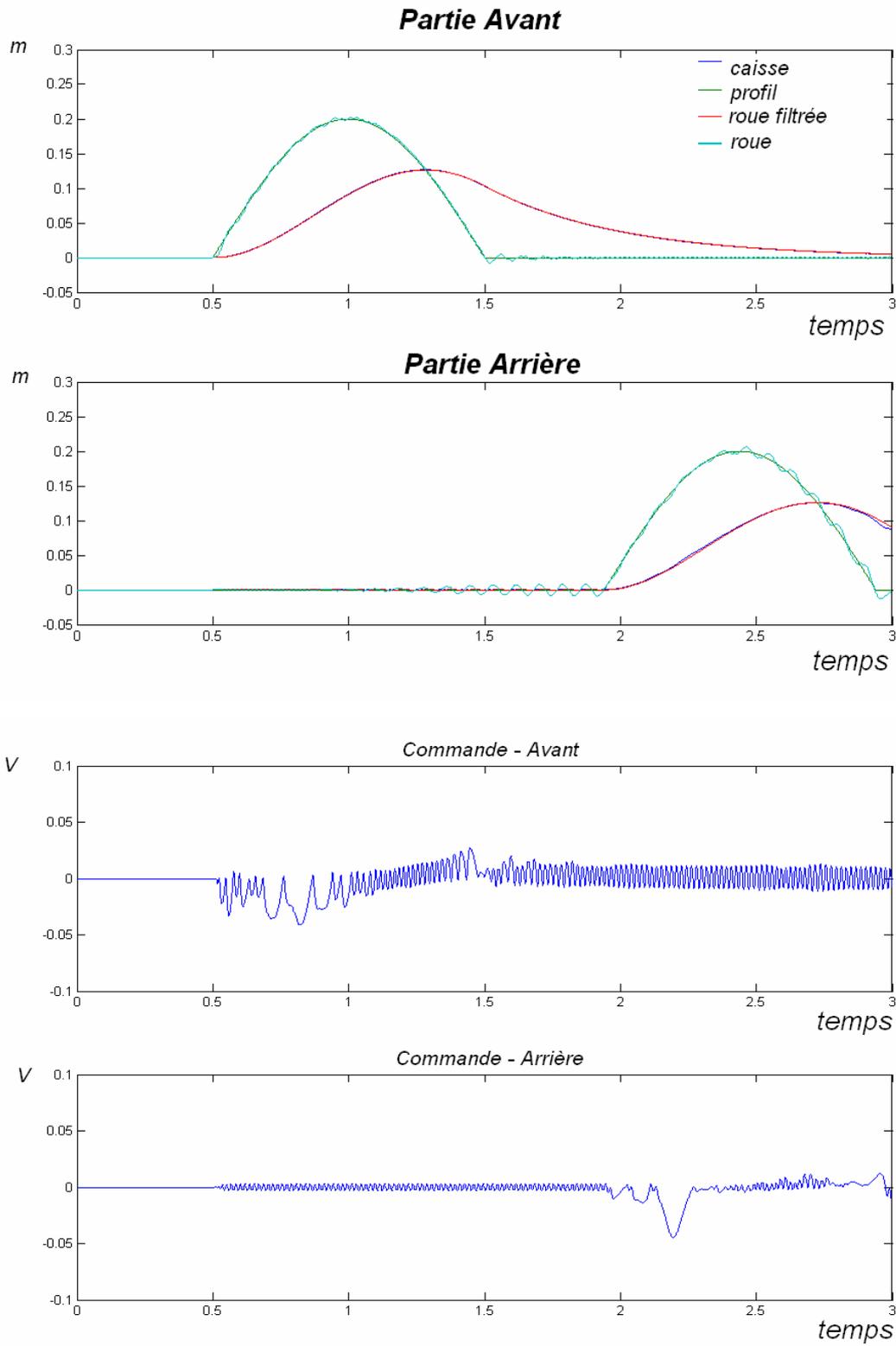


Figure III.6 : Réponse du modèle demi véhicule (dos d'âne) $\varepsilon = 2$

III.2.2.2 configuration « tenue de route optimal » $\varepsilon = 20$

i. Montée sur trottoir

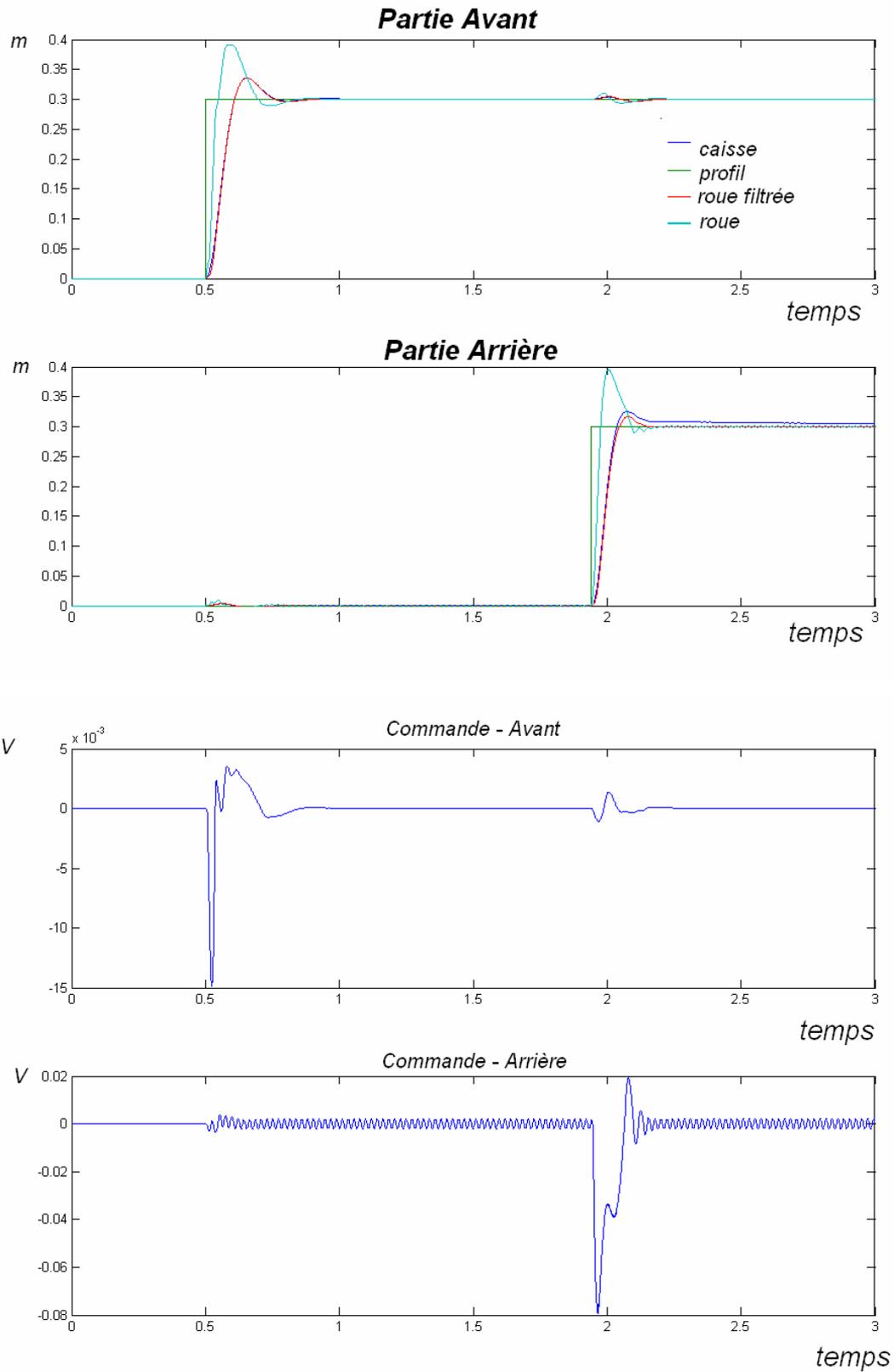


Figure III.7 : Réponse du modèle demi véhicule (trottoir) $\varepsilon = 20$

ii. Passage sur dos d'âne

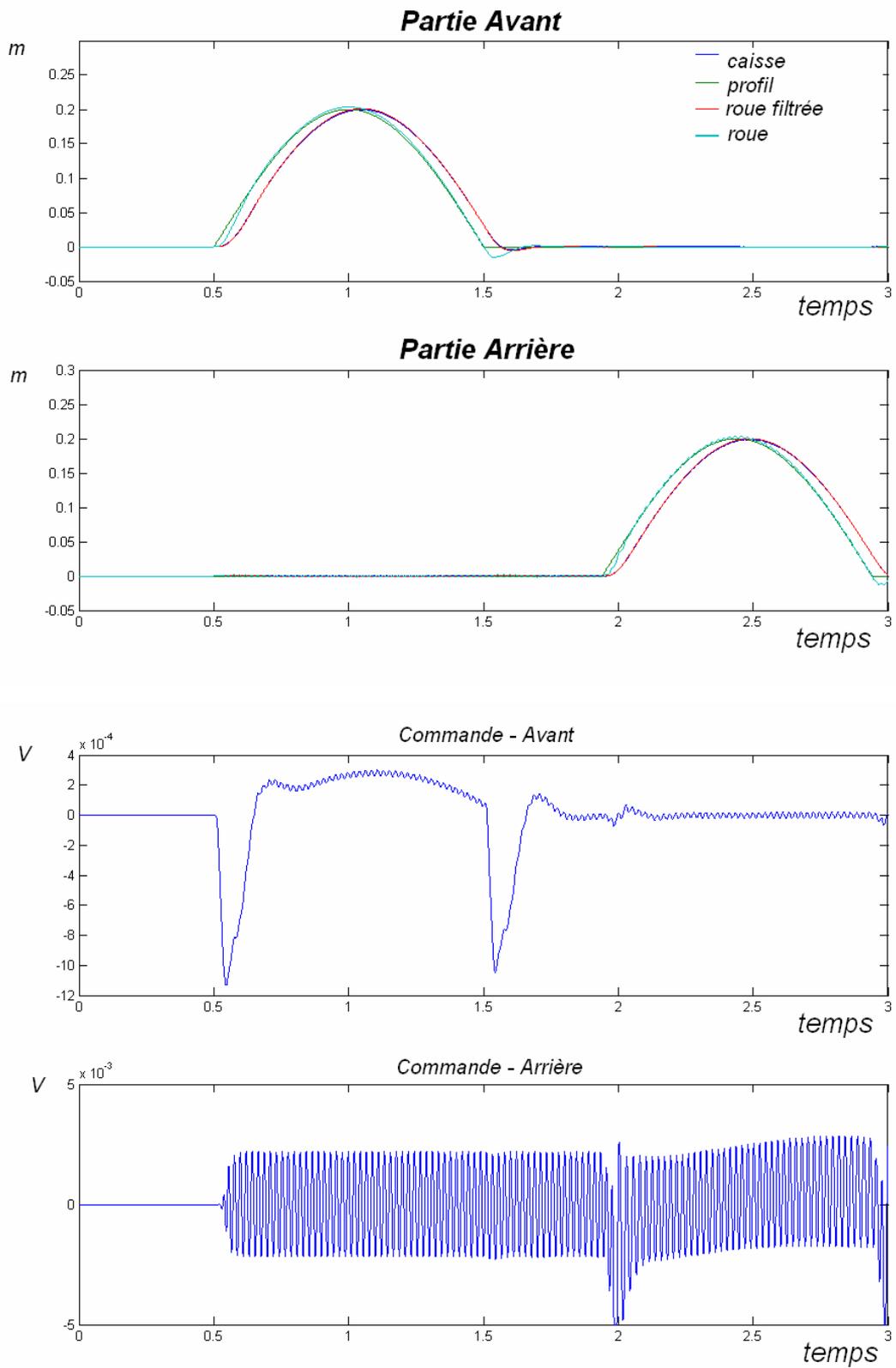


Figure III.8 : Réponse du modèle demi véhicule (dos d'âne) $\varepsilon = 20$

III.2.2.3 configuration « hybride » $\varepsilon = 5$

i. Montée sur trottoir

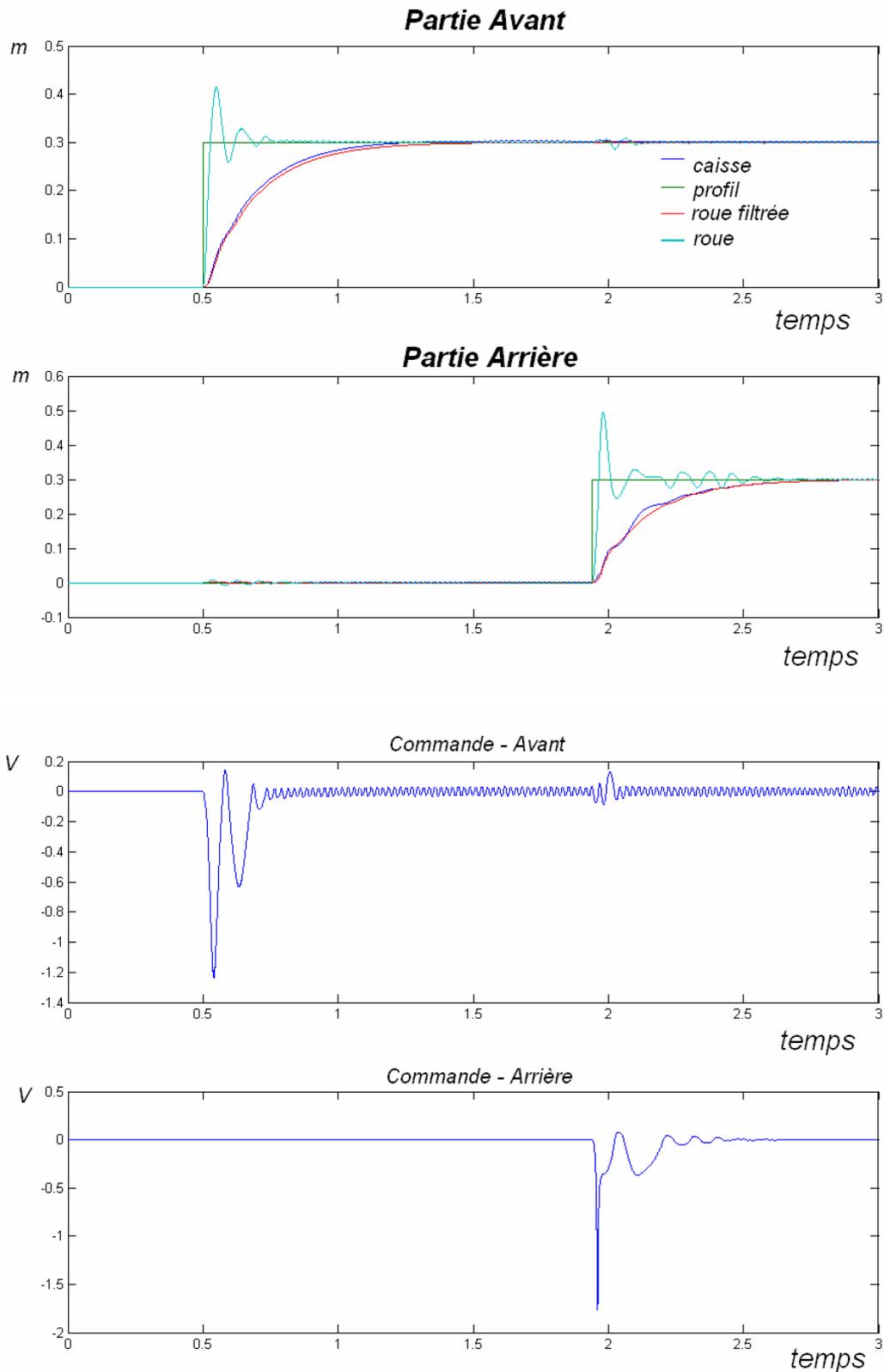


Figure III.9 : Réponse du modèle demi véhicule (trottoir) $\varepsilon = 5$

ii. Passage sur dos d'âne

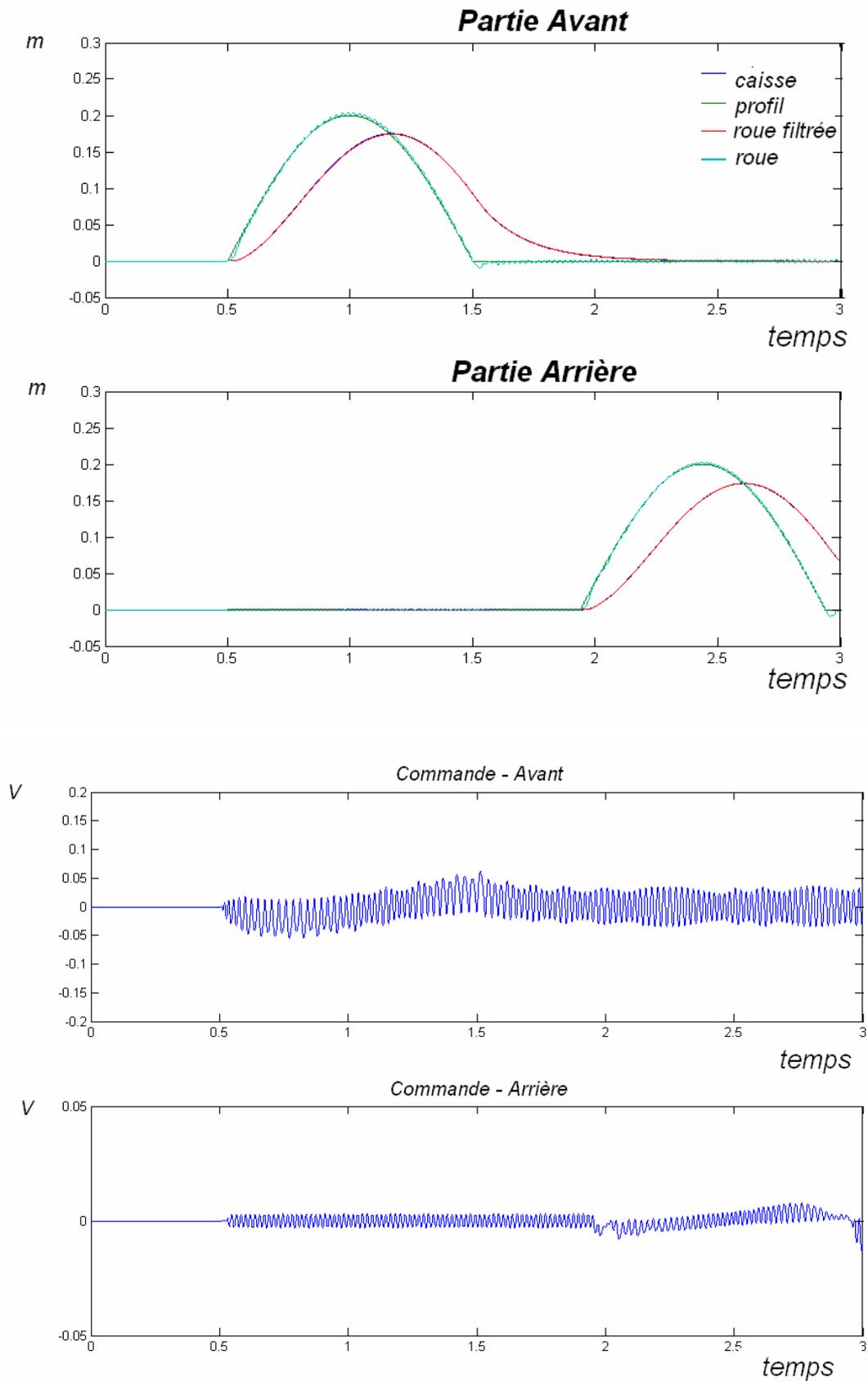


Figure III.10 : Réponse du modèle demi véhicule (dos d'âne) $\varepsilon = 5$

III.2.2.4 Réponse de la suspension passive aux perturbations :

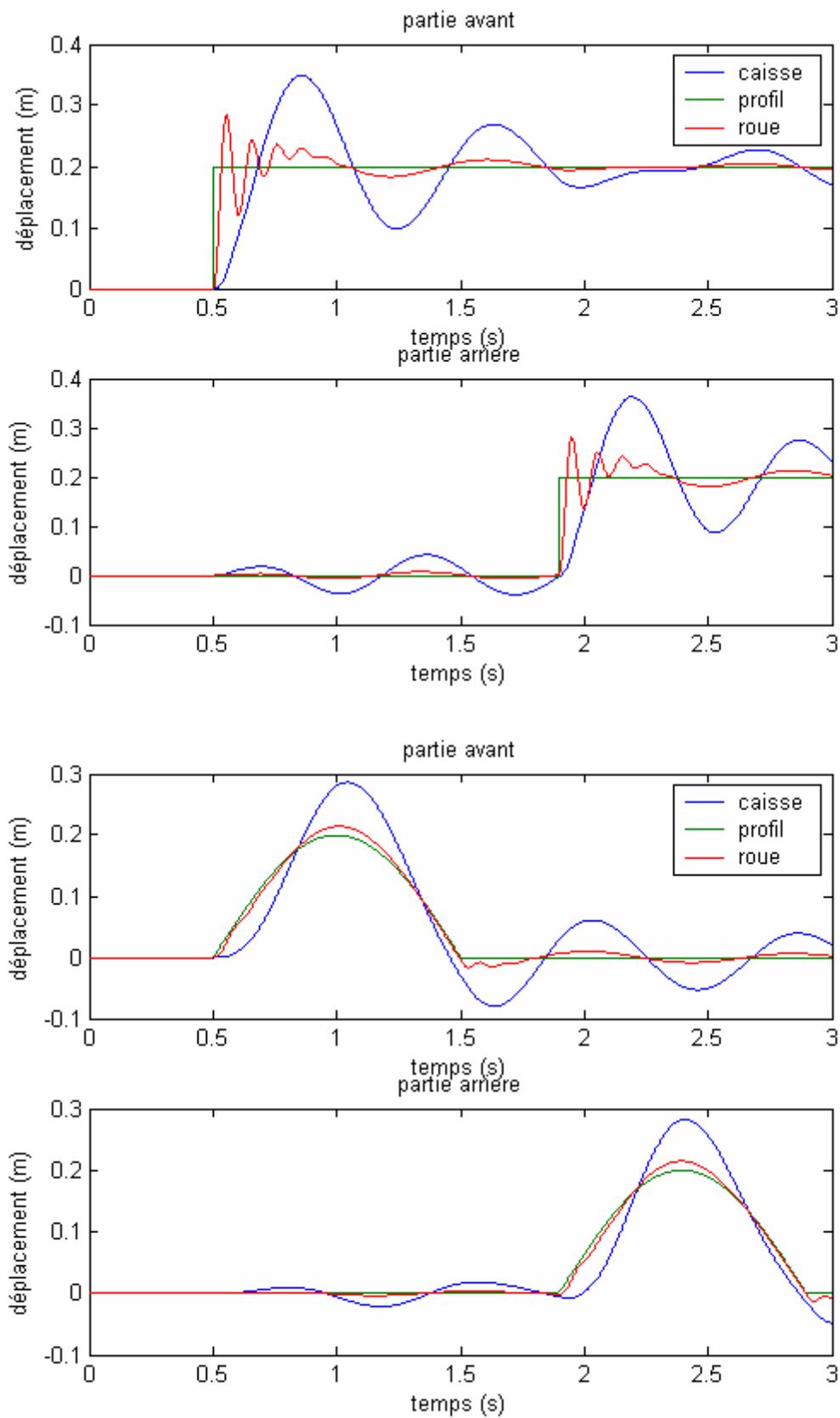


Figure III.11 : Réponse du modèle demi véhicule / suspension passive

III.2.2.5 comparaison des valeurs des critères de performance

Configuration \ Critère	$\varepsilon = 2$	$\varepsilon = 5$	$\varepsilon = 20$	passive
Tenue de Route	0.28249	0.11305	0.04480	0.41501
Confort	0.54571	0.64445	0.68656	0.76391
Energie de commande	2.61519	0.85554	0.04052	X

III.2.2.6 Remarques

- En plus des remarques précédentes relatives au modèle quart de véhicule, on a observé une augmentation de la durée de l'apprentissage et ceci est dû au fait que lors de l'apprentissage ; on introduit des perturbations sur les deux roues, ce qui fait intervenir l'interconnexion ; et l'effet des perturbation de chaque coté de la caisse sur l'autre, et ceci introduit des contraintes supplémentaires à minimiser lors de l'apprentissage, ce qui augmente sa durée.
- Ceci nous permet de confirmer que le modèle demi véhicule est plus représentatif que le modèle quart de véhicule du moment où il permet de prendre en considération l'effet de l'interconnexion.

III.2.3 modèle véhicule complet – vitesse 10 km/h

III.2.3.1 configuration « confort optimal » $\varepsilon = 2$

i. Montée sur trottoir

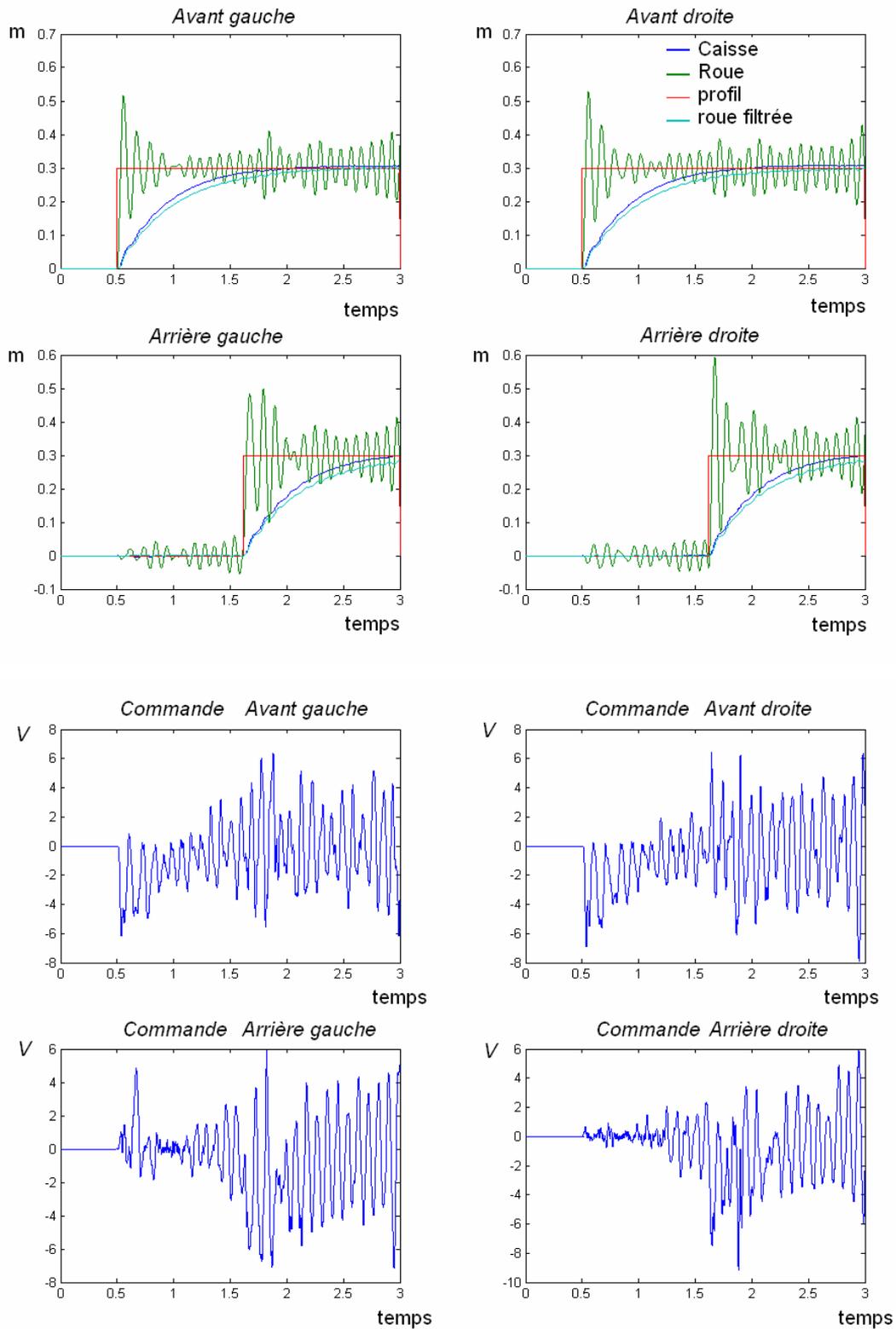


Figure III.12 : Réponse du modèle 3d / $\varepsilon = 2$ - montée sur trottoir

ii. Passage sur dos d'âne

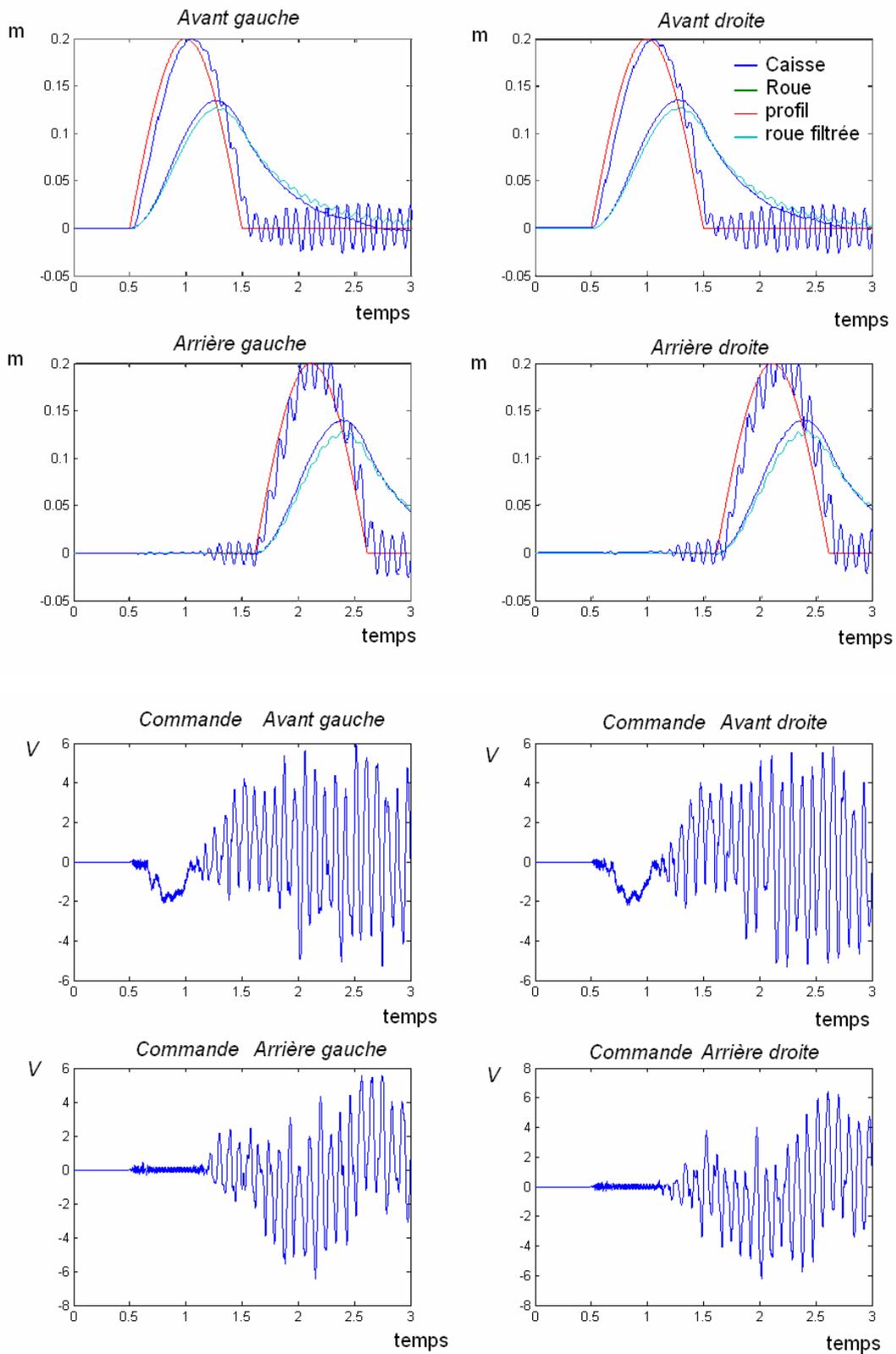


Figure III.13 : Réponse du modèle 3d / $\varepsilon = 2$ - dos d'âne

III.2.3.2 configuration « tenue de route optimal » $\varepsilon = 20$

i. Montée sur trottoir

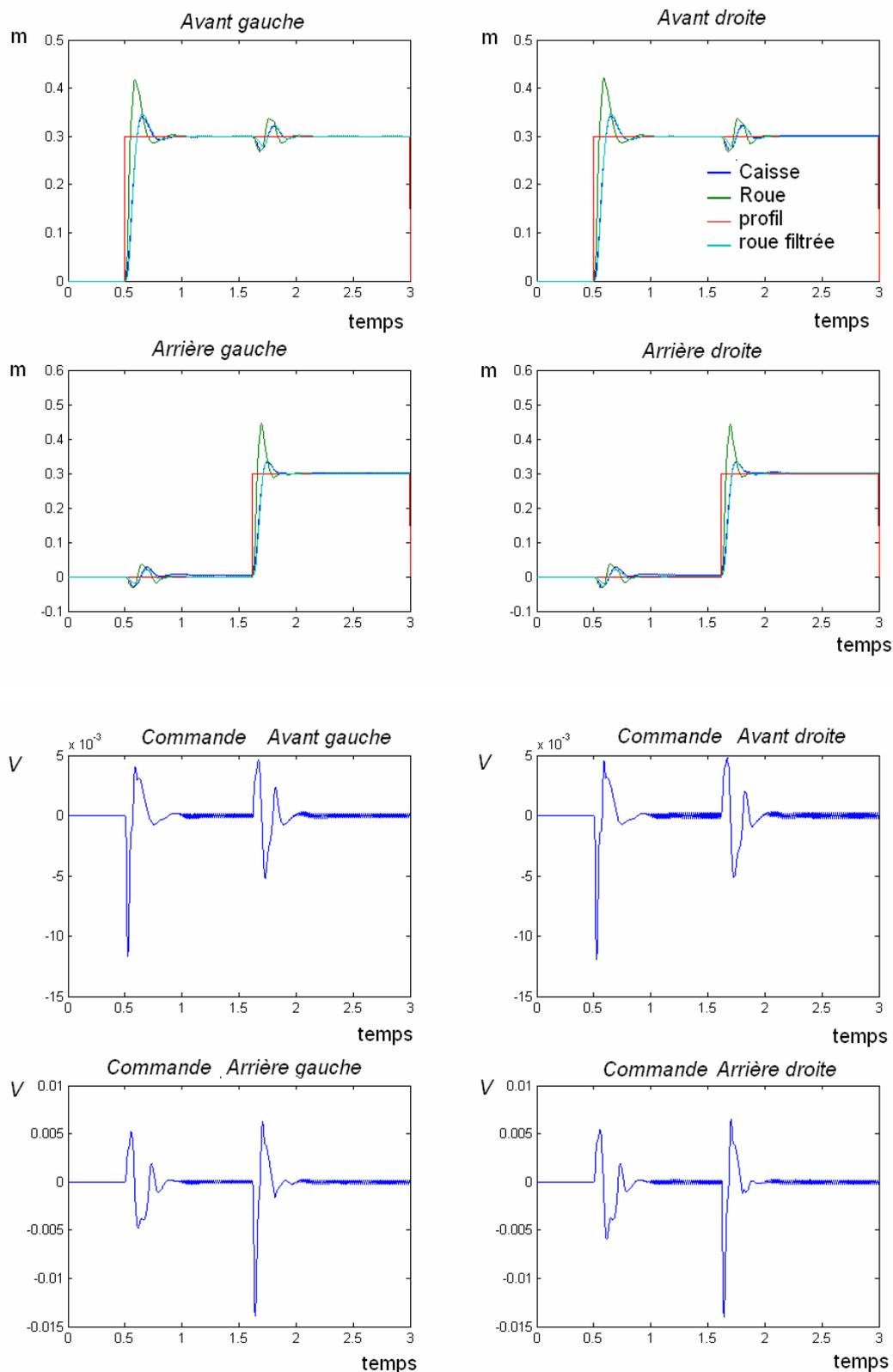


Figure III.14 : Réponse du modèle 3d / $\varepsilon = 20$ - trottoir

ii. Passage sur dos d'âne

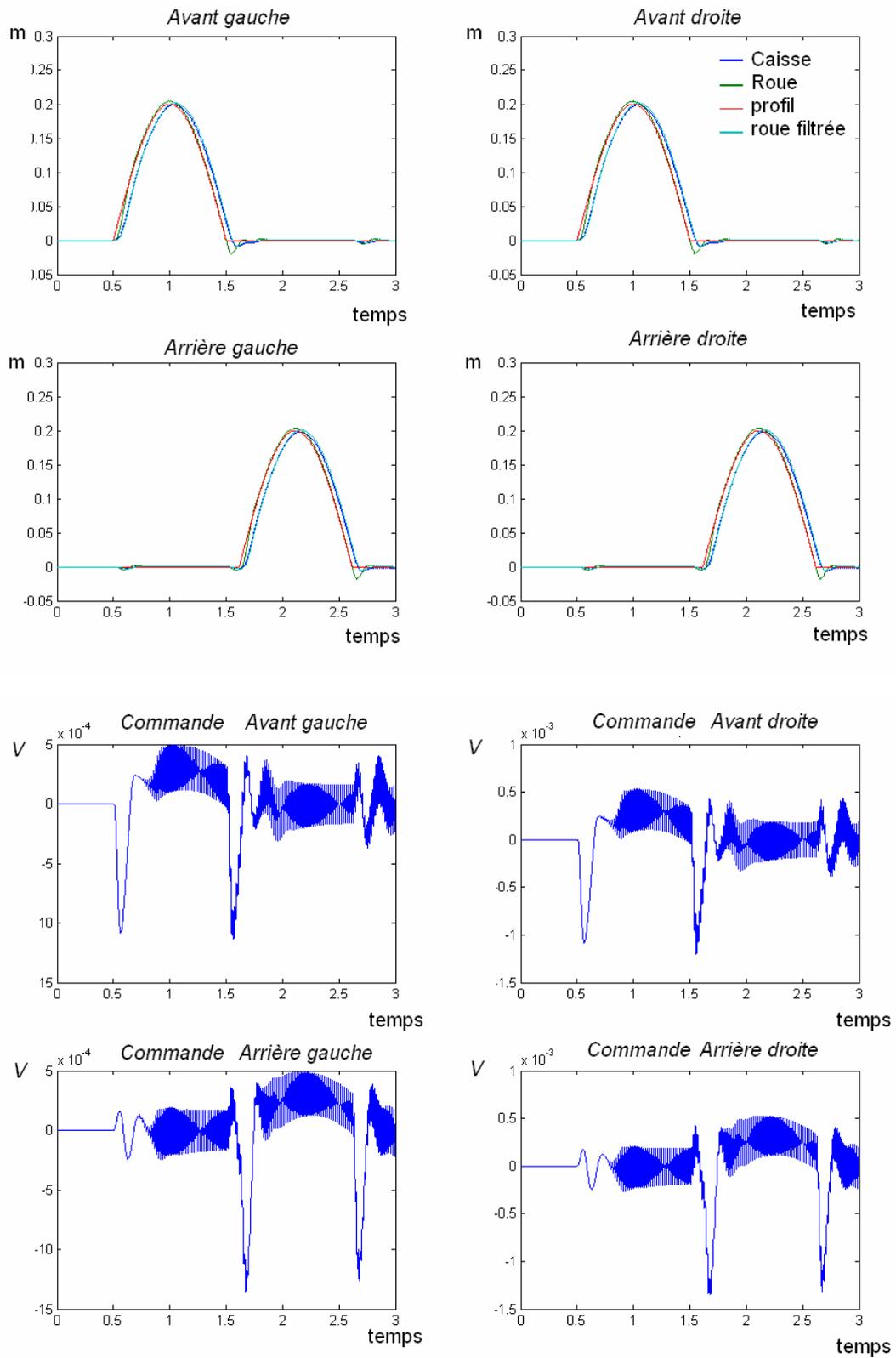


Figure III.15 : Réponse du modèle 3d / $\varepsilon = 20$ - dos d'âne

III.2.3.3 configuration « hybride » $\varepsilon = 5$

i. Montée sur trottoir

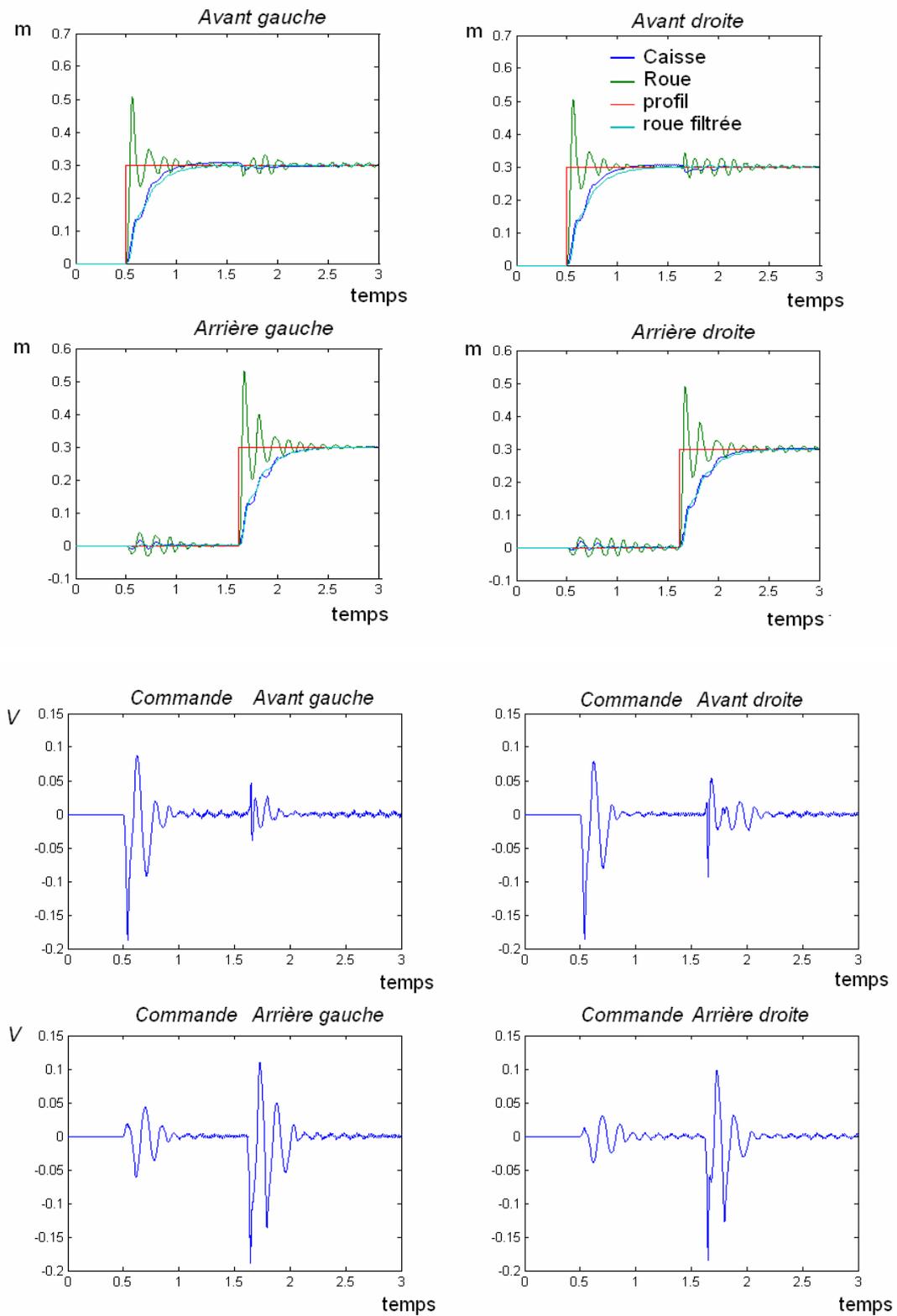


Figure III.16 : Réponse du modèle 3d / $\varepsilon = 5$ - trottoir

ii. Passage sur dos d'âne

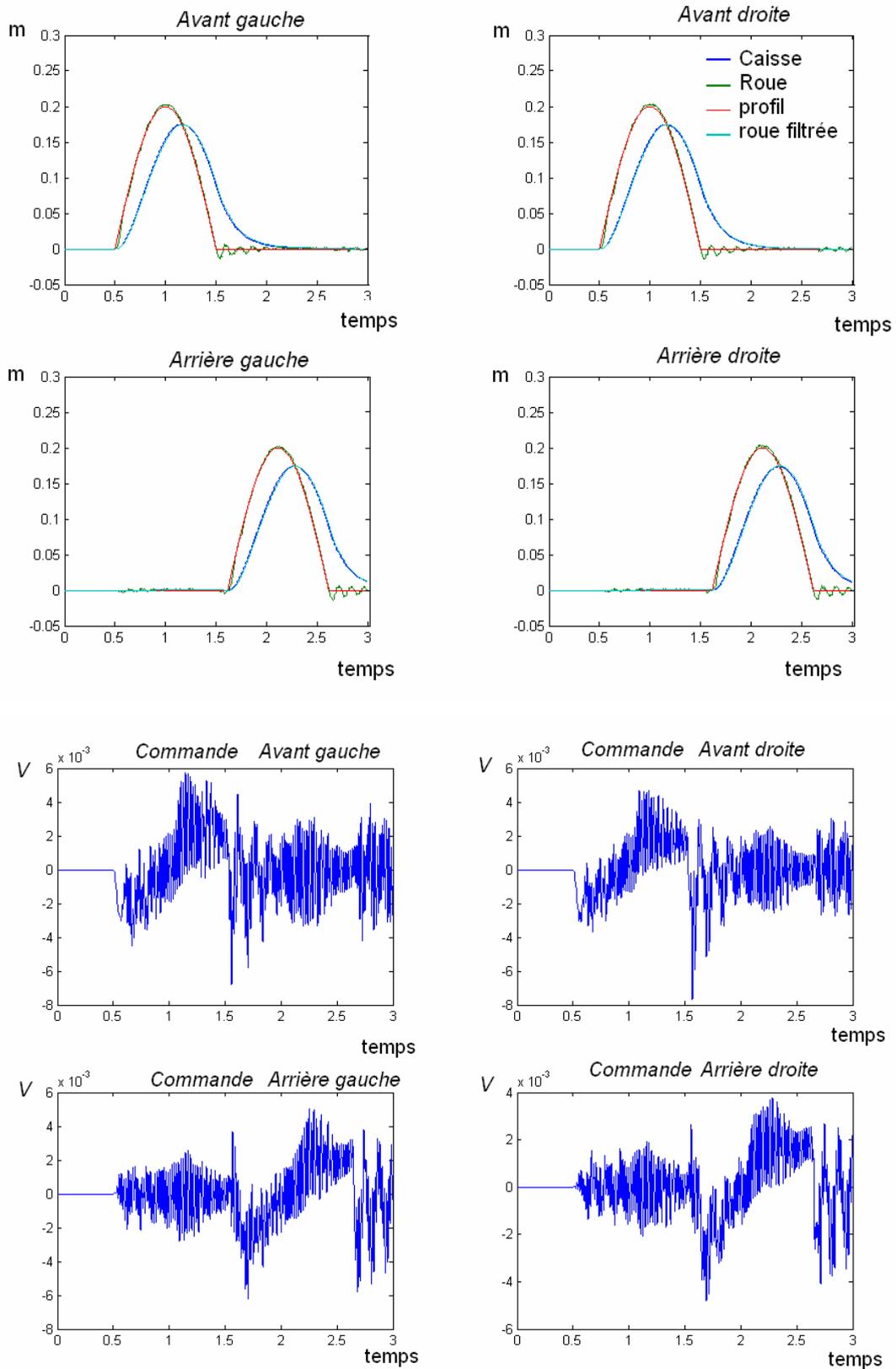


Figure III.17 : Réponse du modèle 3d / $\varepsilon = 5$ - dos d'âne

III.2.3.4 réponse de la suspension passive

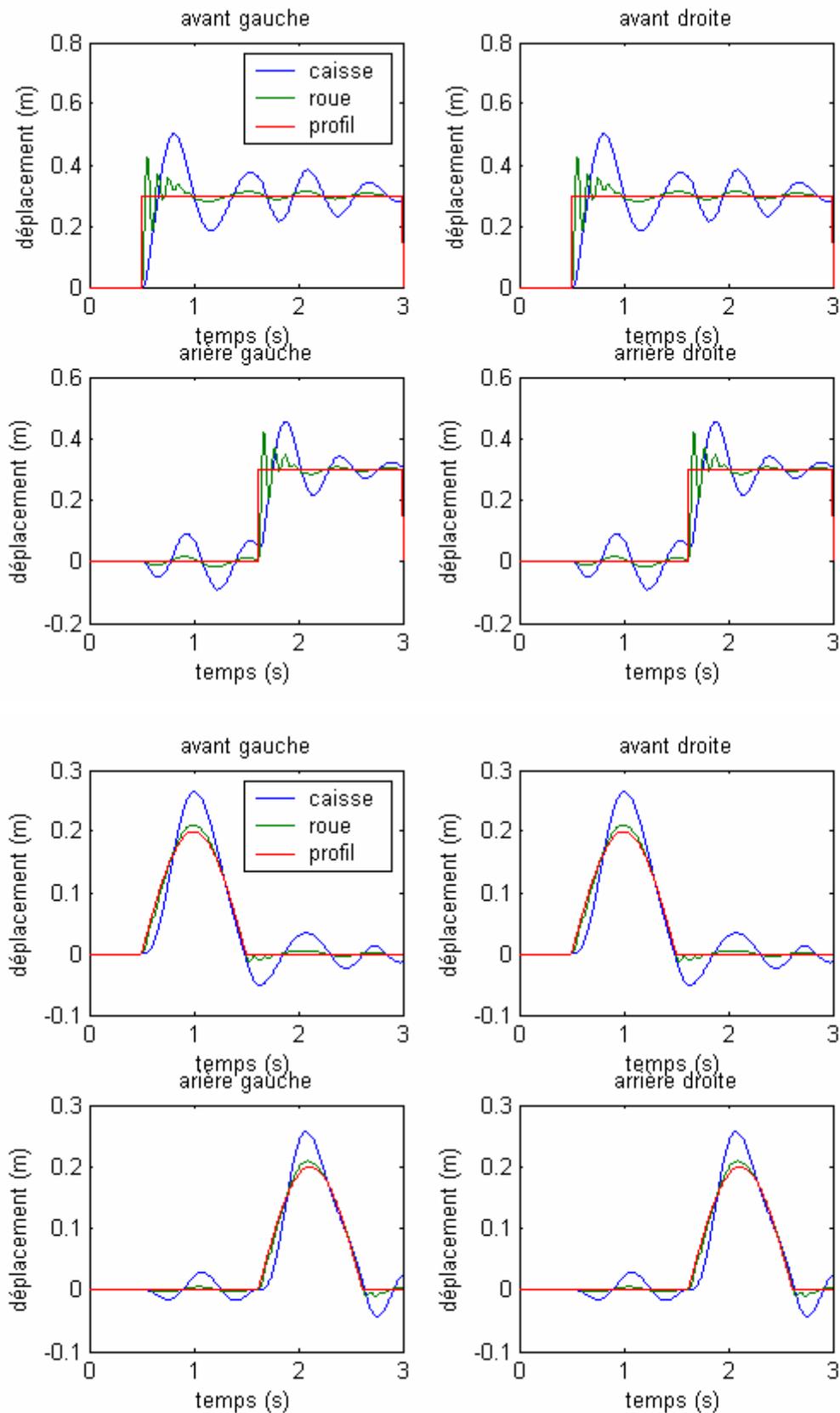


Figure III.18 : Réponse du modèle 3d suspension passive

III.2.3.5 comparaison des valeurs des critères de performance

Configuration \ Critère	$\varepsilon = 2$	$\varepsilon = 5$	$\varepsilon = 20$	passive
Tenue de Route	0.25151	0.12694	0.08375	0.52097
Confort	0.59403	0.65696	0.69573	0.74645
Energie de commande	61.17061	0.25722	0.01786	X

III.2.3.6 Remarques

- Pour ce modèle les durées d'apprentissage sont devenues très importantes, surtout pour $\varepsilon = 2$ et ceci est dû à la prise en compte de toutes les perturbations agissant sur le système ; les tests suivants vont servir à vérifier que les quatre régulateurs interviennent lorsque une perturbation affecte une roue, permettant de filtrer cette perturbation d'une façon optimale et garantissant les meilleures performances au système.

III.2.4 Tests d'interconnexion

i. Trottoir sur roue avant gauche avec $\varepsilon = 20$

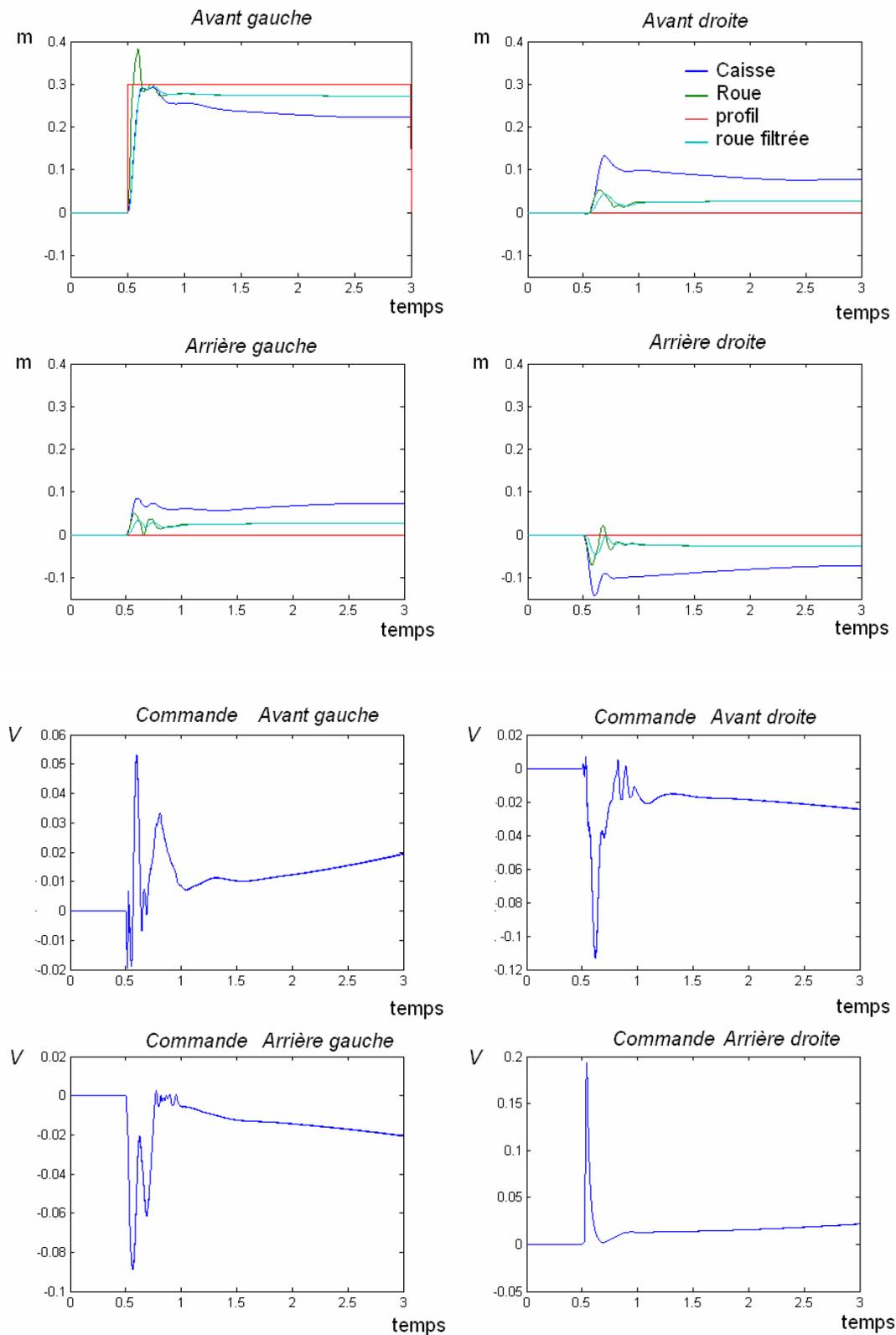


Figure III.19 : Réponse du modèle 3d $\varepsilon = 20$

ii. Bosse sur roue avant gauche avec $\varepsilon = 5$

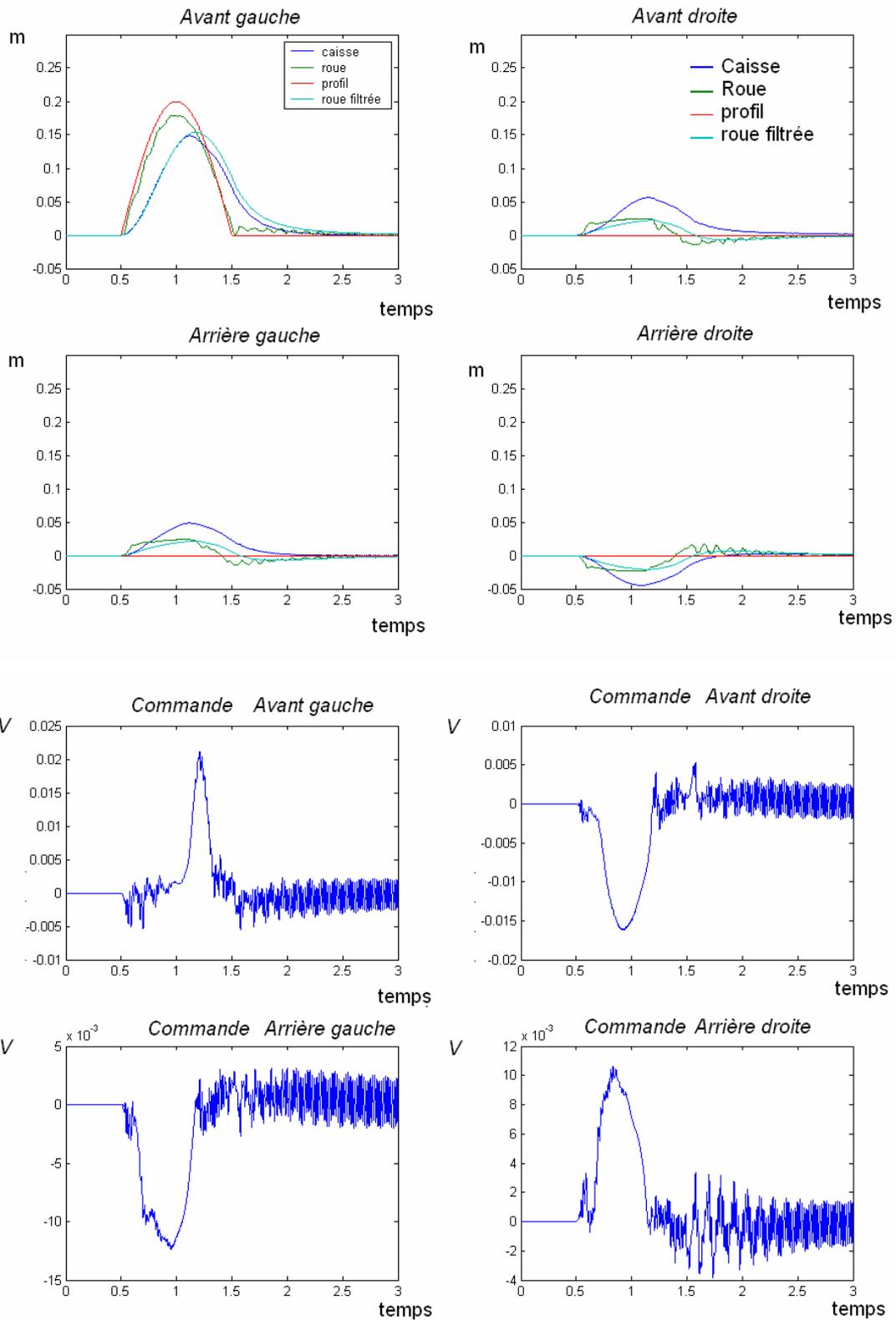


Figure III.20 : Réponse du modèle 3d $\varepsilon = 5$

III.2.5 Test de robustesse vis-à-vis des paramètres

III.2.5.1 modèle quart de véhicule et charge = +100 kg / $\varepsilon = 2$

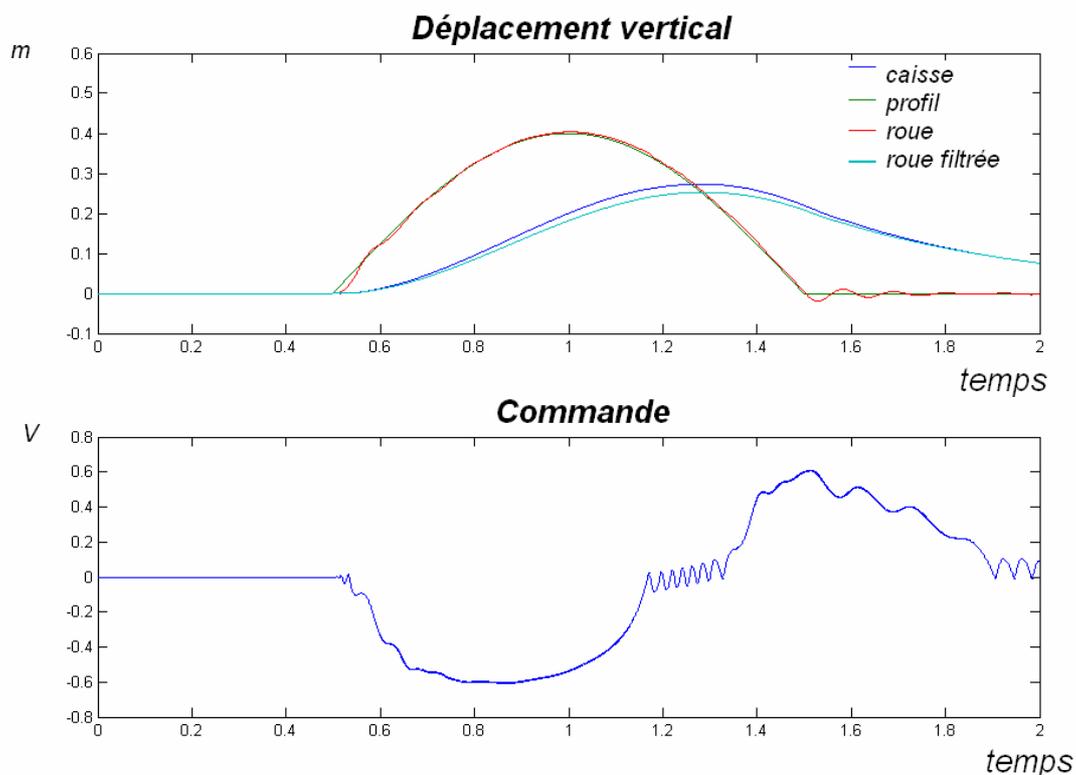
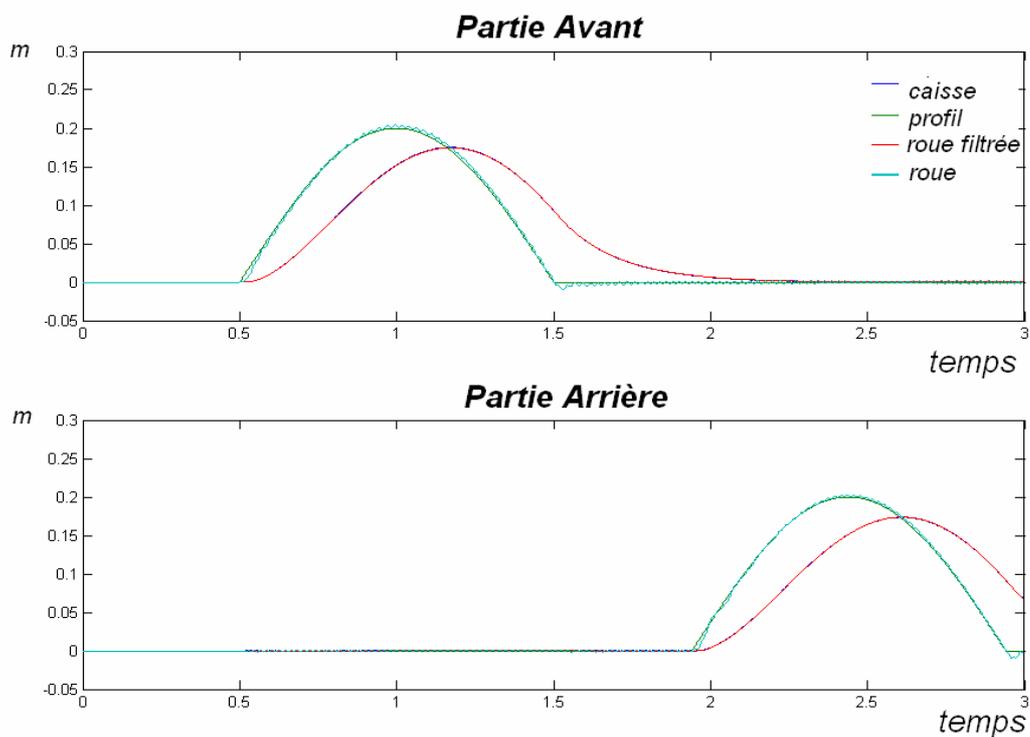


Figure III.21 : Réponse du modèle quart de véhicule et $\varepsilon = 2$

III.2.5.2 modèle demi véhicule et charge = +200 kg / $\varepsilon = 5$



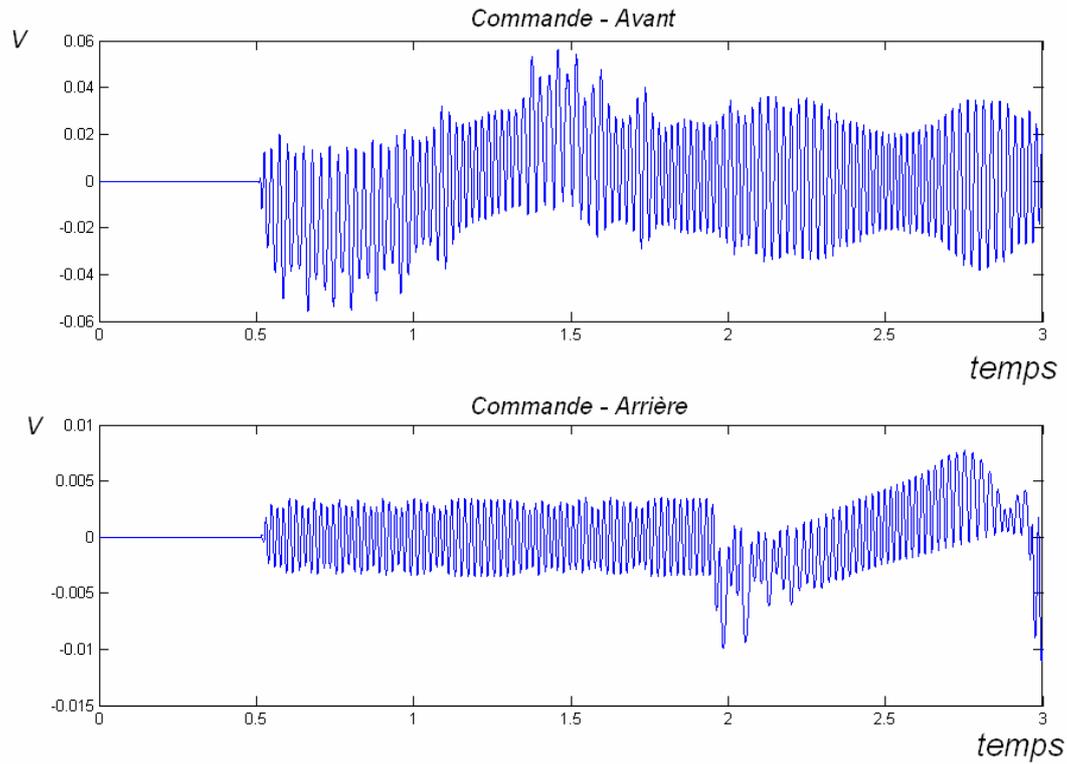
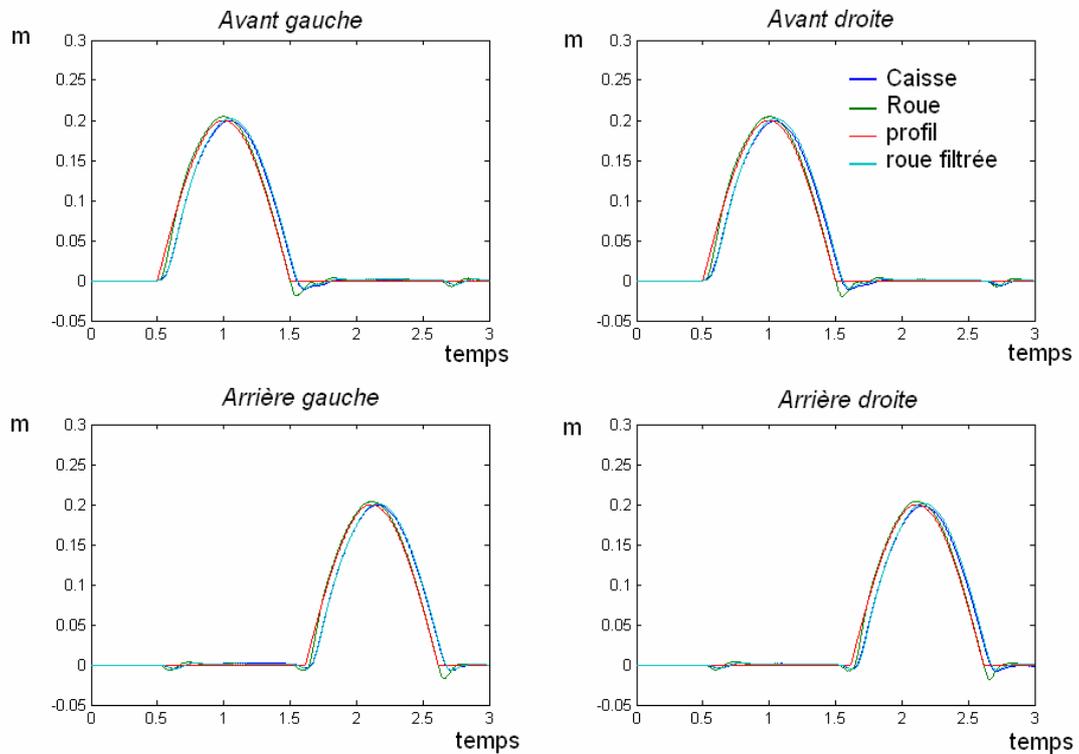


Figure III.22 : Réponse du modèle demi véhicule $\varepsilon = 5$

III.2.5.3 modèle véhicule complet et charge = +400 kg / $\varepsilon = 20$



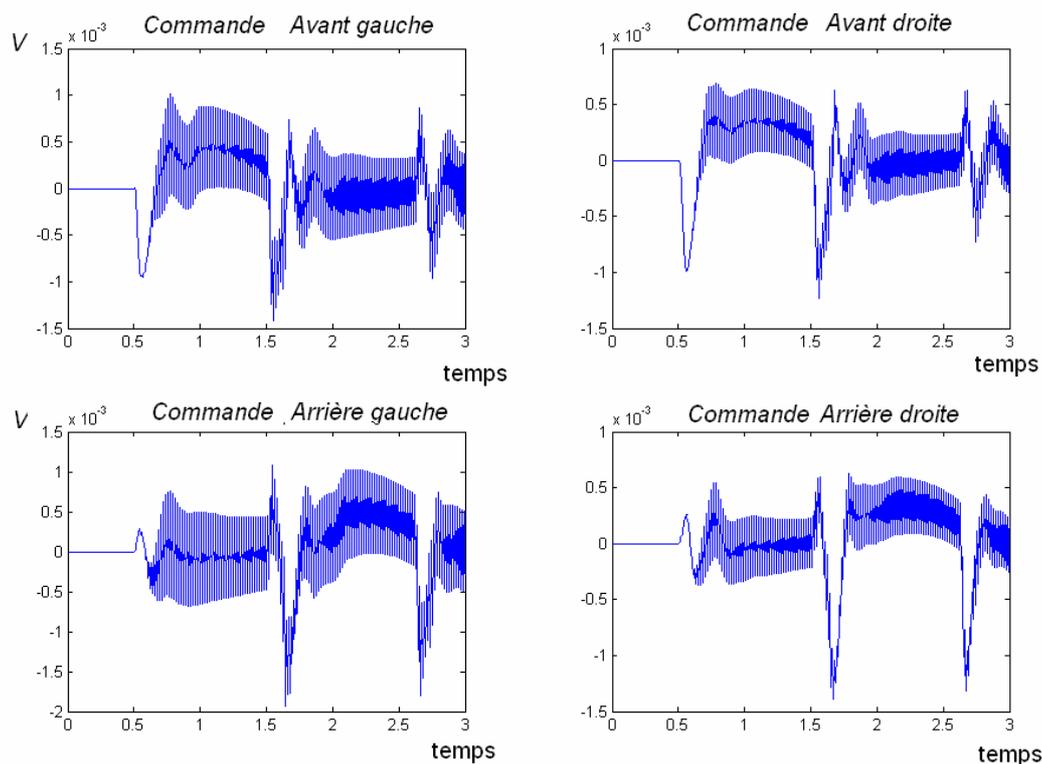


Figure III.20 : Réponse du modèle 3d $\varepsilon = 20$

III.2.5.4 Remarques

A cause de la structure des régulateurs (surtout leur entrées) et les propriétés de généralisation des RNA ainsi que le raisonnement flou utilisé dans la conception des régulateurs on constate que le système est robuste vis-à-vis de la variation paramétrique.

III.3 Conclusions et remarques

a) Amélioration de la qualité de la suspension

Les simulations démontrent pertinemment que la suspension active dans toutes ses configurations présente des réponses meilleures que la suspension passive du point de vue confort et tenue de route ; ceci est illustré à travers les graphes ainsi que les critères de performance.

b) A propos du choix de la configuration (ε)

Le choix de ε donc de la qualité de la suspension dépend de l'utilisation du véhicule, l'environnement où il évolue ; mais doit prendre aussi en compte l'énergie de la commande et la tenue de route qui sont très affectées par ce choix.

L'énergie de commande est très importante pour $\varepsilon=2$ ainsi que sa dynamique ce qui risque d'avoir un mauvais effet sur l'actionneur ; ceci est la conséquence du rejet d'un grand taux de perturbations.

c) A propos de la complexité des modèle

La complexité du modèle utilisé améliore sa réponse aux perturbations réelles mais introduit des problèmes au niveau de l'apprentissage (durée et optimums locaux)

d) A propos de la robustesse des régulateurs Neuroflous

Le réglage neuroflou est caractérisé par une grande robustesse vis-à-vis des changements paramétrique que peut subir le système (usure, changement de charge, conditions climatiques...) ceci est dû à deux raisons principales :

- généralisation des RN
- raisonnement approximatif des régulateurs flous

e) A propos de la durée de l'apprentissage des RN flous

Lors de la phase d'apprentissage on remarque que la durée augmente lorsque ε diminue, ceci peut s'expliquer par la grandeur des efforts délivrés par l'actionneur et le filtrage poussé des perturbations affectant le système, ainsi la commande doit être rapide et énergique ce qui impose un bon ajustage des paramètres du régulateur et ceci se traduit en terme de durée d'apprentissage.

**Conclusions et
perspectives**

Conclusion Générale

A travers ce travail on a développé nos connaissances sur plusieurs points ; ainsi quelques remarques et conclusions s'imposent :

- La modélisation des systèmes dynamiques est une étape très importante dans tout travail visant à commander ou étudier simplement ces systèmes ; les modèles ne sont – en général- pas uniques, le choix d'un modèle dépend d'un certain nombre de facteurs, entre autres le niveau de précision voulu, dans notre cas le modèle complet à 7 degrés de liberté a donné les meilleurs résultats car il prend en compte certaines grandeurs ignorées par les modèles plus réduits (demi et quart) mais au prix d'une complexité dans l'élaboration, la simulation et surtout la commande (durant l'apprentissage des RN), de plus, au cas où notre système sera appelé à faire des calcul du type temps réel lors d'une implémentation pratique, il aura comme contrainte supplémentaire le temps de calcul qui doit être optimisé.
- La technique neurofloue, apparaît comme une alternative intéressante pour la commande des système à large échelle (LSS) où les systèmes non linéaires, du moment où elle ne se base pas sur le modèle du système étudié (commande de type boîte noire) , les calculs sont sensiblement réduits ; et l'apprentissage assure l'obtention des paramètres optimaux, mais elle nécessite cependant quelques connaissances sur le système (dans notre cas les ensembles flous des entrées) , ce problème peut être contourné en les incluant dans le processus d'apprentissage au prix d'une complexité accrue et un risque supérieur de tomber dans les optimums locaux, et d'arrêter prématurément l'entraînement du RNA.
- La comparaison des résultats obtenus avec la réponse de la suspension active a permis de déduire que la qualité de la suspension sur tous les plans (sécurité et confort) est améliorée, mais l'énergie de commande (apport) peut être très importante ce qui peut être un inconvénient ; toutefois l'amélioration de la qualité du comportement du véhicule peut justifier l'emploi de tels systèmes, de plus leurs modularité et adaptabilité aux conditions extrêmes d'utilisation laisse à prévoir un grand développement dans ce domaine dans les années à venir.

- Le compromis sécurité/confort a été confirmé lors de nos simulations, le choix de la configuration dépend donc des conditions d'utilisations du véhicule, on favorise alors un critère par rapport à l'autre.
- la structure de commande décentralisée nous a permis d'obtenir des résultats satisfaisants, et sa robustesse par rapport aux interconnexions des perturbations a été prouvée lors des simulations. Par rapport à la structure centralisée, elle présente l'avantage d'être plus simple en terme de complexité ; surtout pour les système à large échelle (LSS : large scale systems) , en plus dans notre cas , elle nous a permis de tirer profit de la symétrie du système ($4 \times \frac{1}{4}$).

Perspectives

Enfin il ne reste qu'à présenter les perspectives qui pourront être développées sur la base des résultats obtenus dans ce travail :

- Le choix de ε , et donc de la configuration de la suspension peut faire l'objet d'une commande dynamique basée sur plusieurs critères tels que l'état de la chaussée, vitesse du véhicule, sa charge, sa fonction..... la logique floue peut se prêter à de telle décisions à cause de sa structure basée sur le raisonnement approximatif.
- Les modèles peuvent encore être améliorés en prenant en compte le mouvement longitudinal / transversal du véhicule et les modélisations des pneumatiques, afin de concevoir des commandes plus robustes et efficaces.
- Les algorithmes d'apprentissage des RN peuvent être optimisés par l'emploi de techniques telles que les algorithmes génétiques afin d'accélérer l'apprentissage et d'éviter les optimums locaux.
- Etendre la commande à d'autres organes (freinage, direction, motorisation ...) afin d'améliorer d'une façon globale le rendement de l'automobile.
- Implémentation pratique du système obtenu à l'aide de calculateurs (microprocesseurs) afin de tester l'efficacité réelle du régulateur.

**Références
Bibliographiques**

- [1] **Microsoft Encarta** '*Systèmes de suspension*' – **2006 Microsoft corporation**
- [2] **Chakhchoukh Yacine**
'*Modélisation et commande d'une suspension active par logique floue*' **PFE ENP 2004**
- [3] **Moussous Younes**
'*Modélisation et commande de suspension active par réseaux de neurones*' **PFE ENP 2004**
- [4] **Fu-Cheng Wang** '*Design and synthesis of active and passive vehicle suspensions* '
University of Cambridge Septembre 2001
- [5] **S. Ikenaga, F.L. Lewis & J. Compos**
'*Active suspension control of ground vehicle based on a full car model*'
University of Texas – Journal of automotive systems 2001
- [6] **Damien Samier**
'*Sur la modélisation et commande de suspension de véhicules automobiles*'
INPG thèse de doctorat -Novembre 2001
- [7] **M.A Zeroual**
'*Application de la commande neuro-floue à structure décentralisée au robot PUMA 560*'
PFE ENP 1998
- [8] **A .Stibersky, K.Hyniova, J.Honcu & A.Kruczek**
'*Using Fuzzy logic to control active suspension system of one-half car model*'
Czeck technical university 2002
- [9] **R.Rajamani & J.K.Hedrick**
'*Performance of active automotive suspension with hydrolic actuator : theory and experiment*'
University of California 1994
- [10] **Wuwei Chen – James K. Mills & Le Wu**
'*Neurofuzzy and Fuzzy control of automotive semi active suspensions*'
International journal of vehicle autonomous systems -2003-
- [11] **Gerald Huguenin** '*Logique floue & processeurs flous* '
/ EIAJ-GHU mars 2004

- [13] **C.Pelligrinc** ‘*Neurones Artificiels : Modèles et réseaux*’
IA publications 2004-2005
- [14] **Dr Grappa** ‘*Apprentissage Automatique : Les Réseaux de neurones*’
Support de cours – Université de LILLE 2005
- [15] **Claude Touzet**
‘*Réseaux de neurones artificiels -introduction au connexionnisme*’
Editions Ellipses -1992
- [16] **Michael Negnevitsky** ‘*Systèmes Hybrides*’
Economic Research Department / University of California 2005
- [17] **Rachid Ladjadj** ‘*Sur les Réseaux de neurones*’
– complément de cours IR 3^{ème} Année UPFL
- [20] **Nabila Bouncer**
‘*Commande NeuroFloue optimisée par algorithme génétique
d’un bras manipulateur flexible*’ **PFE ENP / 2005**
- [21] **Aouchar Nadjjet / Bekhaled Chafika**
‘*Application des réseaux neuroflous au diagnostique des huiles de
transformateurs*’
-PFE ENP / 2005

Liens Internet :

- [12] www.wikipedia.com ‘Wikipédia L’encyclopédie libre’
GNU free documentation licence
- [18] www.umoncton.ca/science/informatique
‘*Réseaux de neurones artificiels*’
Neural.net /
- [19] www.sciences.ch/htmlfr/infoteorique/infologiquefloue01.html
‘*Info théorique – logique floue*’

ملخص

إن التحكم اللامركزي أكثر استعمالاً في الأنظمة المعقدة، بسبب بساطته مقارنة مع التحكم المركزي. تسمح تقنية الشبكات العصبية الغامضة بتصميم أجهزة تحكم دون الرجوع إلى نموذج النظام، خاصة في الأنظمة اللاخطية. في هذا العمل قمنا بتحسين الخصائص الرئيسية لنظام تعليق السيارات (الراحة والأمن) باستعمال جهاز تحكم ذكي ذو بنية غير مركزية على النموذج الكامل الذي يعتبر كنظام معقد يمكن تجزئته إلى عدة أنظمة أبسط، وقد أثبتنا نجاعته تجاه تأثير الترابط ومختلف الاضطرابات.

كلمات مفتاحية: المنطق الغامض، الشبكات العصبية الاصطناعية، التحكم العصبي الغامض، بنية ANFIS، الأنظمة اللاخطية، الأنظمة المعقدة، الترابط، نموذج السيارة الكامل، التعليق النشط، التحكم اللامركزي.

Abstract

Decentralized control is more adapted to large scale systems, because of its simplicity compared to the centralized control. Neurofuzzy technics allow designing controllers independently of system structure, which can be interesting in the case of nonlinear systems. In this work we improved essential qualities of vehicle suspension (comfort and security) by using an intelligent decentralized controller on the full car model which is a large scale system and can be decomposed in several sub-systems, we proved its robustness facing to the interconnections and various disturbances.

Keywords: Fuzzy logic, neural networks, neurofuzzy control, ANFIS structure, training, nonlinear systems, large scale systems (LSS), interconnection, full car model, active suspension, decentralized control.

Résumé

La commande décentralisée est mieux adaptée aux systèmes à large échelle, à cause de sa simplicité relative par rapport à la commande centralisée. La technique neurofloue permet de concevoir des commandes indépendantes des paramètres des systèmes, ce qui peut être intéressant dans le cas des systèmes non linéaires.

Dans ce travail nous nous sommes intéressés à l'amélioration des qualités essentielles de la suspension de véhicule- à savoir le confort et la sécurité- en utilisant une commande intelligente à structure décentralisée sur le modèle de véhicule complet qui est un système à large échelle décomposable en plusieurs sous systèmes, nous avons prouvé la robustesse de notre commande par rapport aux interconnexions ainsi qu'aux différentes perturbations.

Mots-clés : Logique floue, réseaux de neurones, commande neurofloue, structure ANFIS, apprentissage, systèmes non linéaires, systèmes à grande échelle (LSS), interconnexion, modèle véhicule complet, suspension active, commande décentralisée.