

## **Treball final de màster**

**Estudi: Màster en Enginyeria Industrial**

**Títol: Simulador del comportament dinàmic d'automòbils**

**Document:** Memòria i Annexos

**Alumne:** Daniel Durbau Pimas

**Tutor:** Daniel Trias Mansilla

**Departament:** Enginyeria mecànica i de la construcció industrial

**Àrea:** Enginyeria mecànica

**Convocatòria (mes/any):** Setembre 2019

# Índex de Contingut

<b>ÍNDEX DE CONTINGUT</b> .....	<b>2</b>
<b>INDEX DE FIGURES</b> .....	<b>4</b>
<b>1. INTRODUCCIÓ</b> .....	<b>6</b>
1.1. ANTECEDENTS .....	6
1.2. OBJECTE .....	6
1.3. ABAST .....	6
<b>2. FONAMENT TEÒRIC</b> .....	<b>7</b>
2.1. MOVIMENT DEL VEHICLE .....	7
2.1.1. <i>Dinàmica del moviment rectilini</i> .....	7
2.1.2. <i>Dinàmica dels pneumàtics</i> .....	8
2.1.3. <i>Dinàmica de la transmissió</i> .....	11
2.2. CINEMÀTICA DEL VEHICLE .....	13
2.2.1. <i>Masses Principals</i> .....	13
2.2.2. <i>Graus de llibertat</i> .....	14
2.2.3. <i>Cinemàtica de la direcció</i> .....	14
2.2.4. <i>Mecanisme de la suspensió</i> .....	16
2.3. DINÀMICA DEL VEHICLE .....	17
2.3.1. <i>Dinàmica del vehicle en el pla</i> .....	17
2.3.2. <i>Dinàmica del vehicle sobre l'eix vertical</i> .....	18
2.3.3. <i>Concepte de sobreviratge i subviratge</i> .....	18
2.4. VIBRACIONS DEL VEHICLE .....	19
2.4.1. <i>Model de vibracions</i> .....	19
<b>3. MODELITZACIÓ</b> .....	<b>21</b>
3.1. INTERFÍCIE DE TREBALL .....	21
3.1.1. <i>Variables d'entrada</i> .....	21
3.1.2. <i>Representació de resultats</i> .....	25
3.2. MODELS DINÀMICS: EQUACIONS DE MOVIMENT .....	33
3.2.1. <i>Direcció i trajectòria</i> .....	33
3.2.2. <i>Acceleració i frenada</i> .....	34
3.2.3. <i>Deriva</i> .....	34
3.2.4. <i>Fricció</i> .....	34
<b>4. SIMULACIÓ</b> .....	<b>35</b>
4.1. CAS 1: ACCELERACIÓ RECTILÍNIA .....	35
4.2. CAS 2: ACCELERACIÓ I FRENADA RECTILÍNIA .....	42
4.3. CAS 3: CORBA A VELOCITAT CONSTANT .....	49
4.4. CAS 4: CORBA AMB ACCELERACIÓ .....	58
<b>5. PRESSUPOST</b> .....	<b>68</b>
<b>6. CONCLUSIONS</b> .....	<b>69</b>
<b>7. RELACIÓ DE DOCUMENTS</b> .....	<b>71</b>
<b>8. BIBLIOGRAFIA</b> .....	<b>72</b>
<b>ANNEXOS</b> .....	<b>73</b>
ANNEX A: INSTRUCCIONS D'ÚS PROGRAMA DE SIMULACIÓ .....	73

ANNEX B: CODIS INFORMÀTICS.....	74
<i>B.1: Codi interfície gràfica.....</i>	<i>74</i>
<i>B.2: Codi funció de simulació .....</i>	<i>101</i>
<i>B.3: Codi funció de les equacions de moviment.....</i>	<i>104</i>
<i>B.4: Codi funció d'acceleració i frenada.....</i>	<i>109</i>
<i>B.5: Codi funció de la direcció.....</i>	<i>111</i>
<i>B.6: Codi funció de lliscament.....</i>	<i>113</i>
<i>B.7: Codi funció de fricció.....</i>	<i>116</i>
<i>B.8: Funció Snapshots o representació gràfica de la trajectòria .....</i>	<i>118</i>
<i>B.9: Funció de representació gràfica de resultats.....</i>	<i>122</i>

## Índex de figures

Figura 1: Digrama de forces en el pla (Jazar, 2014) .....	7
Figura 2: Diagrama de forces en inclinació (Jazar, 2014) .....	8
Figura 3: Angles, forces i moments sobre els pneumàtics (Jazar, 2014).....	9
Figura 4: Concepte camber (Introducción a la dinámica en Formula Student, 2018) .....	10
Figura 5: Forces sobre el pneumàtic (Jazar, 2014).....	11
Figura 6: Elements de la transmissió (Jazar, 2014) .....	11
Figura 7: Conversions de la transmissió (Jazar, 2014).....	12
Figura 8: Diagrama potència parell respecte velocitat angular (Jazar, 2014) .....	12
Figura 9: Diagrama caixa de canvis (Jazar, 2014) .....	13
Figura 10: Veocitats de les masses principals del vehicle (Jazar, 2014) .....	14
Figura 11: Graus de llibertat del vehicle (Jazar, 2014) .....	14
Figura 12: Centre de rotació en una corba (Jazar, 2014).....	15
Figura 13: Mecanisme de la suspensió (Jazar, 2014).....	16
Figura 14: Dinàmica del vehicle en el pla (Jazar, 2014).....	17
Figura 15: Sobreviratge (Goodrich).....	19
Figura 16: Subviratge (Goodrich).....	19
Figura 17: Model de vibracions (Jazar, 2014) .....	20
Figura 18: Esquema de suspensió en el vehicle (Jazar, 2014) .....	20
Figura 19: Diagrama de funcions del programa .....	21
Figura 20: Interfície gràfica de variables d'entrada .....	23
Figura 21: Interfície gràfica amb dades correctes.....	24
Figura 22: Interfície gràfica amb dades incorrectes.....	25
Figura 23: Pantalla resultat de la simulació de la trajectoria.....	26
Figura 24: Pantalla de resultats gràfics .....	26
Figura 25: Gràfica de la velocitat.....	27
Figura 26: Gràfica de l'input de direcció .....	27
Figura 27: Gràfica del moviment del cdg en l'eix Z.....	28
Figura 28: Gràfica del balanceig o "pitch".....	28
Figura 29: Grafica del balamceig lateral o "roll" .....	29
Figura 30: Gràfica de l'acceleració lateral .....	29
Figura 31: Gràfiques de les velocitats de les rodes davanteres i posteriors .....	30
Figura 32: Gràfiques del lliscament dels pneumàtics davanteres posteriors .....	30
Figura 33: Gràfiques de les càrregues sobre les rodes davanteres i posteriors .....	31
Figura 34: Gràfica de l'acceleració angular en el moviment de "yaw".....	31
Figura 35: Gràfica del lliscament lateral .....	32
Figura 36: Gràfica del moviment de les rodes en l'eix Z.....	32
Figura 37: Entrada de variables de direcció i acceleració .....	33
Figura 38: Entrada de variables Cas 1.1 .....	36
Figura 39: Trajectoria Cas 1.1.....	36
Figura 40: Resultats gràfics Cas 1.1 .....	37
Figura 41: Entrada variables cas 1.2.....	38
Figura 42: Trajectoria cas 1.2 .....	38
Figura 43: Resultats gràfics cas 1.2.....	39
Figura 44: Velocitat cas 1 .....	39
Figura 45: Gràfiques z-motion cas 1 .....	40
Figura 46: Gràfiques velocitat rodes davanteres cas 1.....	40
Figura 47: Gràfiques velocitat rodes posteriors cas 1.....	41
Figura 48: Gràfiques lliscament rodes davanteres cas 1 .....	41
Figura 49: Gràfiques lliscament rodes posteriors cas 1 .....	41
Figura 50: Gràfiques càrregues rodes davanteres cas 1 .....	42
Figura 51: Gràfiques càrregues rodes posteriors cas 1 .....	42
Figura 52: Entrada de dades cas 2.1 .....	43

Figura 53: Trajectoria cas 2.1 .....	43
Figura 54: Resultats gràfics cas 2.1 .....	44
Figura 55: Entrada de dades cas 2.2 .....	45
Figura 56: Trajectoria cas 2.2 .....	45
Figura 57: Resultats gràfics cas 2.2 .....	46
Figura 58: Gràfica velocitat cas 2.....	46
Figura 59: Gràfiques z-motion cas 2 .....	47
Figura 60: Gràfiques velocitat rodes davanteres cas 2.....	47
Figura 61: Gràfiques velocitat rodes posteriors cas 2.....	48
Figura 62: Gràfiques lliscament rodes davanteres cas 2.....	48
Figura 63: Gràfiques lliscament rodes posteriors cas 2.....	48
Figura 64: Gràfiques càrregues rodes davanteres cas 2.....	49
Figura 65: Gràfiques càrregues rodes posteriors cas 2.....	49
Figura 66: Entrada variables cas 3.1.....	50
Figura 67: Trajectoria cas 3.1 .....	51
Figura 68: Resultats gràfics cas 3.1.....	51
Figura 69: Entrada variables cas 3.2.....	52
Figura 70: Trajectoria cas 3.2 .....	53
Figura 71: Resultats gràfics cas 3.2.....	53
Figura 72: Gràfiques velocitat cas 3.....	54
Figura 73: Gràfiques z-motion cas 3.....	54
Figura 74: Gràfiques acceleració lateral cas 3.....	55
Figura 75: Gràfiques velocitat rodes davanteres cas 3.....	55
Figura 76: Gràfiques velocitat rodes posteriors cas 3.....	56
Figura 77: Gràfiques lliscament rodes davanteres cas 3.....	56
Figura 78: Gràfiques lliscament rodes posteriors cas 3.....	56
Figura 79: Gràfiques càrregues rodes davanteres cas 3.....	57
Figura 80: Gràfiques càrregues rodes posteriors cas 3.....	57
Figura 81: Gràfiques lliscament lateral cas 3.....	58
Figura 82: Gràfiques z-motion eix davanter-posterior cas 3.....	58
Figura 83: Entrada variables cas 4.1.....	59
Figura 84: Trajectòria cas 4.1 .....	60
Figura 85: Resultats gràfics cas 4.1.....	60
Figura 86: entrada variables cas 4.2.....	61
Figura 87: Trajectòria cas 4.2 .....	62
Figura 88: Resultats gràfics cas 4.2.....	62
Figura 89: Gràfiques velocitat cas 4.....	63
Figura 90: Gràfiques velocitat rodes davanteres cas 4.....	63
Figura 91: Gràfiques velocitat rodes posteriors cas 4.....	64
Figura 92: Gràfiques lliscament rodes davanteres cas 4.....	64
Figura 93: Gràfiques lliscament rodes posteriors cas 4.....	65
Figura 94: Gràfiques càrregues rodes davanteres cas 4.....	65
Figura 95: Gràfiques càrregues rodes posteriors cas 4.....	66
Figura 96: Gràfiques lliscament lateral cas 4.....	66
Figura 97: Gràfiques z-motion eix davanter-posterior cas 4.....	67

# 1. INTRODUCCIÓ

## 1.1. Antecedents

A les pràctiques d'Enginyeria del Transport del 1r curs del Màster en Enginyeria Industrial es simula el comportament en el pla de vehicles, actualment, mitjançant un conjunt de rutines en MATLAB de molt difícil interacció amb l'estudiant, que dificulta la comprensió dels conceptes que es volen analitzar.

## 1.2. Objecte

L'objecte d'aquest treball és el de modificar l'actual programa de simulació del comportament dinàmic de vehicles per oferir una eina de treball més còmode i intuïtiva per als futurs estudiants de l'assignatura.

## 1.3. Abast

Es pretén presentar un nou programa que ha de permetre:

- Introduir les dades bàsiques del vehicle de manera senzilla: massa, posició del centre de gravetat, radi dels pneumàtics, parell motor, etc.
- Descriure les accions del conductor pel que fa al gir del volant i acceleració o frenada.

A partir d'aquesta informació ha de resoldre les equacions de la dinàmica i proporcionar:

- Informació per analitzar el sobreviratge i el subviratge.
- Informació sobre les càrregues als pneumàtics.
- Proporcionar els resultats de manera gràfica i comprensible.

## 2. FONAMENT TEÒRIC

### 2.1. Moviment del vehicle

#### 2.1.1. Dinàmica del moviment rectilini

Previ al càlcul de les forces resultants en el moviment rectilini d'un vehicle es veurà el càlcul en estàtic d'aquest tipus de sistemes. Per al moviment rectilini d'un vehicle simètric i rígid, assumim que les forces a les rodes de l'esquerra són iguals a les forces a les rodes de la dreta, per d'aquesta manera simplificar el càlcul.

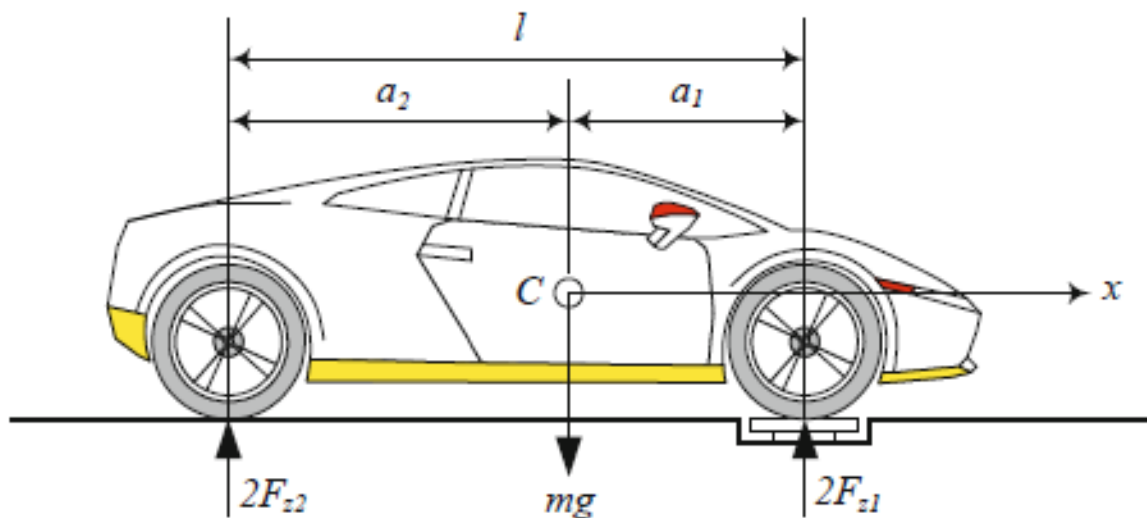


Figura 1: Digrama de forces en el pla (Jazar, 2014)

D'aquesta manera obtenim que els valors de les forces estàtiques de contacte d'un vehicle són les següents:

$$F_{z1} = \frac{1}{2} mg \frac{a_2}{l} \quad (\text{Eq. 1})$$

$$F_{z2} = \frac{1}{2} mg \frac{a_1}{l} \quad (\text{Eq. 2})$$

Quan un cotxe accelera en un terreny amb un angle d'inclinació  $\theta$ , les forces normals sota les rodes davantera i posterior corresponen a  $F_{z1}$  i  $F_{z2}$ :

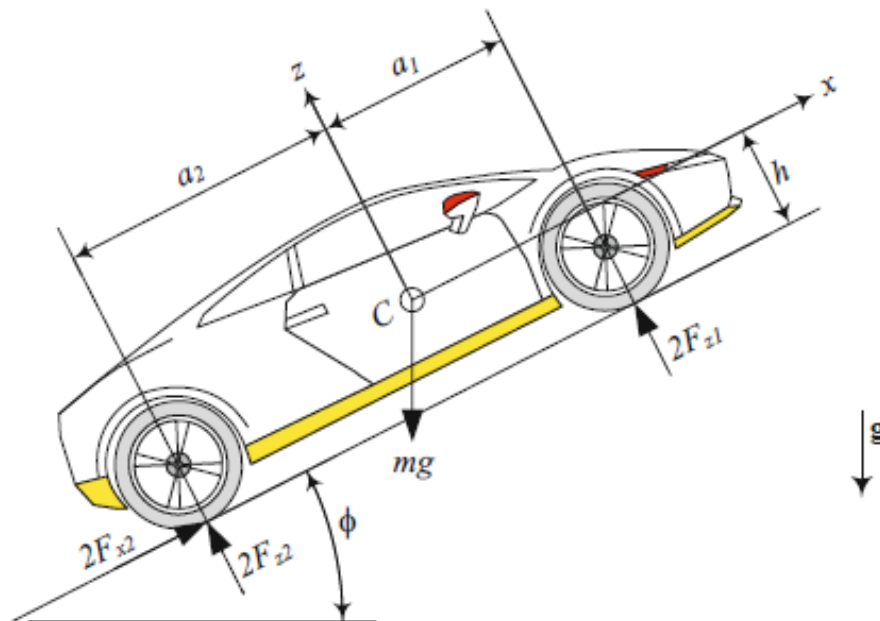


Figura 2: Diagrama de forces en inclinació (Jazar, 2014)

$$F_{z1} = \frac{1}{2} mg \left( \frac{a_2}{l} \cos \theta - \frac{h}{l} \sin \theta \right) - \frac{1}{2} ma \frac{h}{l} \quad (\text{Eq. 3})$$

$$F_{z2} = \frac{1}{2} mg \left( \frac{a_1}{l} \cos \theta + \frac{h}{l} \sin \theta \right) + \frac{1}{2} ma \frac{h}{l} \quad (\text{Eq. 4})$$

On el primer terme és la força estàtica, la qual depèn de la pendent i de la posició del centre de masses. El segon terme, el qual està en positiu per a la roda posterior i negatiu per a la davantera, és el corresponent a la força dinàmica ja que és dependent de l'acceleració.

### 2.1.2. Dinàmica dels pneumàtics

Per descriure la interacció entre els pneumàtics i el terra i el seu corresponent sistema de forces assumim una superfície plana i un centre de coordenades cartesianes al centre de l'empremta del pneumàtic, tal i com podem veure en la figura següent.



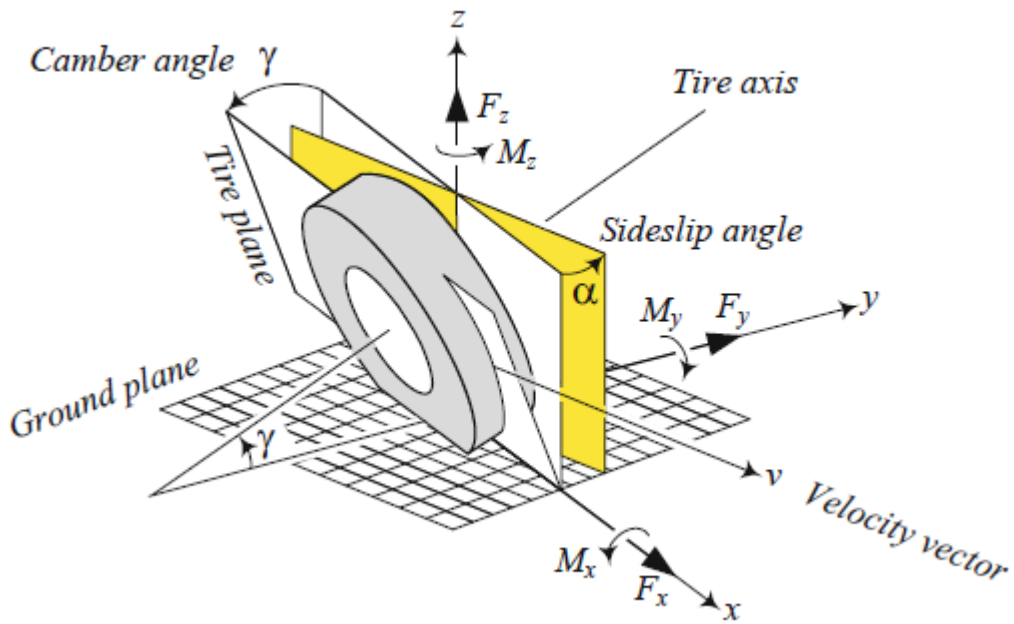


Figura 3: Angles, forces i moments sobre els pneumàtics (Jazar, 2014)

On l'eix x segueix la direcció de la línia d'intersecció del pneumàtic amb el terra, l'eix z es perpendicular a la superfície o terra, i l'eix y segueix la direcció i sentit que fa complir amb la regla de la mà dreta.

Una roda en moviment sobre una superfície plana i carregada verticalment té un radi efectiu, anomenat radi de gir que compleix amb:

$$R_w = \frac{v_x}{w_w} \quad (\text{Eq. 5})$$

On  $v_x$  és la velocitat en l'eix de desplaçament i  $w_w$  és la velocitat angular de la roda. A més el radi efectiu és aproximadament igual a:

$$R_w \approx R_g - \frac{R_g - R_h}{3} \quad (\text{Eq. 6})$$

I és un valor entre el radi geomètric o radi sense càrrega ( $R_g$ ) i el radi amb càrrega ( $R_h$ )

$$R_h < R_w < R_g \quad (\text{Eq. 7})$$

La força del pneumàtic en la direcció de l'eix x és la combinació de la força longitudinal  $F_x$  i la resistència al gir:

$$F_x = \mu_x(s)F_z \quad (\text{Eq. 8})$$

On  $s$  és la relació de lliscament lateral del pneumàtic

$$s = \frac{R_g w_w}{v_x} \quad (\text{Eq. 9})$$

$$\mu_x(s) = C_s s \quad \text{on} \quad s \ll 1 \quad (\text{Eq. 10})$$

La força del pneumàtic en la direcció de l'eix  $y$  és la combinació de les forces d'empenta degudes al viratge i al *camber*.

$$F_y = -C_\gamma \gamma - C_\alpha \alpha \quad (\text{Eq. 11})$$

El concepte de *camber* és veu reflectit de manera clara en la figura que tenim a continuació:

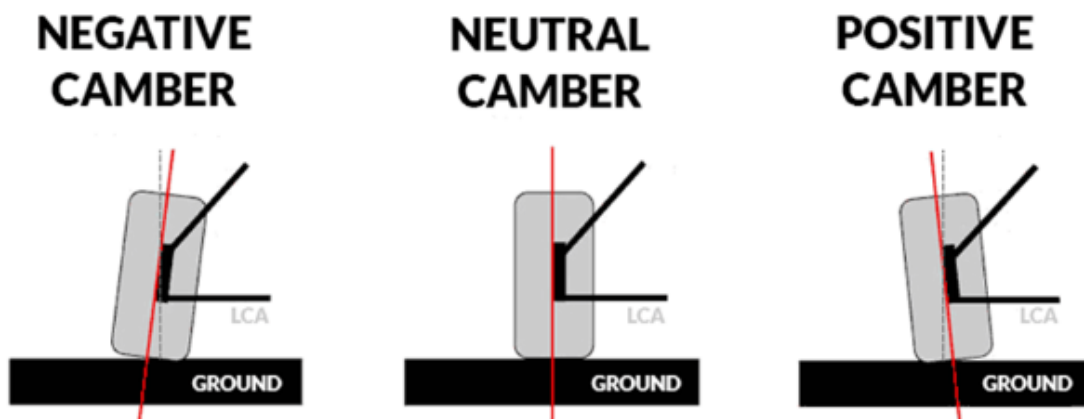


Figura 4: Concepte *camber* (Introducció a la dinàmica en Formula Student, 2018)

Una roda en moviment sobre el terra també genera una força longitudinal coneguda com a resistència al rodament. Aquesta força segueix la direcció de moviment de la roda però s'aplica en sentit oposat i és proporcional a la força normal en el pneumàtic:

$$F_r = \mu_r F_z \quad (\text{Eq. 12})$$

On el paràmetre  $\mu_r$  és l'anomenat coeficient de fricció a rodament i és funció de les propietats mecàniques del pneumàtic, la velocitat, la temperatura, la càrrega, la carretera, etc.

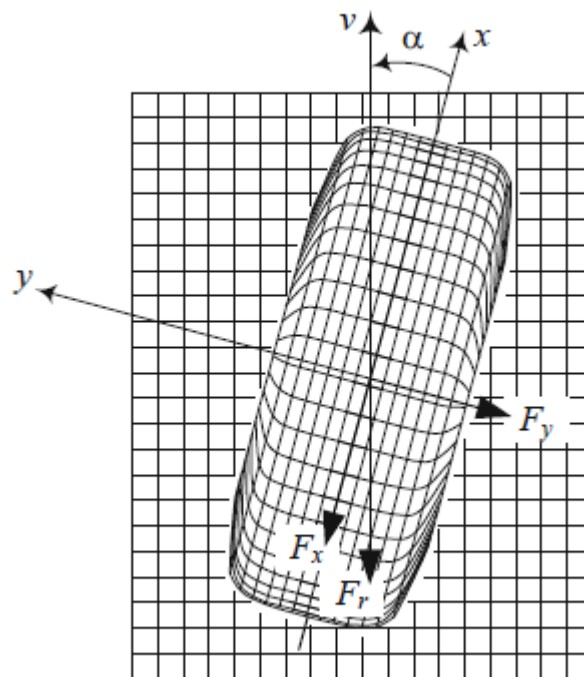


Figura 5: Forces sobre el pneumàtic (Jazar, 2014)

### 2.1.3. Dinàmica de la transmissió

El sistema de transmissió és l'encarregat de transmetre el parell i la potència generats pel motor fins a les rodes de tracció. Per aconseguir-ho, els principals elements necessaris presents a un vehicle són el motor, l'embragatge, la caixa de canvis i l'eix de transmissió.

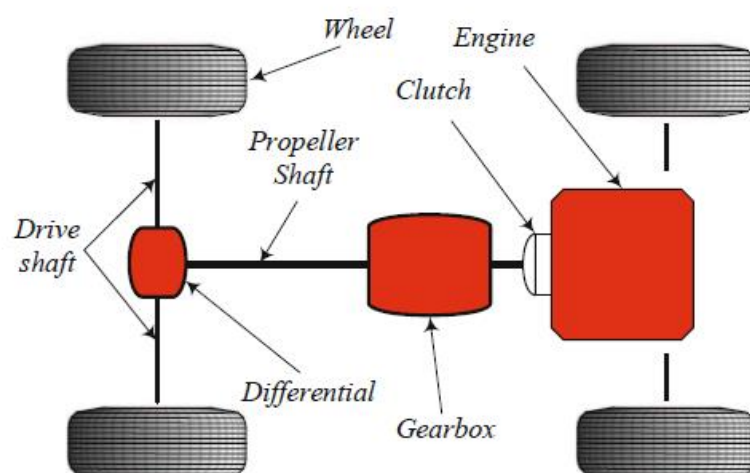


Figura 6: Elements de la transmissió (Jazar, 2014)

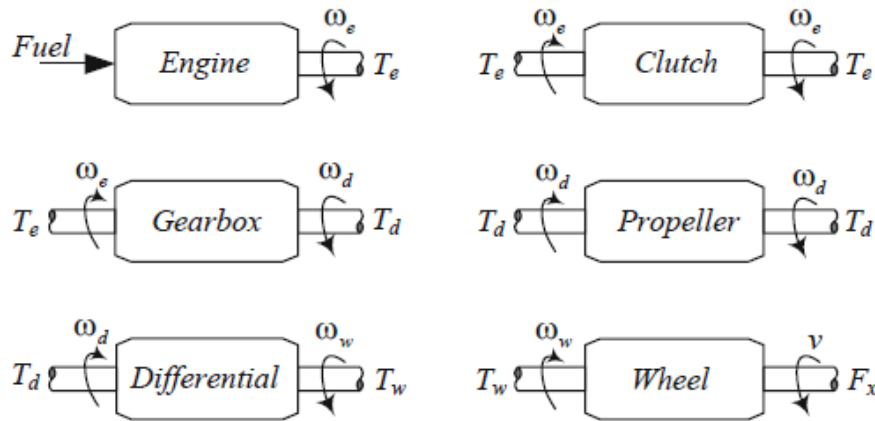


Figura 7: Conversions de la transmissió (Jazar, 2014)

La potència màxima  $P_e$  a la que pot arribar un motor de combustió interna està directament relacionada amb la velocitat angular del motor  $w_e$ . Aquesta relació s'ha de determinar experimentalment per a cada motor, tot i així, el rendiment de potència pot arribar a expressar-se amb una funció matemàtica:

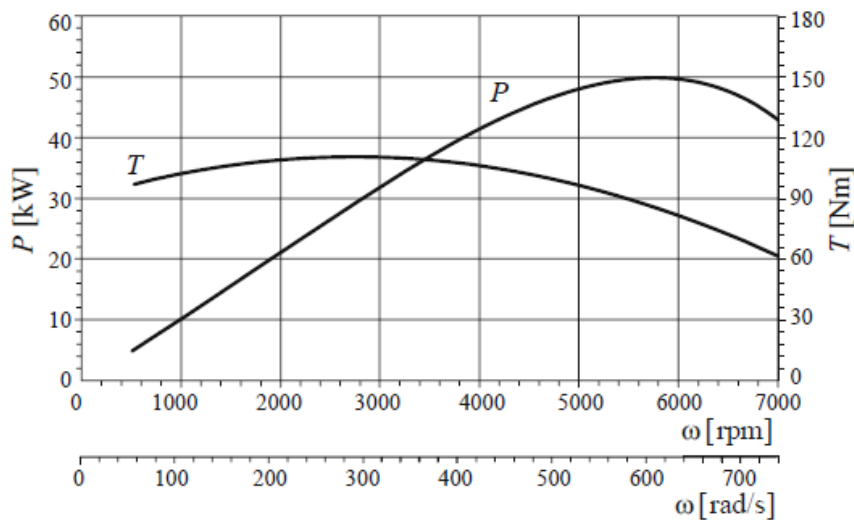


Figura 8: Diagrama potència parell respecte velocitat angular (Jazar, 2014)

$$P_e = P_1 w_e + P_2 w_e^2 + P_3 w_e^3 \tag{Eq. 13}$$

On els valors de  $P_1$ ,  $P_2$  i  $P_3$  equivalen a:

$$P_1 = \frac{P_M}{w_M} \tag{Eq. 14}$$

$$P_2 = \frac{P_M}{w_M^2} \tag{Eq. 15}$$

$$P_3 = \frac{P_M}{w_M^3} \tag{Eq. 16}$$

On  $w_M$  és la velocitat angular en radians per segon, a la qual el motor arriba al seu màxim de potència  $P_M$  (en Nm/s). El parell del motor queda definit per l'expressió següent:

$$T_e = \frac{P_e}{w_e} = P_1 + P_2 w_e + P_3 w_e^2 \quad (\text{Eq. 17})$$

Un motor ideal és aquell que produeix una potència constant independentment de la velocitat, tal que:

$$P_e = P_0 \quad (\text{Eq. 18}) \quad T_e = \frac{P_0}{w_e} \quad (\text{Eq. 19})$$

Per tal de fer treballar el motor a una potència el més constant possible i pròxima a la potència màxima  $P_M$  s'utilitzen les caixes de canvis. Per tal de dissenyar una caixa de canvis adient al motor es fan servir les equacions de velocitat i les equacions de tracció:

$$w_e = \frac{n_i n_d}{R_w} v_x \quad (\text{Eq. 20})$$

$$T_e = \frac{1}{\eta} \frac{R_w}{n_i n_d} F_x \quad (\text{Eq. 21})$$

Aquestes equacions afirmen que la velocitat d'avanç  $v_x$  d'un vehicle és proporcional a la velocitat angular del motor  $w_e$  i la força de tracció del pneumàtic  $F_x$  és proporcional al parell motor  $T_e$ , on  $R_w$  és el radi eficaç del pneumàtic,  $n_d$  és la relació de transmissió diferencial,  $n_i$  és la transmissió de la caixa de canvis. La relació entre el número d'engranatge  $i$  i  $\eta$  és la eficiència general de la transmissió.

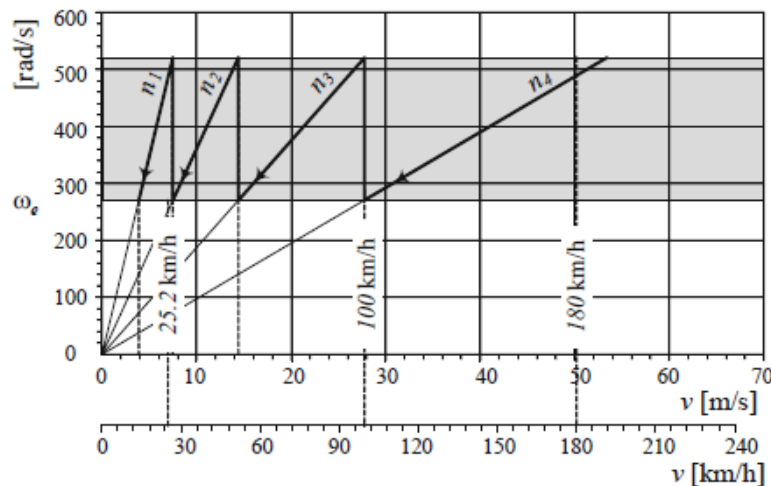


Figura 9: Diagrama caixa de canvis (Jazar, 2014)

## 2.2. Cinemàtica del vehicle

### 2.2.1. Masses Principals

Considerem un sistema de 5 masses principals per als vehicles de 4 rodes, sobre cada una de les quals tindrem els seus corresponents graus de llibertat. Aquestes 5 masses són les

formades per les 4 rodes i per la massa conjunta del xassís, motor i tot el conjunt d'elements del cos de vehicle.

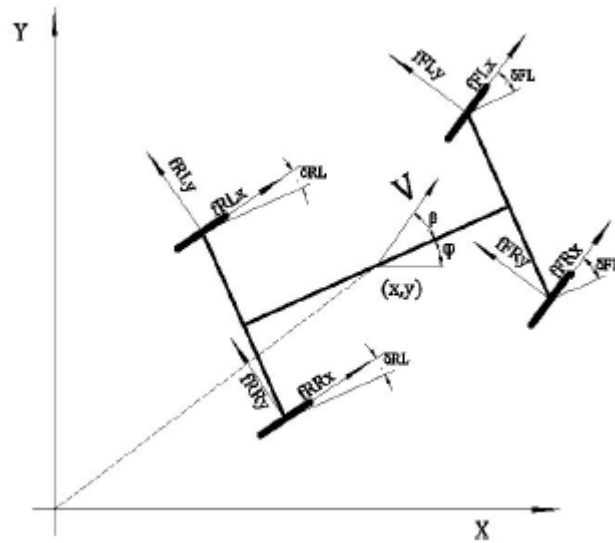


Figura 10: Veocitats de les masses principals del vehicle (Jazar, 2014)

### 2.2.2. Graus de llibertat

Considerem 14 graus de llibertat per als vehicles de 4 rodes, tenint desplaçament i gir sobre els 3 eixos principals del cos, i desplaçament vertical i angular a cada una de les rodes, de manera que tenim graus de llibertat en  $X, Y, Z, \varphi, \theta, \Psi, w_{i,j}, Z_{i,j}$

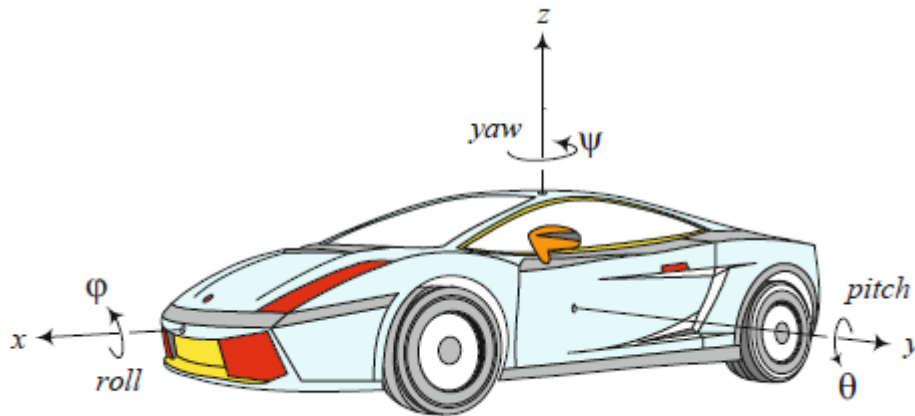


Figura 11: Graus de llibertat del vehicle (Jazar, 2014)

### 2.2.3. Cinemàtica de la direcció

La direcció és l'element necessari per orientar un vehicle cap a la direcció desitjada i que compleixi amb la trajectòria que esperàvem.

Quan un vehicle es troba en una corba, les rodes més properes al centre de rotació són les que anomenem rodes interiors, i les que es troben més allunyades, són les anomenades

rodes exteriors. Tot i així quan la velocitat d'un vehicle és molt lenta hi ha una condició cinemàtica entre les rodes interiors i exteriors anomenada condició d'Ackerman.

La gran majoria d'automòbils, que considerem que són vehicles de 4 rodes, solen tenir la direcció sobre les rodes davanteres, mantenint les posteriors fixes.

La relació cinemàtica entre les rodes interiors i exteriors ve donada per l'expressió:

$$\cot \delta_o - \cot \delta_i = \frac{w}{l} \quad (\text{Eq. 22})$$

On  $\delta_i$  fa referència a l'angle de direcció de la roda interior i  $\delta_o$  fa referència a l'angle de direcció de la roda exterior,  $w$  és el *track* o distància entra rodes interiors i exteriors i  $l$  és la distància entre rodes davanteres i rodes posteriors.

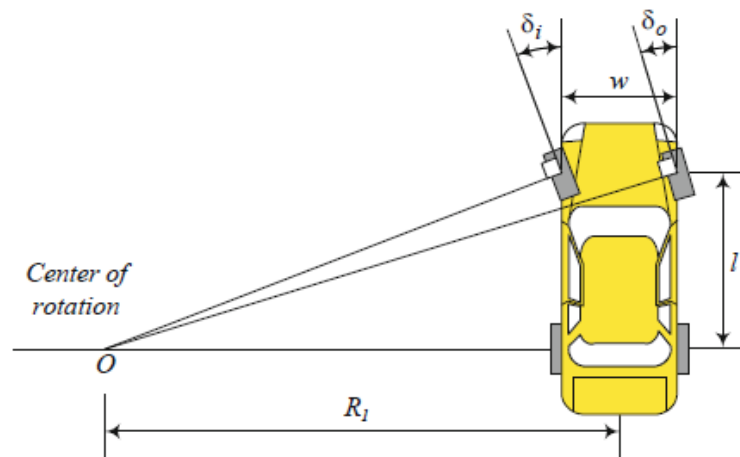


Figura 12: Centre de rotació en una corba (Jazar, 2014)

El centre de masses d'un vehicle amb que porta una direcció no rectilínia, dibuixarà un cercle de radi  $R$  tal que:

$$R = \sqrt{a_2^2 + l^2 \cot^2 \delta} \quad (\text{Eq. 23})$$

$$\cot \delta = \frac{\cot \delta_o + \cot \delta_i}{2} \quad (\text{Eq. 24})$$

L'angle  $\delta$  és l'equivalent a l'angle de direcció d'un vehicle de dues rodes amb una mateixa distància entre roda davantera i posterior i un mateix angle de rotació  $R$ .

## 2.2.4. Mecanisme de la suspensió

La suspensió és el mecanisme que uneix les rodes al xassís del vehicle permetent un moviment relatiu. Podem distingir entre dos tipus generals de suspensions, les dependents, on la roda esquerra i dreta del mateix eix estan connectades rígidament, i les independents, on la roda esquerra i la roda dreta estan desconnectades.

L'eix d'alabeig és la línia instantània sobre la qual roda el cos d'un vehicle. L'eix d'alabeig es troba connectant el centre del rotació de les suspensions frontals i posteriors del vehicle. El centre de gir instantani d'una roda respecte al cos s'anomena centre d'alabeig de la suspensió. Per trobar el centre d'alabeig de la meitat frontal o posterior del cotxe, hauríem de determinar els centres d'alabeig de la suspensió i trobar la intersecció de les línies que connecten els centres d'alabeig de la suspensió amb els centres de les seves empremtes associades.

S'utilitzen tres trames de coordenades per expressar l'orientació d'un pneumàtic i de la roda respecte al cos del vehicle: el sistema de la roda  $W$ , el sistema del cos  $C$  i el sistema del pneumàtic  $T$ . El sistema de coordenades de roda  $W(x_w, y_w, z_w)$  està unit al centre d'una roda. Transmet tota tracció i rotació de la roda, excepte el gir. Per tant, els eixos  $x_w$  i  $z_w$  sempre es troben en el pla dels pneumàtics, mentre que l'eix  $y_w$  es troba sempre al llarg de l'eix de gir.

Quan la roda es troba en posició recte i el sistema  $W$  és paral·lel al sistema de coordenades del vehicle, col·loquem el sistema de coordenades  $C(x_c, y_c, z_c)$  al centre de la roda, paral·lel al centre de coordenades del vehicle.

El sistema de coordenades  $T(x_t, y_t, z_t)$  està col·locat al centre de l'empremta del pneumàtic. L'eix  $z_t$  sempre és perpendicular al terra i l'eix  $x_t$  segueix la línia formada per la intersecció del pla del pneumàtic i el terra.

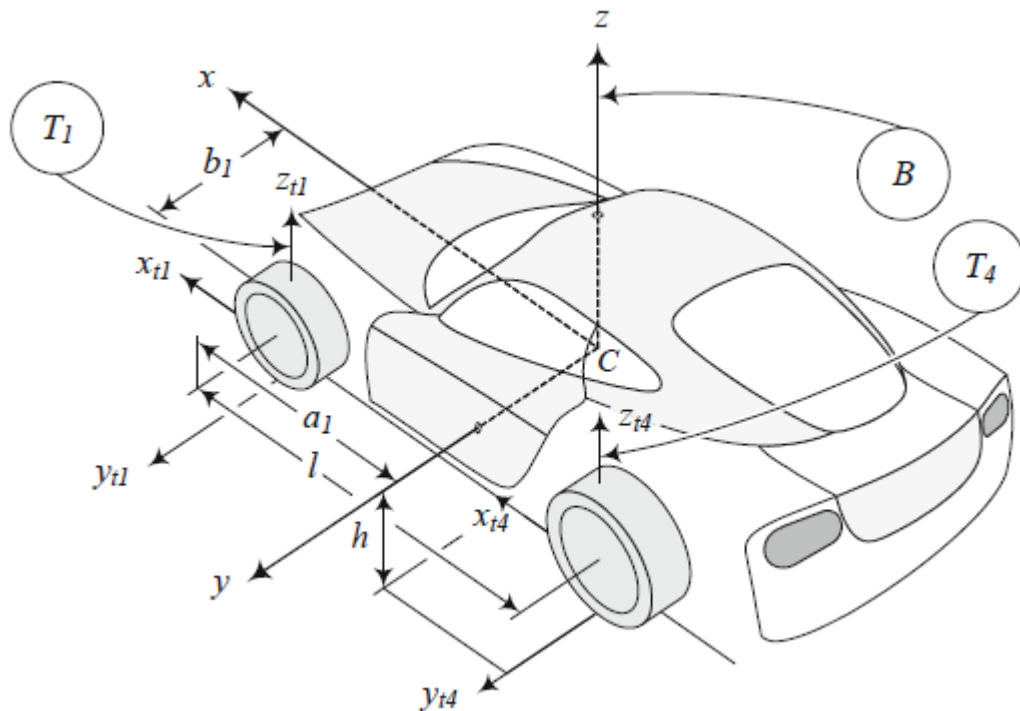


Figura 13: Mecanisme de la suspensió (Jazar, 2014)



## 2.3. Dinàmica del vehicle

### 2.3.1. Dinàmica del vehicle en el pla

El model dinàmic per a un vehicle en el pla és un model aplicable sempre que tinguem un moviment endavant, lateral o de rotació respecte l'eix z del cos del vehicle, o *yaw*, i les velocitats d'aquests siguin importants i suficients per examinar el comportament del vehicle. En aquest tipus de model s'ignora el moviment en l'eix x del vehicle o *roll*.

Les equacions de moviment de Newton-Euler per a un vehicle rigid amb centre de coordenades el propi cos del vehicle, ens relacionen les forces aplicades sobre aquest.

$$F_x = m v'_x - m w_z v_y \tag{Eq. 25}$$

$$F_y = m v'_y + m w_z v_x \tag{Eq. 26}$$

$$M_z = w'_z I_z \tag{Eq. 27}$$

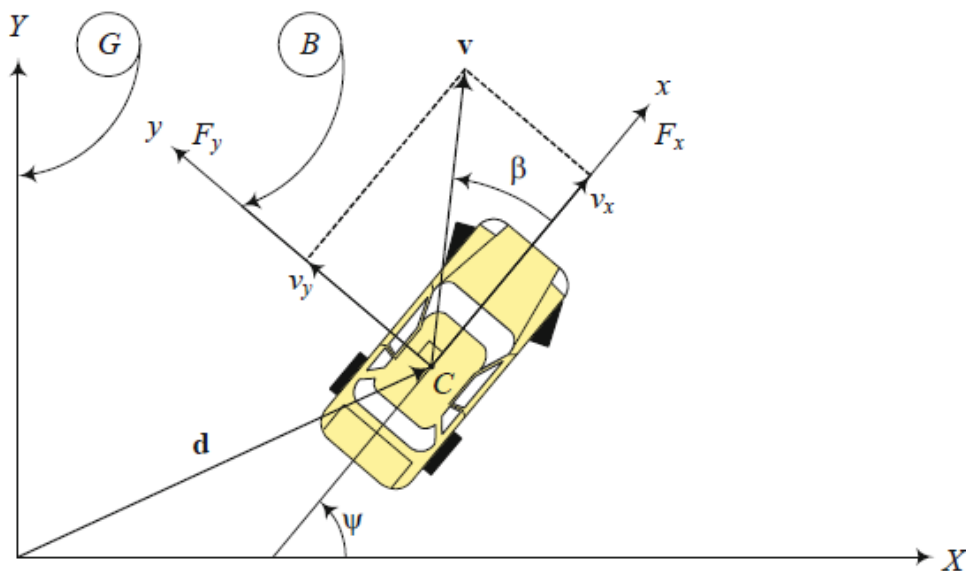


Figura 14: Dinàmica del vehicle en el pla (Jazar, 2014)

En vehicles amb tracció davantera, les equacions dinàmiques del vehicle són expressades mitjançant les tres equacions diferencials ordinàries de primer grau següents:

$$v'_x = \frac{F_x}{m} + r v_y \tag{Eq. 28}$$

$$v'_y = \frac{1}{m v_x} (-a_1 C_{af} + a_2 C_{ar}) r - \frac{1}{m v_x} (C_{af} + C_{ar}) v_y + \frac{1}{m} C_{af} \delta - r v_x \tag{Eq. 29}$$

$$r' = \frac{1}{I_z v_x} (-a_1^2 C_{af} - a_2^2 C_{ar}) r - \frac{1}{I_z v_x} (a_1 C_{af} - a_2 C_{ar}) v_y + \frac{1}{I_z} a_1 C_{af} \delta \tag{Eq. 30}$$

On la segona i tercera equacions s'han d'escriure en format de matrius de manera que el sistema quedi:

$$[v_y \ r]^T \quad \text{o bé} \quad [\beta \ r]^T \quad (\text{Eq. 31})$$

### 2.3.2. Dinàmica del vehicle sobre l'eix vertical

El model de bicicle que millor defineix la dinàmica del vehicle sobre l'eix vertical és aquell en el qual els graus de llibertat corresponents al *roll* i *yaw* així com els moviments sobre els eixos x i z estan contemplats. Aquest model es pot expressar mitjançant un sistema de cinc equacions diferencials:

$$v'_x = \frac{1}{m} F_x + r v_y \quad (\text{Eq. 32})$$

$$\begin{bmatrix} \dot{v}_y \\ \dot{p} \\ \dot{\varphi} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{C_\beta}{mv_x} & \frac{C_p}{m} & \frac{C_\varphi}{m} & \frac{C_r - v_x}{m} \\ \frac{E_\beta}{I_x v_x} & \frac{E_p}{I_x} & \frac{E_\varphi}{I_x} & \frac{E_r}{I_x} \\ 0 & 1 & 0 & 0 \\ \frac{D_\beta}{I_z v_x} & \frac{D_p}{I_z} & \frac{D_\varphi}{I_z} & \frac{D_r}{I_z} \end{bmatrix} \begin{bmatrix} v_y \\ p \\ \varphi \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_\delta}{m} \\ \frac{E_\delta}{I_x} \\ 0 \\ \frac{D_\delta}{I_z} \end{bmatrix} \delta \quad (\text{Eq. 33})$$

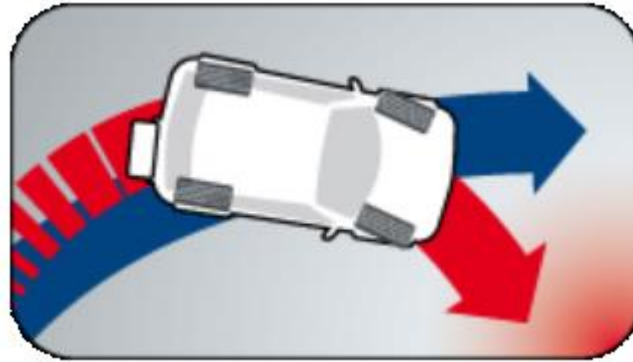
On els coeficients C E i D corresponent als coeficients de forces laterals, del moment en el gir de "guinyada", i del moment en el gir "d'alabeig" respectivament, i on els subíndex  $\beta, p, \varphi$  i  $r$  corresponen en cada cas a l'angle de lliscament, a la velocitat de balanceig, a l'angle de "guinyada" i a la velocitat de "guinyada".

D'aquesta manera el paràmetres  $C_\beta$  correspondria al coeficient de força lateral de l'angle de lliscament del vehicle,  $C_p$  correspondria al coeficient de força lateral de la velocitat "d'alabeig" i així successivament.

El vehicle rep com a variable d'entrada un angle de direcció  $\delta$  a partir del qual tenim cinc variables de sortida, que són  $v_x, v_y, p, \varphi, r$ . Tot i així, si mantenim les velocitat endavant com a constant ( $v_x$ ) i utilitzant-la com a paràmetre podem separar la primera equació de les altres. Aquesta és una pràctica bastant habitual en molts casos de simulació del comportament de vehicles.

### 2.3.3. Concepte de sobreviratge i subviratge

Aquests són dos conceptes molt comuns en l'estudi del comportament dinàmic de vehicles en trajectòries corbes. Per un costat tenim el sobreviratge, el qual és una deriva de l'eix posterior superior a la de l'eix davanter. A arrel d'aquest fenomen el cotxe sembla que vulgui girar més del que és esperat. Tots dos fenòmens es donen principalment al prendre trajectòries amb un gir massa pronunciat per a la velocitat amb la que s'hi entra, o bé en acceleracions durant la corba, però en el cas del sobreviratge és molt més comú d'experimentar en vehicles amb tracció en l'eix posterior.



*Figura 15: Sobreviratge (Goodrich)*

Per altra banda, el subviratge el podem definir com una deriva de l'eix davanter superior a l'experimentada en l'eix posterior de manera que la sensació és que el vehicle segueix en línia recta tot i haver girat el volant per prendre una corba. Al contrari que en el sobreviratge, aquest cas és dóna molt més freqüentment en vehicles amb tracció davantera.



*Figura 16: Subviratge (Goodrich)*

## 2.4. Vibracions del vehicle

### 2.4.1. Model de vibracions

Les vibracions mecàniques són el resultat de la continua transformació d'energia cinètica en energia potencial tant en un sentit com en l'altre. Quan l'energia potencial és al seu màxim l'energia cinètica és zero i viceversa.

Un sistema en vibració es pot assimilar al sistema format per tres elements, com són una massa, una molla i un dissipador o amortidor:

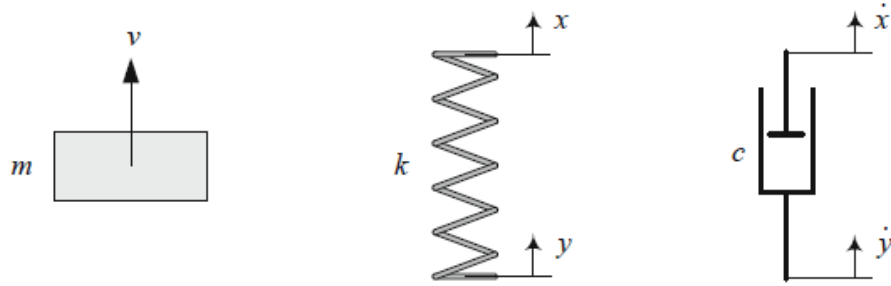


Figura 17: Model de vibracions (Jazar, 2014)

En el cas dels vehicles, aquests tenen sistemes amb múltiples graus de llibertat, i les vibracions que pateixen són bastant dependents de les freqüències naturals del mateix i de les seves formes.

Aquestes característiques es poden determinar resolent un problema de valors i vectors propis. Els models de vehicles vibratoris més pràctics, començant des del més senzill fins el més complex, són els models d'un vuitè d'automòbil, un quart d'automòbil, un bicicle, mig automòbil i un automòbil complet. Tenir matrius de massa simètrica, rigidesa i amortiment de sistemes amb múltiples graus de llibertat simplifica el càlcul dels problemes de valors i vectors propis. Per tenir matrius coeficients simètriques, definim l'energia cinètica, l'energia potencial i la funció de dissipació del sistema per quadratures i derivem les equacions de moviment aplicant el mètode de Lagrange.

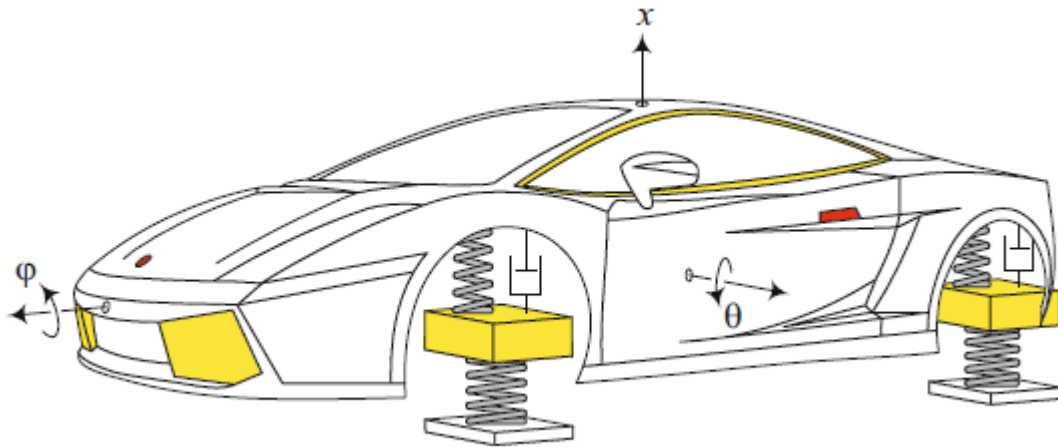


Figura 18: Esquema de suspensió en el vehicle (Jazar, 2014)

### 3. MODELITZACIÓ

#### 3.1. Interfície de treball

A partir de les modificacions que s'han fet al programa original de simulació dinàmica de vehicles, l'usuari haurà de treballar utilitzant dos del fitxers de *Matlab*. Un és l'encarregat de fer la funció d'interfícies gràfica per a una còmode manipulació de les variables d'entrada, mentre que l'altre, serà la funció principal que cridarà a executar les diferents funcions secundàries utilitzant les dades aportades com a variables d'entrada, i finalment presentant els resultats de forma gràfica.

L'esquema que es presenta a continuació mostra els diferents arxius ja existents del model, sobre els quals s'han fet les modificacions pertinents per tal de que tot el conjunt de funcions treballi amb els paràmetres entrats a partir de l'interfície gràfica creada.

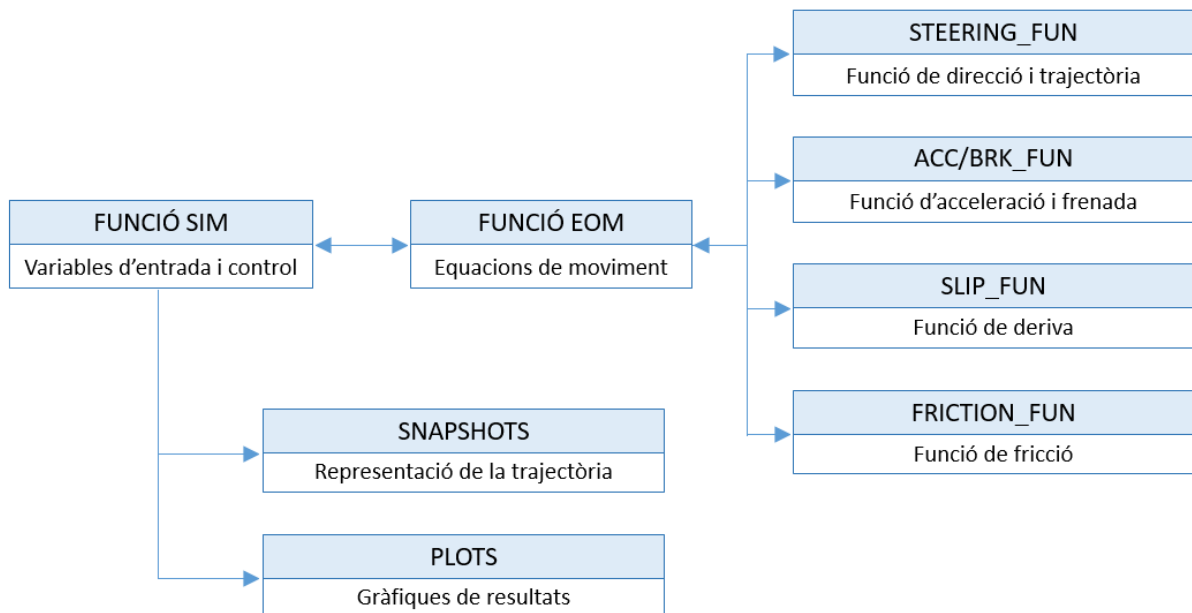


Figura 19: Diagrama de funcions del programa

##### 3.1.1. Variables d'entrada

Les variables d'entrada s'introdueixen manualment dins de la pestanya que s'obre al executar l'arxiu *interficie\_gràfica\_vehicles*. Dins d'aquest panell de control ens apareix una configuració de dades per defecte que podem modificar per adaptar les variables d'entrada al cas d'estudi desitjat.

Dins d'aquest panell podem definir dos grans blocs, el primer on s'introdueixen les variables d'entrada de tipus més tècnic, que són bàsicament característiques mecàniques del vehicle i el segon on tenim les possibles accions del conductor, pel que fa al control de l'acceleració i la frenada així com la direcció.

Dins d'aquest primer bloc tenim un apartat referit a la geometria del vehicle, on podem introduir la massa total del vehicle en quilograms, la distància des de l'eix davanter fins al

centre de gravetat del vehicle, en metres, així com la distància des de l'eix posterior a aquest centre, i el radi efectiu dels pneumàtics, també en metres.

Tenim un segon apartat on podem introduir els paràmetres del motor, en concret el parell d'aquest, i en el que hem d'introduir el valor en el camp que correspongui segons si la tracció és davantera, posterior o quatre per quatre, deixant l'eix no motriu en zero.

Seguidament podem introduir modificacions en els paràmetres que ens permeten calcular l'acció de les suspensions, tant en l'eix davanter com en el posterior. Els valors a modificar com a variables d'entrada per al càlcul de l'acció de la suspensió són els paràmetres  $k$  referent al coeficient de la molla i el paràmetre  $c$  corresponent al coeficient dissipador.

Finalment per tancar el primer bloc podem introduir la velocitat que portarà el vehicle a l'inici de la simulació en les unitats de quilometres per hora.

Pel que fa al segon bloc aquest està format per una matriu on podem introduir les accions del conductor entre deu intervals de temps. Per una banda tenim l'acceleració o frenada, en la que un valor de -1 equival a prémer el pedal d'acceleració al 100%, un valor de zero significa ni accelerar ni frenar i un valor de 1 es refereix a prémer el pedal del fre al 100%. Entre aquests dos intervals qualsevol combinació de percentatges és vàlida per introduir. Per l'altre banda amb un sistema similar tenim el paràmetre de la direcció. Un -1 equival a un gir a de les rodes davanteres cap a l'esquerra de 30 graus mentre que un 1 positiu equival a un gir de la mateixa magnitud cap a la dreta.

Després d'aquesta matriu de variables tenim una petita taula informativa com a recordatori dels convenis de signes i valors dels paràmetres que acabem de comentar. Finalment caldrà prémer el botó *EXECUTAR SIMULACIÓ* per tal de confirmar les dades entrades i que aquestes es quedin en un arxiu de text extern per després poder ser llegides per les diferents funcions de càlcul.

**GEOMETRIA DEL VEHICLE**

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

**PARÀMETRES DEL MOTOR**

Parell davanter:  Nm

Parell posterior:  Nm

**PARÀMETRES DE LA SUSPENSIO**

Kf:  Cf:

Kr:  Cr:

**VELOCITAT INICIAL DEL VEHICLE**

Velocitat inicial:  km/h

**ACCIONS DEL CONDUCTOR**

Temps	Acceleració	Direcció
0	-0.5	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	0	0.3
6	0	0.3
7	0	0.3
8	0	0
9	0	0.5
10	0	0.5

Informació

Acceleració:  Accel.  Frenada

Direcció:  Esquerra  Dreta

**EXECUTAR SIMULACIÓ**

Figura 20: Interficie gràfica de variables d'entrada

Per assegurar-nos que les dades s'han entrat correctament, sobretot pel que fa a la matriu de les accions del conductor, ens apareixerà un missatge que ens confirmarà que les dades són correctes o ens advertirà que alguna no ho és i que s'han de revisar i rectificar abans d'executar la funció de càlcul. Aquesta funció s'assegura de que els valors entrats estiguin dintre dels límits especificats anteriorment.

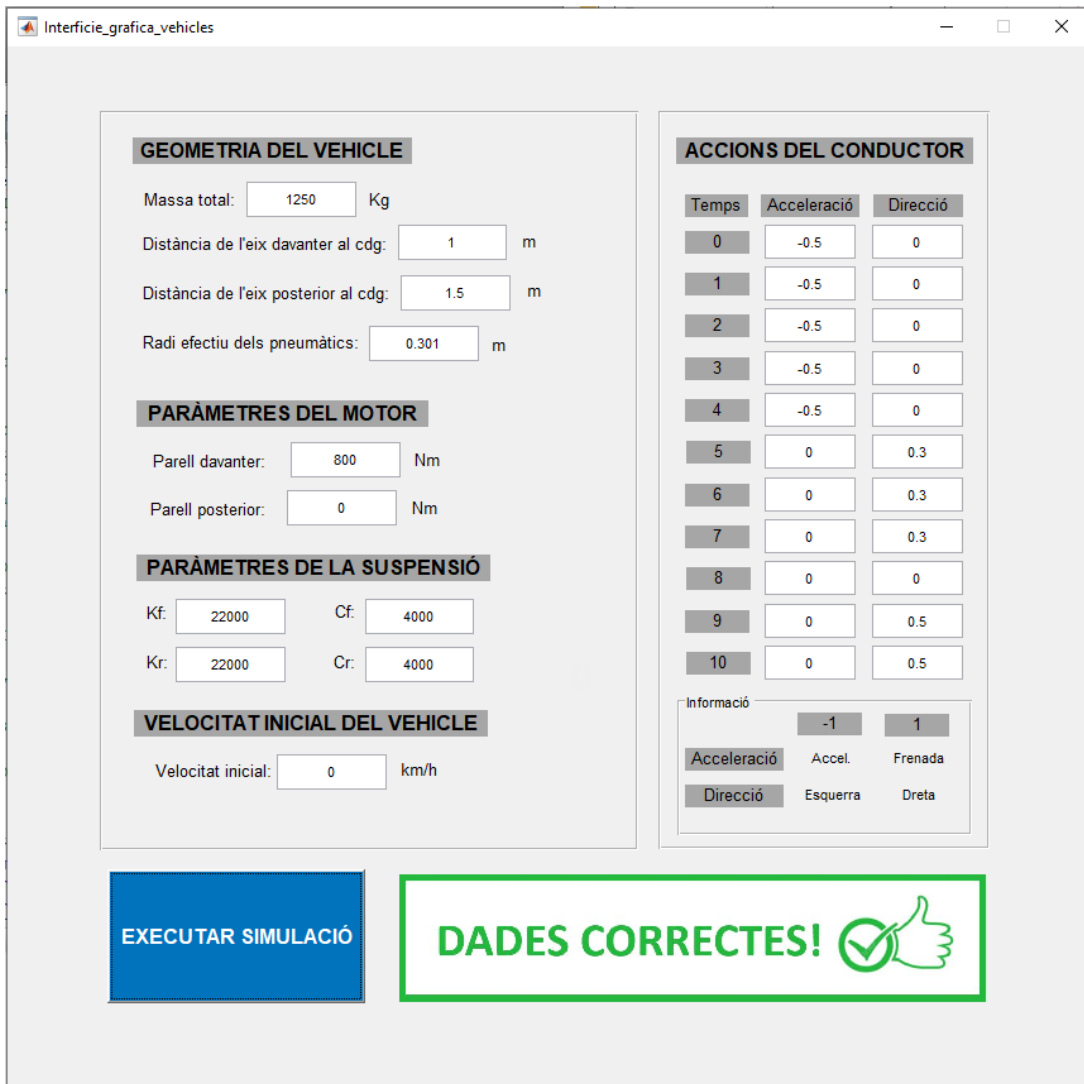


Figura 21: Interficie gràfica amb dades correctes



**GEOMETRIA DEL VEHICLE**

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

**PARÀMETRES DEL MOTOR**

Parell davanter:  Nm

Parell posterior:  Nm

**PARÀMETRES DE LA SUSPENSIO**

Kf:  Cf:

Kr:  Cr:

**VELOCITAT INICIAL DEL VEHICLE**

Velocitat inicial:  km/h

**ACCIONS DEL CONDUCTOR**

Temps	Acceleració	Direcció
0	10	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	0	0.3
6	0	0.3
7	0	0.3
8	0	0
9	0	0.5
10	0	0.5

Informació

Acceleració:

Direcció:

**EXECUTAR SIMULACIÓ**

**REVISAR DADES!**

Figura 22: Interficie gràfica amb dades incorrectes

Un cop les dades han sigut entrades i s'ha comprovat que són correctes el següent pas és executar directament l'arxiu amb la funció de simulació *FWsim* el qual s'encarrega d'executar i representar els resultats aportats per les diverses funcions de càlcul auxiliars a partir del valors entrats mitjançant la interfície gràfica d'entrada de dades.

### 3.1.2. Representació de resultats

Un cop s'ha executat la simulació, el programa ens retorna els resultats de forma gràfica, els quals podem separa en dos grans apartats:

- Uns representació gràfica del comportament en el pla del vehicle a partir de les condicions d'inici i les accions del conductor durant els deu intervals de temps que hem simulat. Es pot apreciar la trajectòria del vehicle sobre l'eix x i y, expressats amb metres així com pot donar-nos una idea de la velocitat i acceleració si ens fixem en la separació entre les diferents figures. A més separació entre figures representa una velocitat més alta, pel que quan aquestes es separen progressivament ens està indicant un procés d'acceleració o frenada si fos el cas contrari.

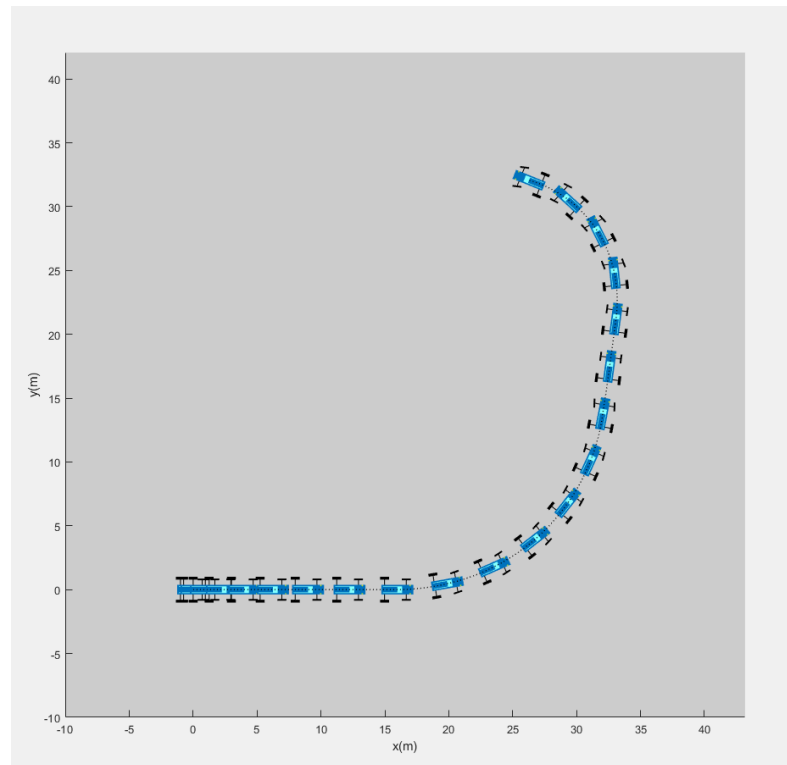


Figura 23: Pantalla resultat de la simulació de la trajectoria

- Un conjunt de quinze gràfics on poder visualitzar diferents paràmetres en funció del temps de simulació que a continuació analitzarem en més detall

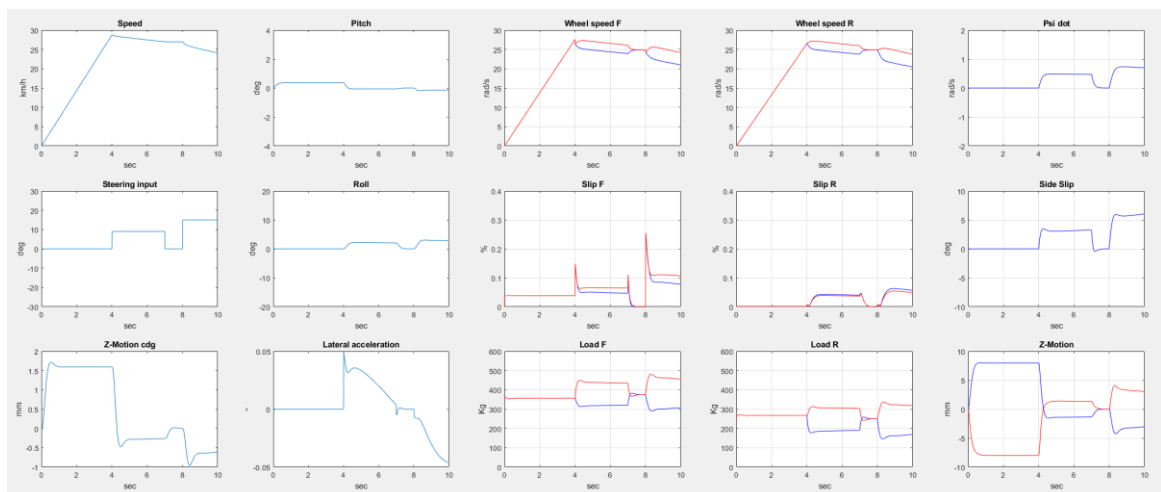


Figura 24: Pantalla de resultats gràfics

**Velocitat:** En aquest gràfic podem apreciar la diferència de velocitat, en quilometres per hora, durant els deu intervals de temps associada a les accions del conductor sobre l'accelerador i el fre.

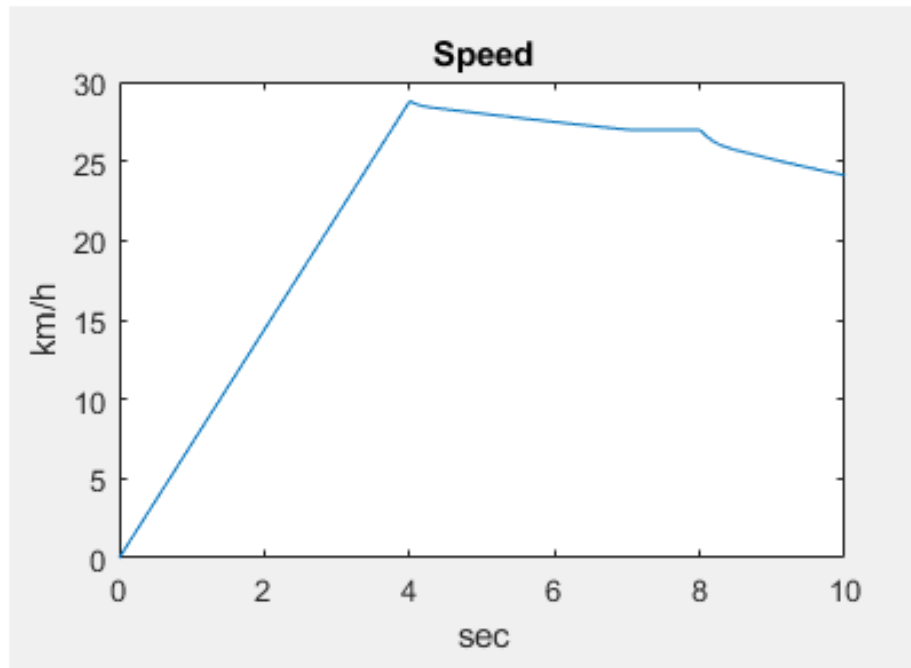


Figura 25: Gràfica de la velocitat

**Variable d'entrada de direcció:** On es pot apreciar el gir de les rodes del vehicle en graus en els diferents intervals de temps introduïts en la matriu d'accions del conductor.

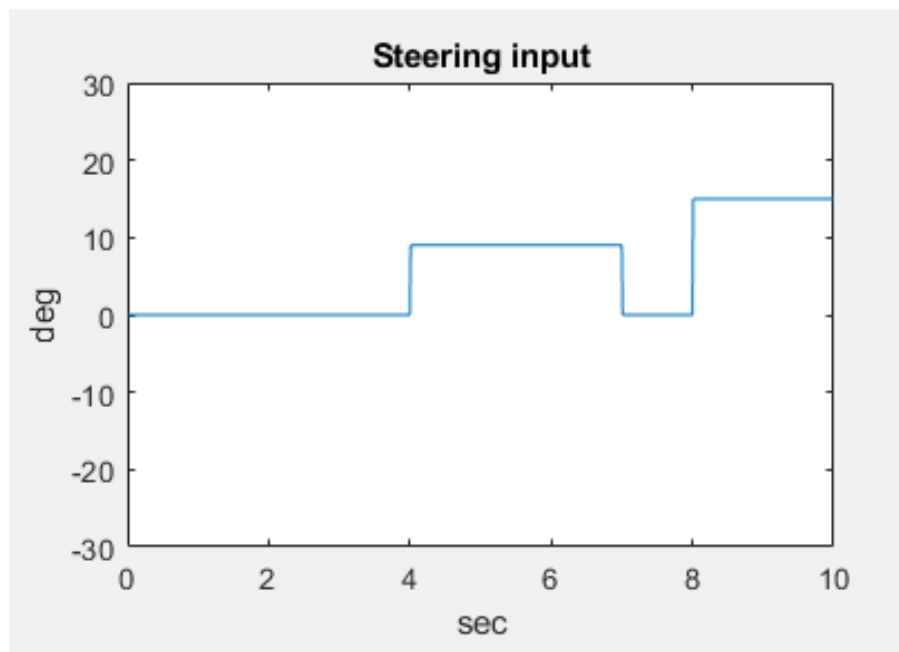


Figura 26: Gràfica de l'input de direcció

**Moviment del centre de gravetat en l'eix Z:** Es pot veure el moviment del vehicle en l'eix vertical en mil·límetres.

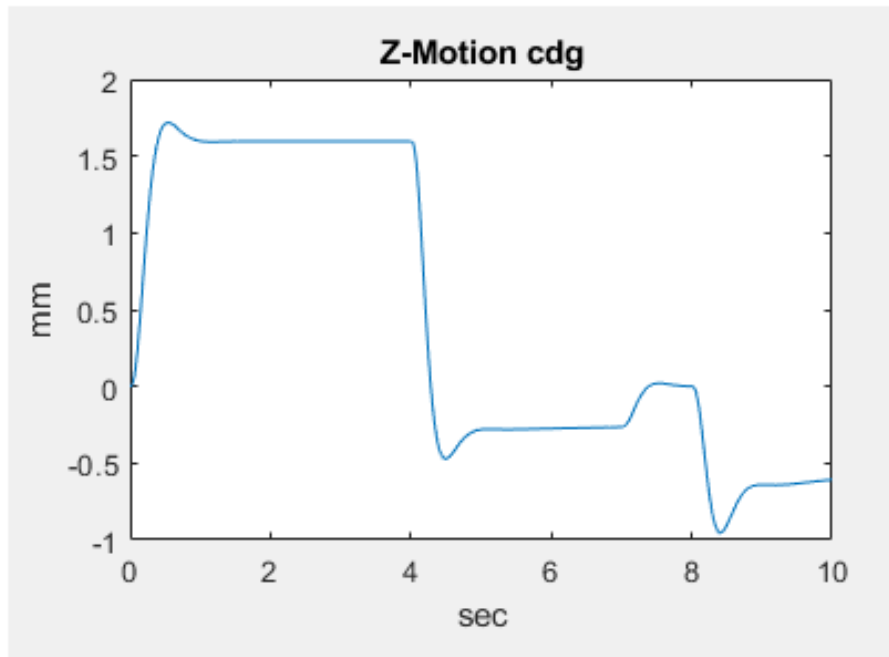


Figura 27: Gràfica del moviment del cdg en l'eix Z

**Pitch:** Ens permet apreciar el desplaçament angular, en graus, sobre l'eix paral·lel a l'eix de les rodes i situat sobre el CDG o balanceig endavant i endarrer en funció del temps

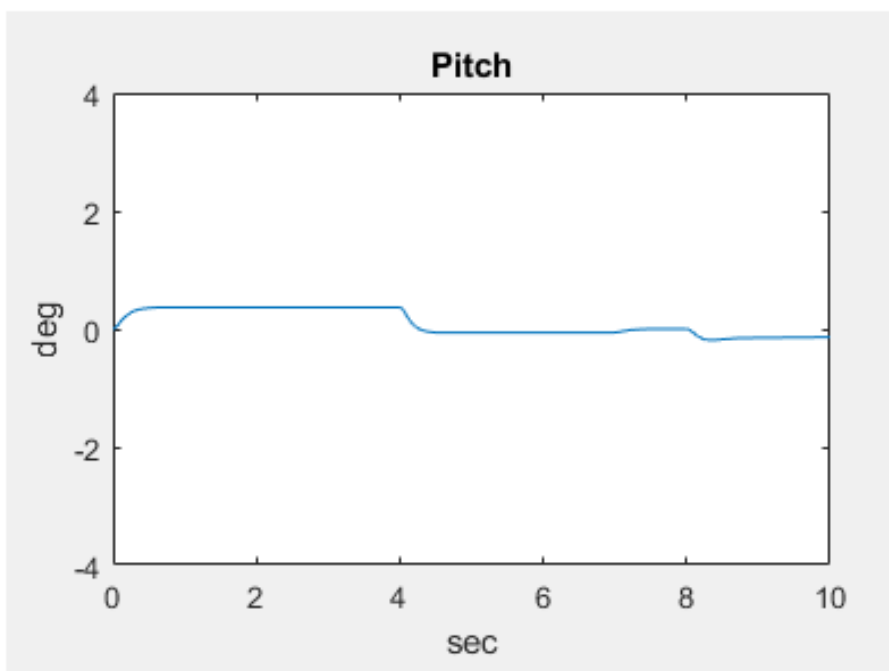


Figura 28: Gràfica del balanceig o "pitch"

**Roll:** Ens permet visualitzar el moviment angular, en graus, sobre l'eix paral·lel a la direcció del cotxe en funció del temps.

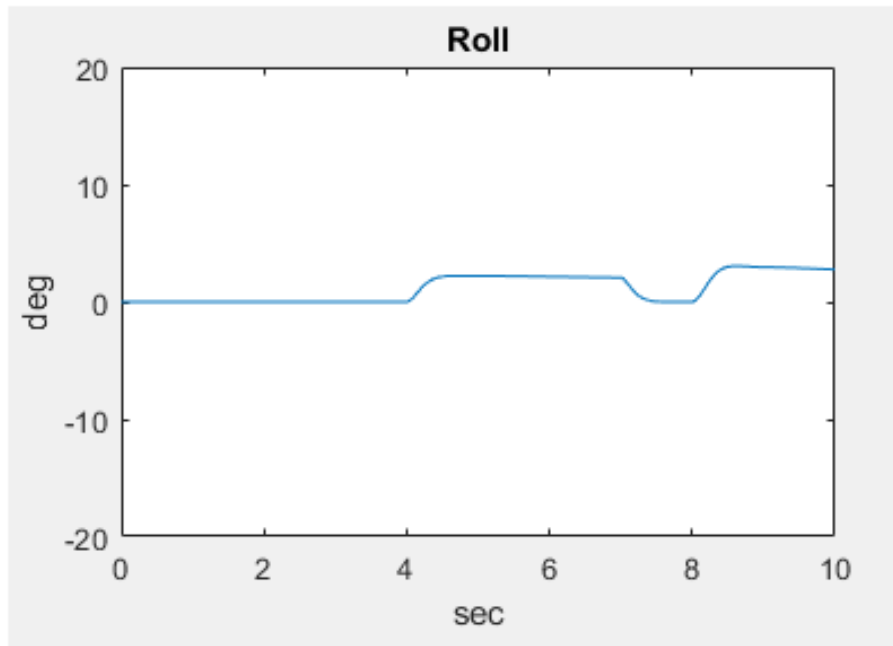


Figura 29: Gràfica del balnceig lateral o "roll"

**Acceleració lateral:** Aquest gràfic ens mostra l'acceleració lateral que pateix el centre de masses del vehicle degut a les forces inercials en trajectòries corbes.

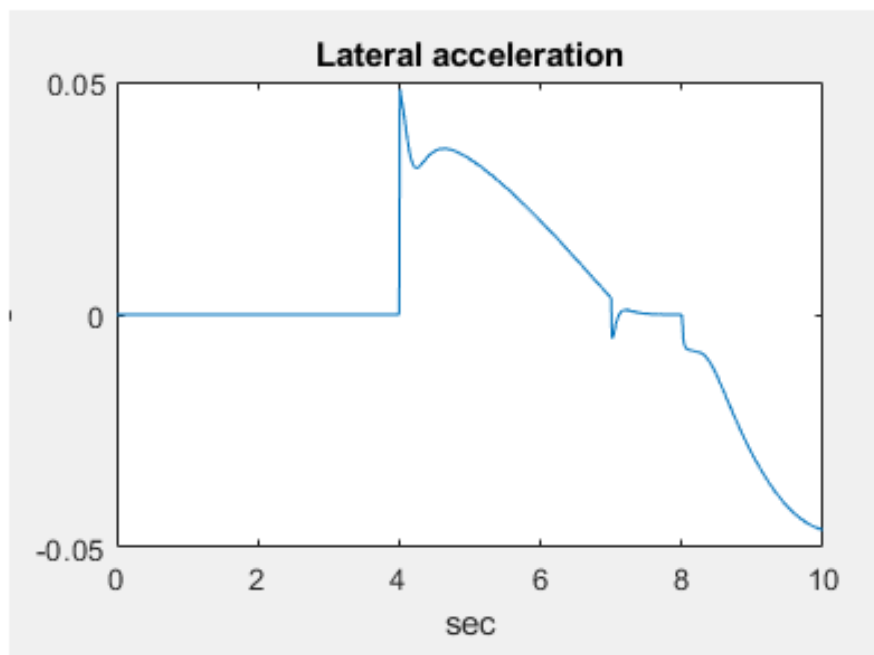


Figura 30: Gràfica de l'acceleració lateral

**Velocitat de les rodes davantera i posterior:** Podem observar la velocitat angular de les diferents rodes, esquerra i dreta tant davanteres com posteriors. Podem diferenciar entre les rodes esquerra i dreta a partir dels colors representats en el gràfic, essent el vermell el referent a la dreta i el blau a l'esquerra.

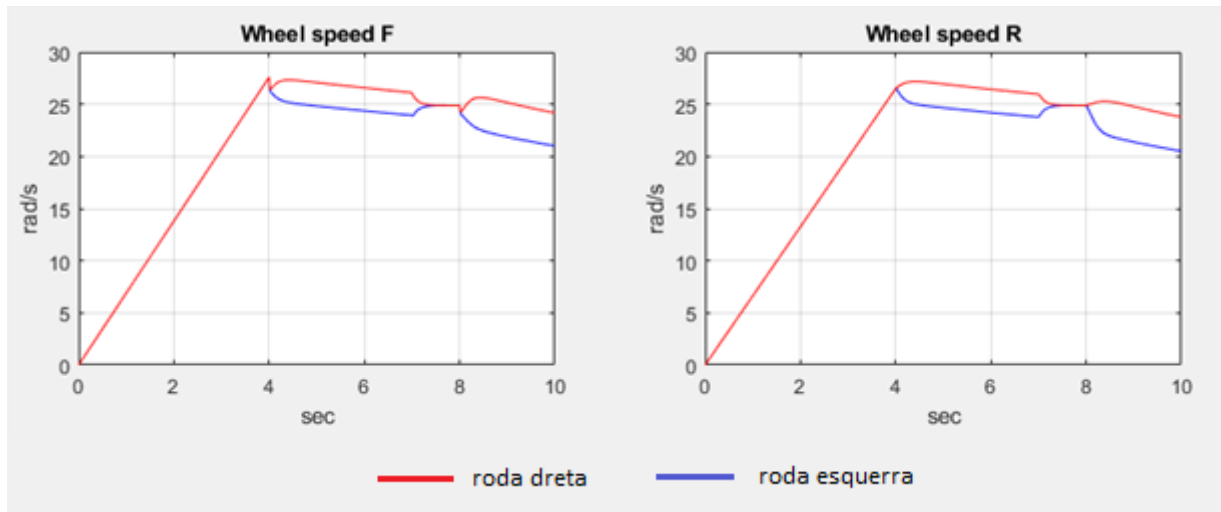


Figura 31: Gràfiques de les velocitats de les rodes davanteres i posteriors

**Lliscament de les rodes davanteres i posteriors:** Podem apreciar el lliscament de les rodes en percentatge, igual que en cas anterior, tan per l'esquerra i dreta davanteres com posteriors

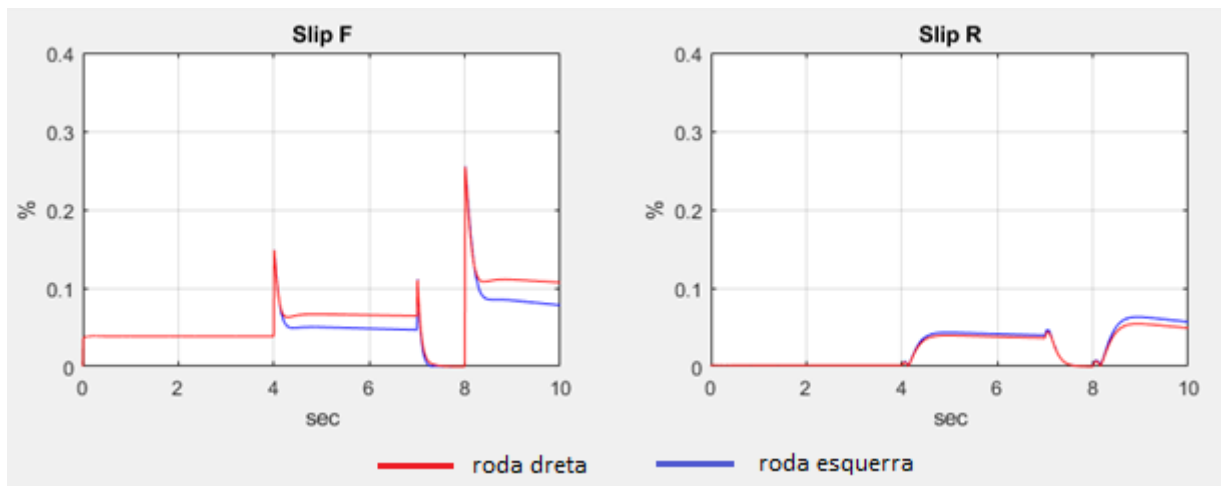


Figura 32: Gràfiques del lliscament dels pneumàtics davanteres posteriors

**Càrrega sobre les rodes davanteres i posteriors:** En aquestes gràfiques podem diferenciar les càrregues aplicades a cada una de les rodes independentment.

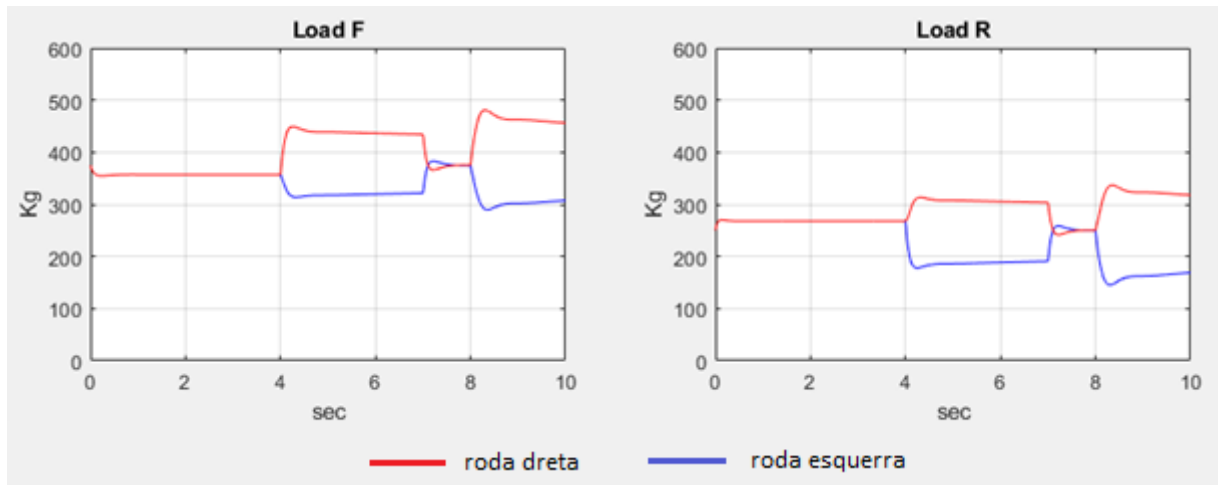


Figura 33: Gràfiques de les càrregues sobre les rodes davanteres i posteriors

**Derivada de  $\Psi$ :** Ens reflecteix l'acceleració angular que pot patir el cotxe al pivotar sobre el seu eix vertical z o moviment de yaw.

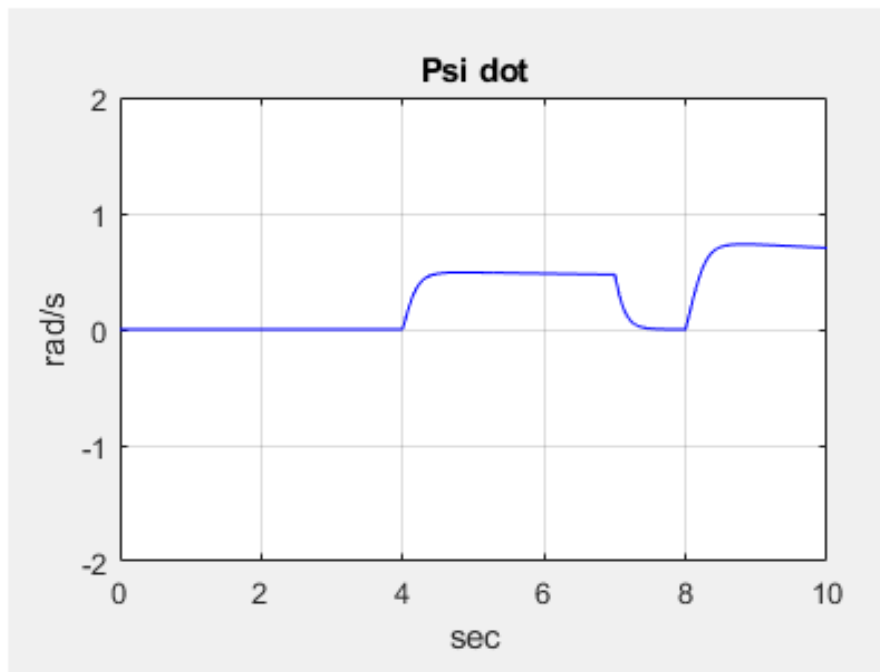


Figura 34: Gràfica de l'acceleració angular en el moviment de "yaw"

**Lliscament lateral:** En aquest gràfic podem apreciar el lliscament lateral del centre de gravetat del vehicle degut a la pèrdua d'adherència dels pneumàtics amb el terra.

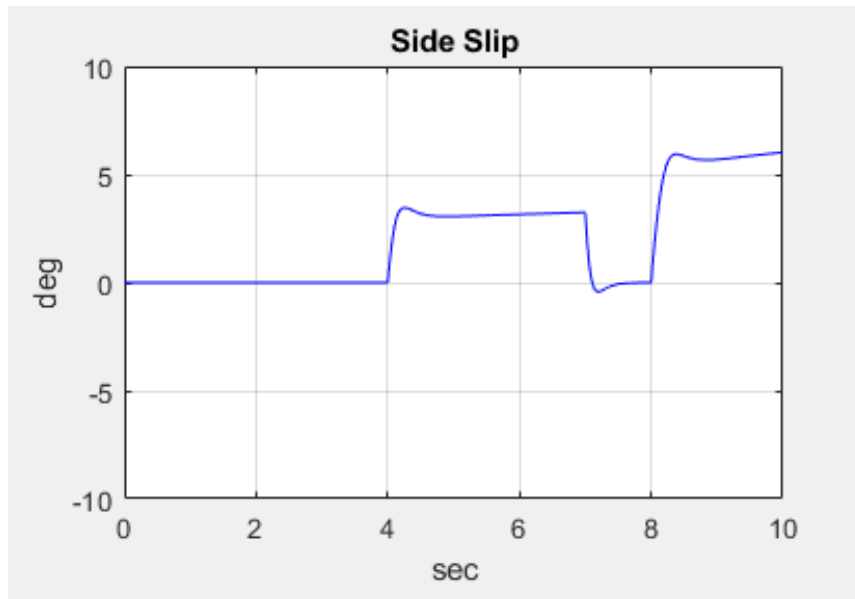


Figura 35: Gràfica del lliscament lateral

**Moviment de les rodes sobre l'eix vertical:** Finalment en l'última representació gràfica ens permet visualitzar el desplaçament vertical del vehicle.

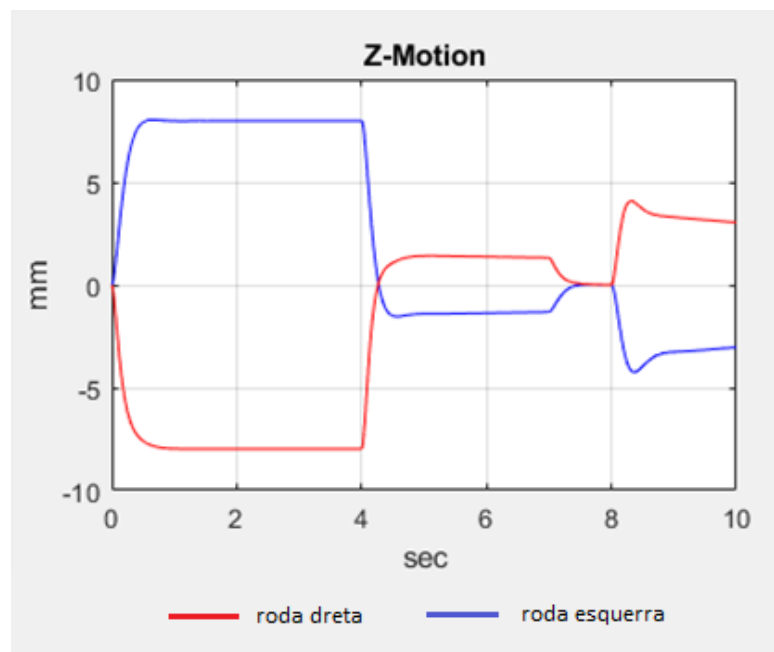


Figura 36: Gràfica del moviment de les rodes en l'eix Z



### 3.2. Models dinàmics: equacions de moviment

En quant a la base de càlcul per a la simulació del comportament dinàmic del vehicle, tenim una funció principal en quant a equacions de moviment la qual s'alimenta de 4 funcions secundàries que realitzen els càlculs pertinents a la direcció i trajectòria, l'acceleració i la frenada, la deriva del vehicle i la fricció sobre la superfície de contacte. A continuació comentarem breument cada una d'aquestes funcions, però podem trobar el codi detallat de totes elles a l'annex.

#### 3.2.1. Direcció i trajectòria

La funció corresponent a la direcció i trajectòria és l'encarregada de llegir les dades entrades per l'usuari sobre la columna dreta de la matriu d'accions del conductor (marcada en vermell) i interpretar-ne els valors per a cada un dels intervals de temps designats.

ACCIONS DEL CONDUCTOR		
Temps	Acceleració	Direcció
0	-0.5	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	0	0.3
6	0	0.3
7	0	0.3
8	0	0
9	0	0.5
10	0	0.5

Informació		
	-1	1
Acceleració	Accel.	Frenada
Direcció	Esquerra	Dreta

Figura 37: Entrada de variables de direcció i acceleració

### 3.2.2. Acceleració i frenada

Pel que fa a la funció corresponent a l'acceleració i frenada, marcada en blau en la figura anterior, de manera similar a l'anterior s'encarrega de llegir les dades introduïdes en la columna d'acceleració en el bloc d'accions del conductor de l'interfície gràfica per a cada un dels intervals de temps designats i posteriorment utilitzar aquests valors per als càlculs pertinents.

### 3.2.3. Deriva

La funció de deriva o *slip* té l'objectiu de calcular el lliscament de cada una de les rodes del vehicle en els dos eixos paral·lels al terra, a partir dels quals troba una resultant per a cada roda aplicant el teorema de Pitàgores.

La variable d'entrada d'usuari que afecta directament aquesta funció és la del radi del pneumàtic del vehicle. També influeixen directament en el càlcul les velocitats de cada una de les rodes sobre els eixos x i y així com la seva velocitat angular.

$$slip = \frac{w \cdot r - v}{v} \quad (Eq. 34)$$

### 3.2.4. Fricció

En la funció del càlcul de la fricció dels pneumàtics del vehicle sobre el terra intervenen varies de les variables d'entrada que podem modificar sobre la interfície de treball. Aquestes són les distàncies des del centre de gravetat fins a l'eix davanter i l'eix posterior, així com els paràmetres per al càlcul de l'efecte de les suspensions del vehicle.

El plantejament general de la funció és el de calcular el coeficient de fricció dinàmic de cada una de les rodes, calcular les forces aplicades també sobre cada una de les rodes, per així finalment fer-ne el producte i obtenir les forces de lliscament sobre cada roda i en els diferents eixos geomètrics.

## 4. SIMULACIÓ

A Continuació s'utilitzarà el model de simulació dinàmic per a l'avaluació de tres casos representatius en la trajectòria de vehicles. Els resultats d'aquestes simulacions seran analitzats i comparats per poder apreciar les principals diferències en el comportament dels vehicles. Un dels aspectes que més ens interessa avaluar i comparar és la diferència en el comportament per a cada un dels casos depenent de la configuració de tracció del vehicle, és a dir si aquest té una tracció davantera o posterior amb el conseqüent desplaçament del centre de gravetat del vehicle.

Els casos avaluats i que a continuació veurem en detall són el de l'acceleració rectilínia d'un vehicle partint d'una velocitat inicial de zero, el de l'acceleració i posterior frenada rectilínia partint d'una certa velocitat inicial, i finalment el d'una trajectòria corba a velocitat constant.

En tots els casos s'han utilitzat unes característiques del vehicle idèntiques a diferència de la tracció i la distància al centre de gravetat dels eixos davanter i posterior, per tal de poder comparar els resultats esperats entre ells. A continuació detallarem les dades emprades:

- Massa total del vehicle: 1250 kg.
- Distància del cdg a l'eix davanter: 1 metre quan la tracció és davantera o 1,5 quan és posterior.
- Distància del cdg a l'eix posterior: 1,5 metre quan la tracció és davantera o 1 quan és posterior.
- Radi efectiu dels pneumàtics: 0,301 metres.
- Parell davanter: 800 Nm quan la tracció és davantera o 0 Nm quan és posterior.
- Parell posterior: 0 Nm quan la tracció és davantera o 800 Nm quan és posterior.
- $K_f$  i  $K_r$ : 22000.
- $C_f$  i  $C_r$ : 4000.

### 4.1. Cas 1: Acceleració rectilínia

En aquest primer es vol analitzar el comportament d'un vehicle el qual parteix d'una posició de repòs i accelera de forma constant, prement el pedal del gas a un 50 per cent durant els deu segons de la simulació i mantenint la direcció estable i en angle recte.

En primer lloc executarem la simulació per al cas amb tracció davantera, seguidament el mateix cas amb tracció posterior i finalment compararem els resultats entre ells:

- Vehicle amb tracció davantera:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.5	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	-0.5	0
6	-0.5	0
7	-0.5	0
8	-0.5	0
9	-0.5	0
10	-0.5	0

Informació

-1
1

Acceleració

Accel.

Frenada

Direcció

Esquerra

Dreta

EXECUTAR SIMULACIÓ

## DADES CORRECTES!

Figura 38: Entrada de variables Cas 1.1

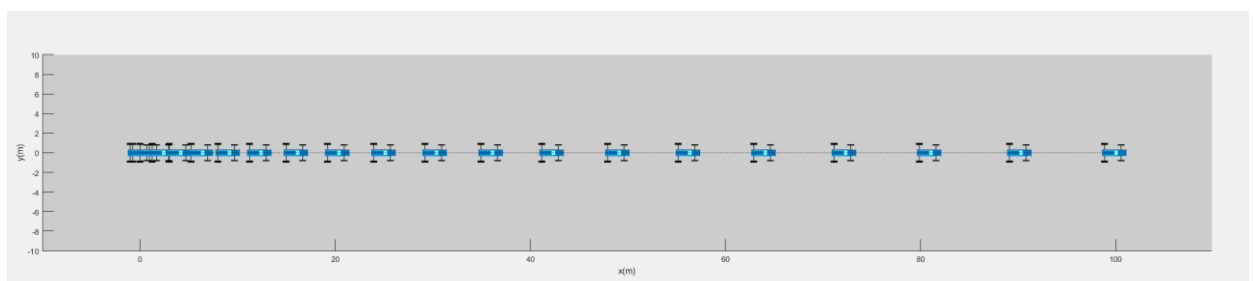


Figura 39: Trajectoria Cas 1.1

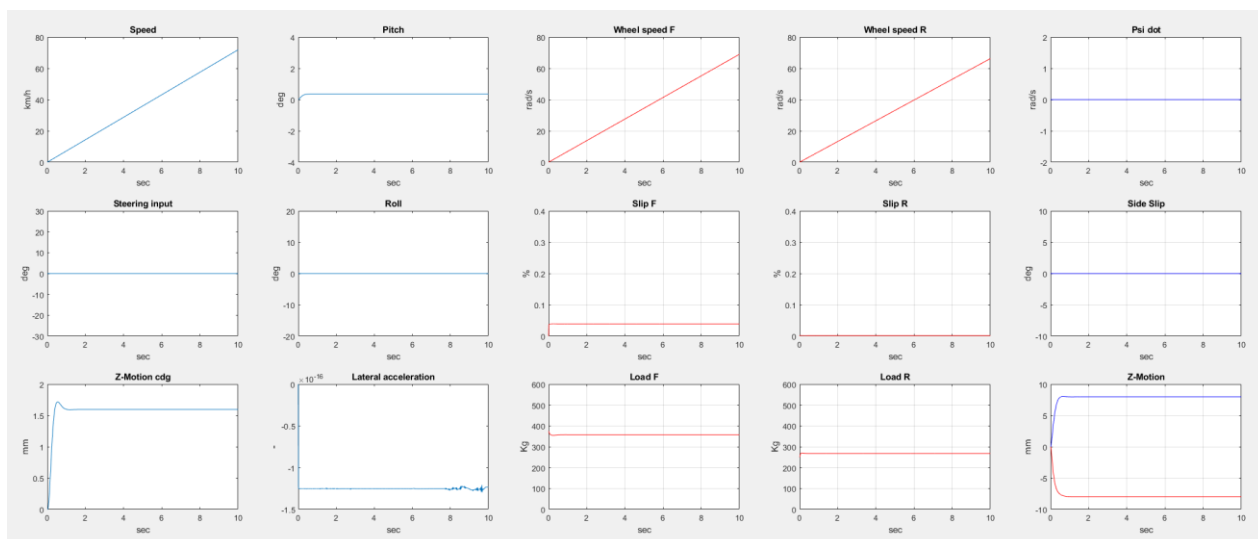


Figura 40: Resultats gràfics Cas 1.1

- Vehicle amb tracció posterior:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.5	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	-0.5	0
6	-0.5	0
7	-0.5	0
8	-0.5	0
9	-0.5	0
10	-0.5	0

Informació

-1
1

Acceleració

Accel.

Frenada

Direcció

Esquerra

Dreta

**EXECUTAR SIMULACIÓ**

**DADES CORRECTES!**

Figura 41: Entrada variables cas 1.2

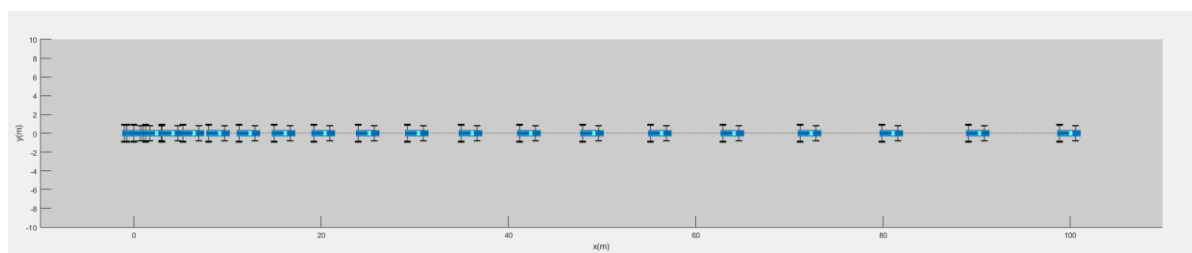


Figura 42: Trajectoria cas 1.2

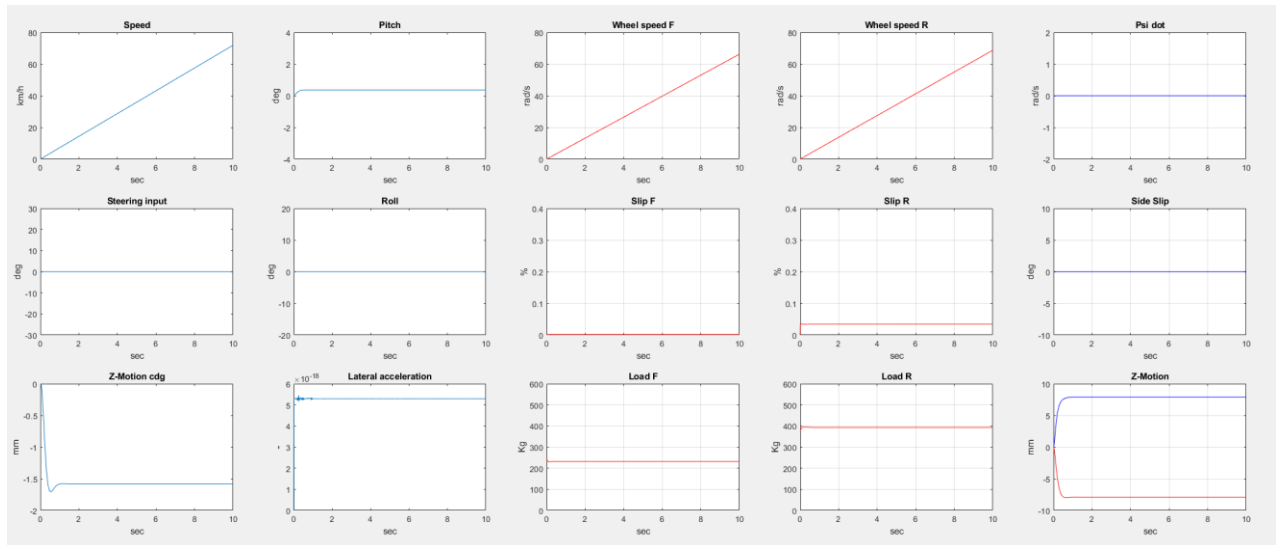


Figura 43: Resultats gràfics cas 1.2

- **Comparativa entre les dues configuracions:**

Per aquest primer cas i després d'haver executat les simulacions per a totes dues configuracions de tracció podem analitzar aquells paràmetres que coincideixen i aquells que difereixen en cada una d'aquestes.

En aquestes primeres simulacions podem apreciar valors significatius en els paràmetres que afecten l'eix longitudinal del vehicle ja que la trajectòria és totalment recta.

La velocitat per exemple, igual per ambdues configuracions, és com era d'esperar una recta amb pendent positiu i constant degut a l'acceleració continua i igual durant tot l'interval de temps de la simulació:

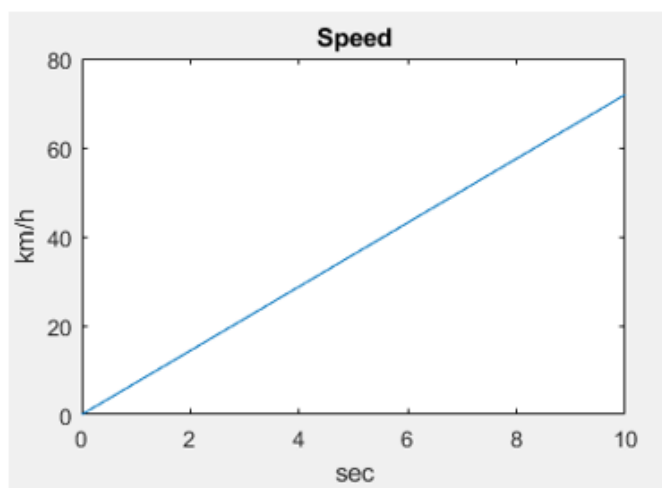


Figura 44: Velocitat cas 1

El més interessant però, és l'anàlisi d'aquells paràmetres en el que si que hi ha discrepàncies entre una simulació i l'altra. En el costat esquerra de la comparativa veurem el

resultat de la simulació per al vehicle amb tracció davantera i en el dret el del vehicle amb tracció posterior per als diferents paràmetres implicats.

En primer cas trobem gràfics totalment oposats pel que fa al moviment del centre de gravetat del vehicle en l'eix vertical, desplaçament causat per el paper de la suspensió del vehicle la qual es comporta d'una manera o d'una altra en funció de l'eix tractor del vehicle.

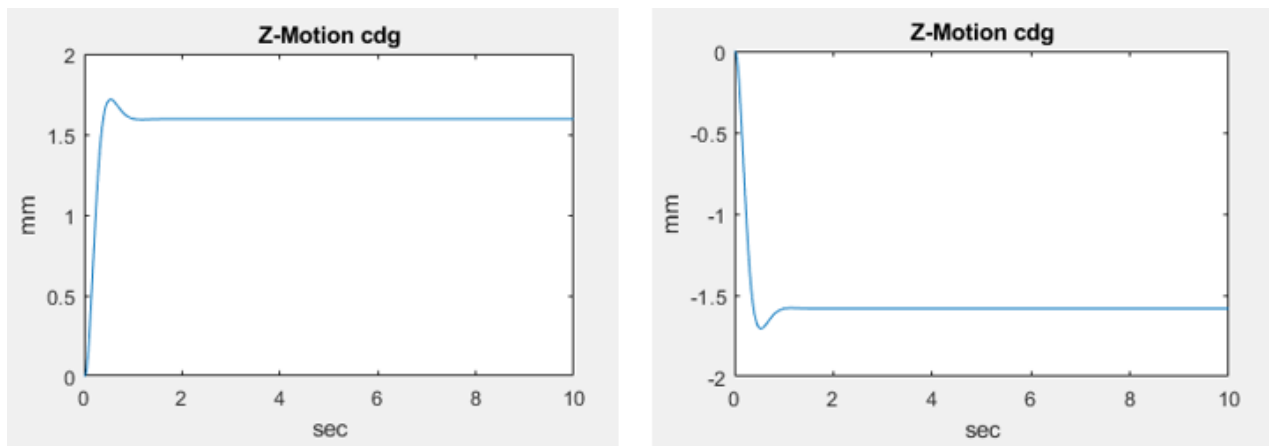


Figura 45: Gràfiques z-motion cas 1

On si s'aprecien petites diferències és en les velocitats de les diferents rodes analitzades per separat. Per exemple en el cas de les rodes davanteres veiem que la velocitat final és lleugerament més alta en el costat esquerre, corresponent al vehicle amb tracció davantera.

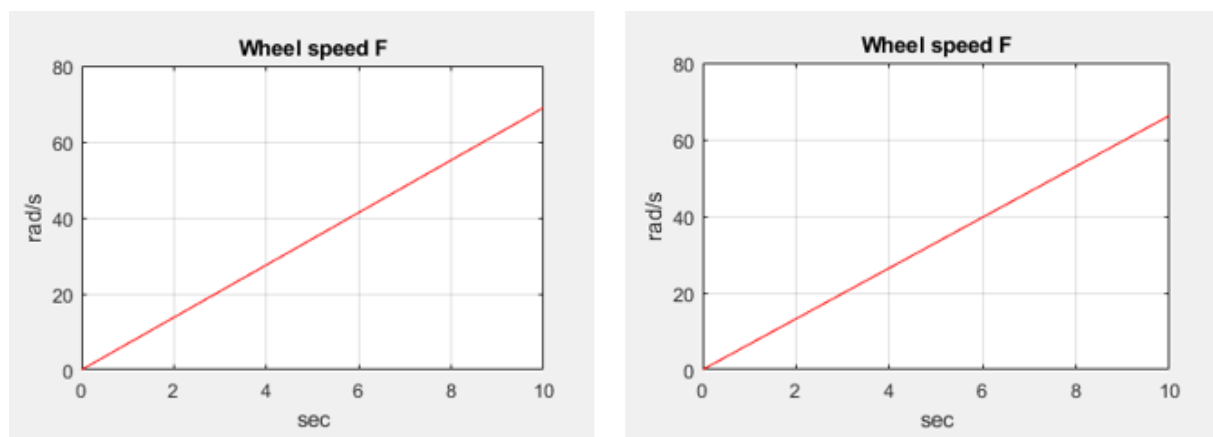


Figura 46: Gràfiques velocitat rodes davanteres cas 1

Per contrari, la velocitat de les rodes posteriors és lleugerament superior en la simulació del vehicle amb tracció posterior, reflectida en el gràfic del costat dret.



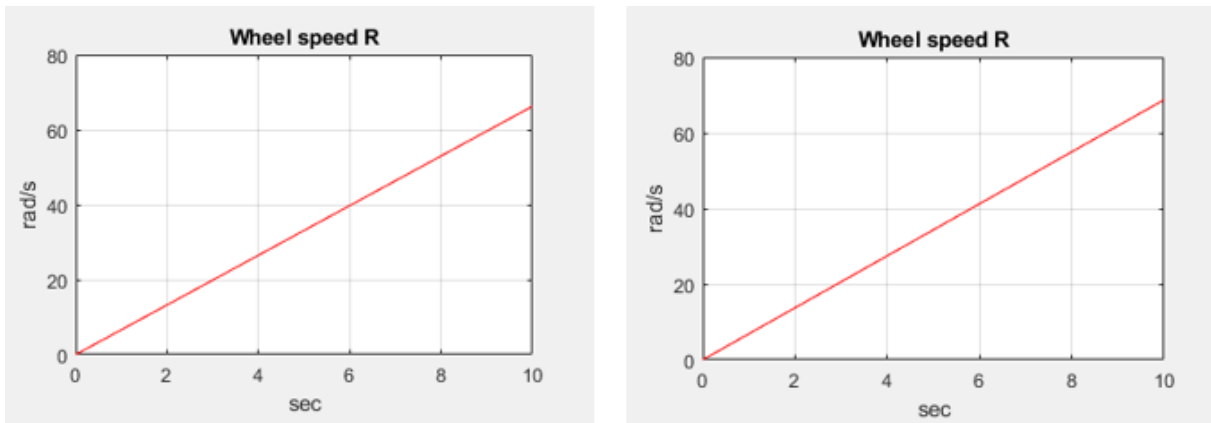


Figura 47: Gràfiques velocitat rodes posteriors cas 1

Seguidament podem comparar el lliscament de les rodes davanteres i posteriors en tan per cent i com era d'esperar tenim que en el vehicle amb tracció davantera les rodes davanteres tenen un cert lliscament mentre que en les posteriors el lliscament és zero, i en la simulació del vehicle amb tracció posterior podem veure tot el contrari, en les rodes davanteres el lliscament es zero mentre que en les posteriors pren un valor superior.

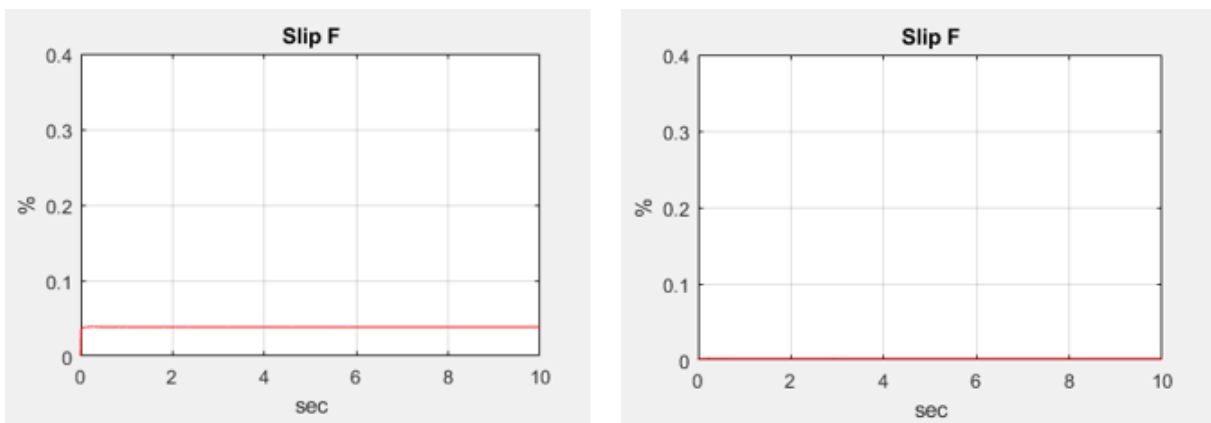


Figura 48: Gràfiques lliscament rodes davanteres cas 1

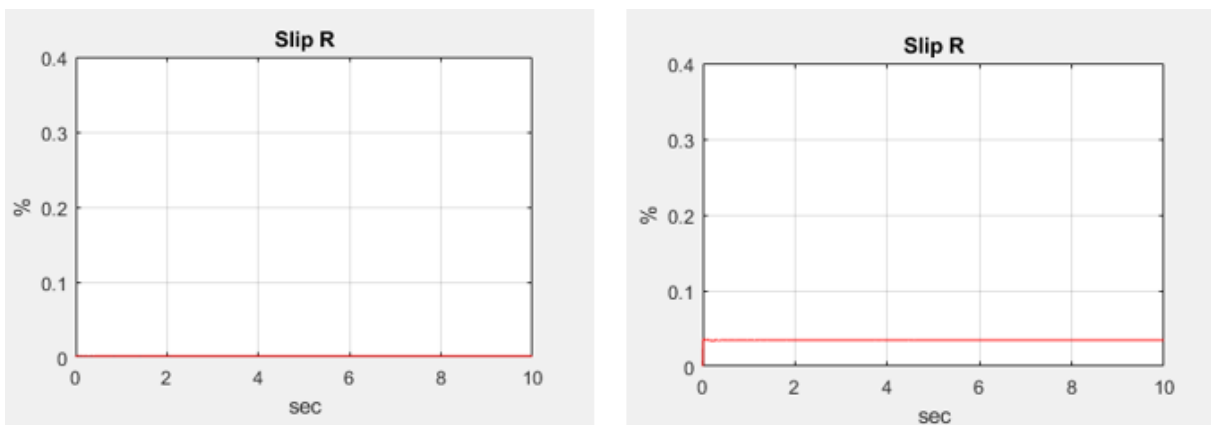


Figura 49: Gràfiques lliscament rodes posteriors cas 1

A continuació un dels paràmetres més significatius és el de les carregues sobre cadascuna de les rodes. El factor que més afecta a aquest paràmetre és la distància del centre de gravetat a cadascun dels eixos, trobant més pes sobre l'eix de tracció en ambdós casos. Tot i així es pot apreciar com la carrega es desplaça lleugerament cap a les rodes posteriors en ambdues configuracions degut a l'efecte de l'acceleració rectilínia.

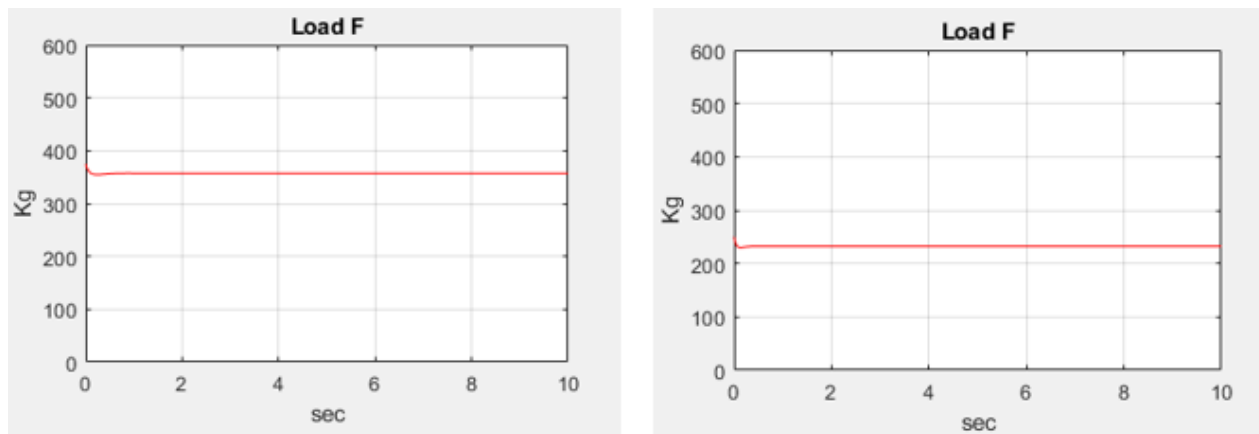


Figura 50: Gràfiques càrregues rodes davanteres cas 1

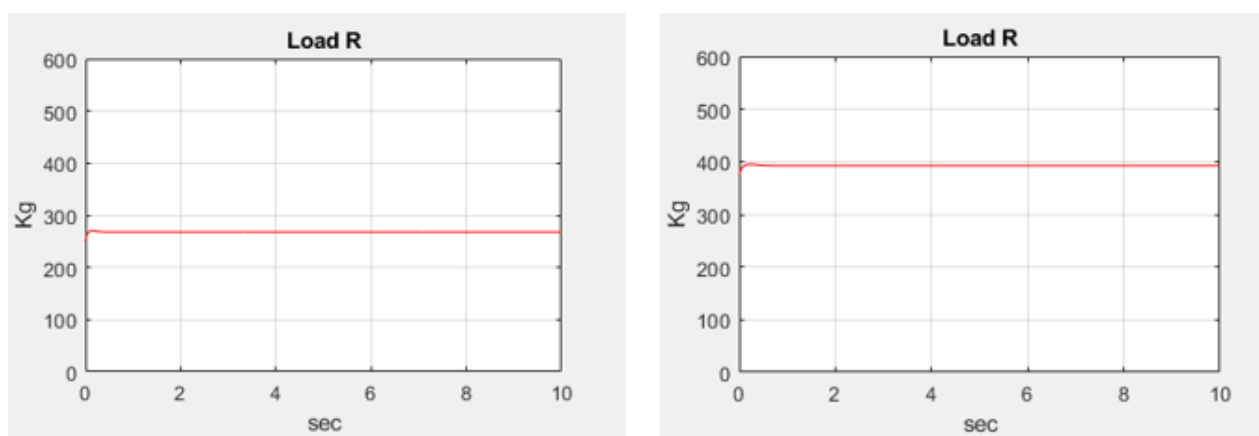


Figura 51: Gràfiques carregues rodes posteriors cas 1

## 4.2. Cas 2: Acceleració i frenada rectilínia

En aquest segon cas es vol avaluar el comportament del vehicle davant d'una acceleració prement el pedal del gas en un 50 per cent durant 5 segons, aquest cop però, portant ja una velocitat inicial de 20 quilometres per hora. Després d'aquests 5 segons d'acceleració es traurà el peu de l'accelerador per prémer el fre a un 20 per cent de la seva capacitat.

En primer lloc executarem la simulació per al cas amb tracció davantera, seguidament el mateix cas amb tracció posterior i finalment compararem els resultats entre ells:

- Vehicle amb tracció davantera:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.5	0
1	-0.5	0
2	-0.5	0
3	-0.5	0
4	-0.5	0
5	-0.5	0
6	0.2	0
7	0.2	0
8	0.2	0
9	0.2	0
10	0.2	0

Informació

-1      1

Acceleració    Accel.    Frenada

Direcció        Esquerra    Dreta

EXECUTAR SIMULACIÓ

DADES CORRECTES!

Figura 52: Entrada de dades cas 2.1

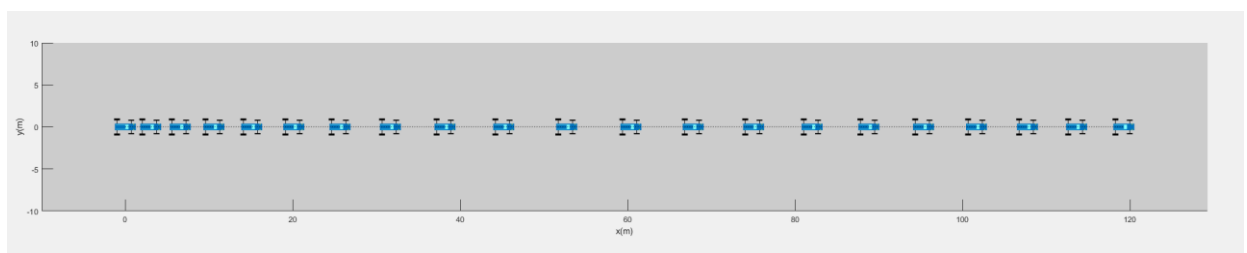


Figura 53: Trajectoria cas 2.1

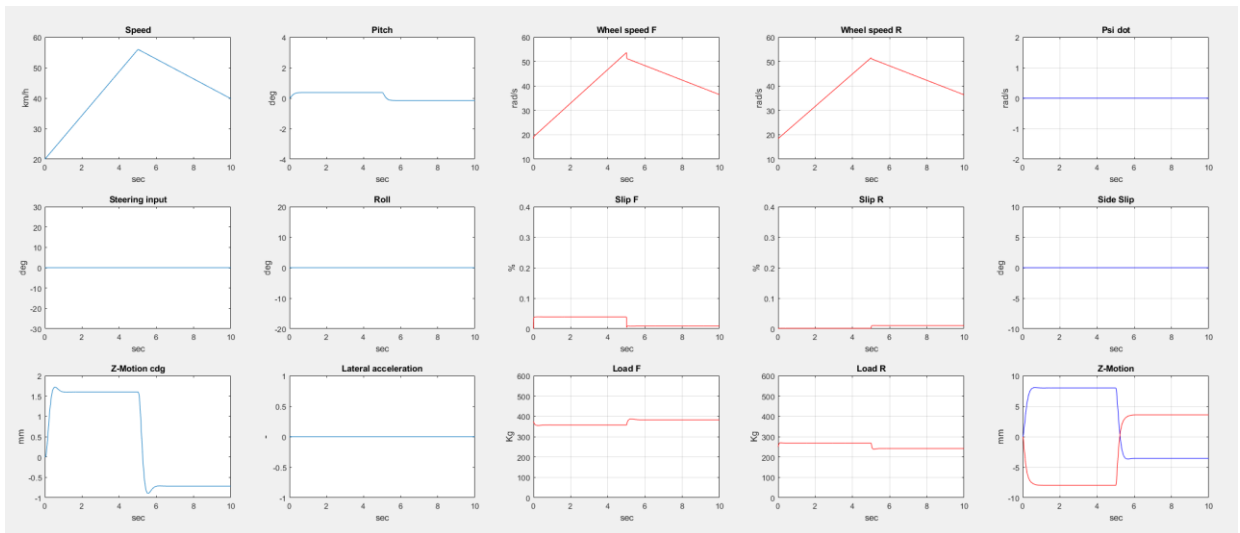


Figura 54: Resultats gràfics cas 2.1

- Vehicle amb tracció posterior:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
1	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
2	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
3	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
4	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
5	<input type="text" value="-0.5"/>	<input type="text" value="0"/>
6	<input type="text" value="0.2"/>	<input type="text" value="0"/>
7	<input type="text" value="0.2"/>	<input type="text" value="0"/>
8	<input type="text" value="0.2"/>	<input type="text" value="0"/>
9	<input type="text" value="0.2"/>	<input type="text" value="0"/>
10	<input type="text" value="0.2"/>	<input type="text" value="0"/>

Informació

-1
1

Acceleració    Accel.    Frenada

Direcció        Esquerra    Dreta

EXECUTAR SIMULACIÓ

DADES CORRECTES!

Figura 55: Entrada de dades cas 2.2

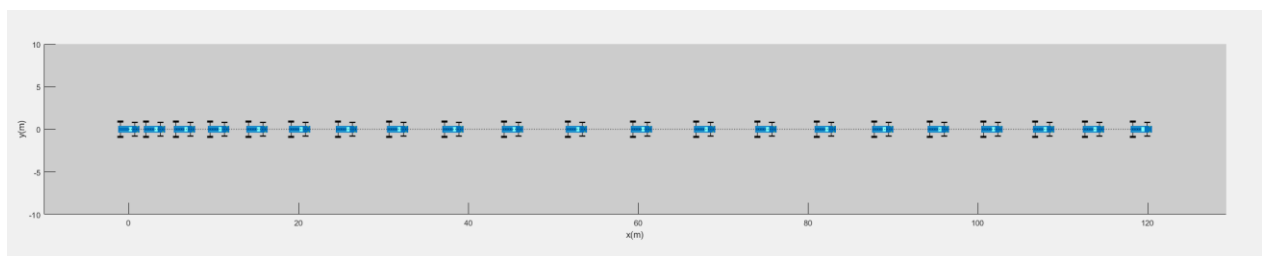


Figura 56: Trajectòria cas 2.2

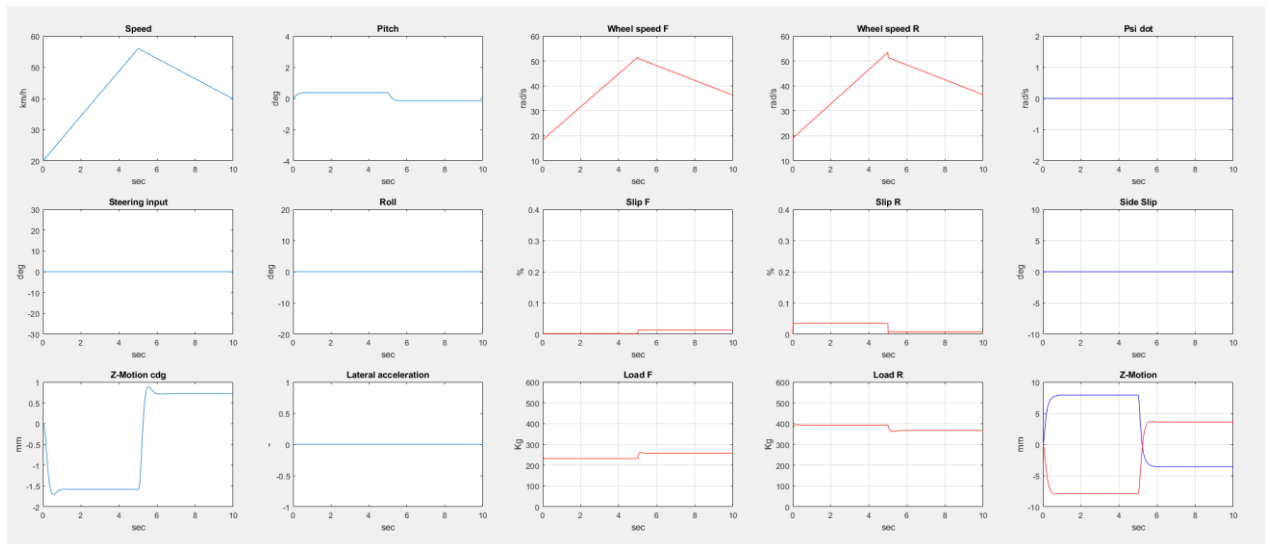


Figura 57: Resultats gràfics cas 2.2

- **Comparativa entre les dues configuracions:**

Per aquest segon cas i després d'haver executat les simulacions per a totes dues configuracions de tracció podem analitzar aquells paràmetres que coincideixen o difereixen en cada una d'aquestes.

En primer lloc observem la gràfica de velocitat, la qual és igual per a totes dues configuracions i en la que es pot apreciar una pendent positiva i constant fins al segon 5, deguda a l'acceleració, seguida per una pendent negativa menys pronunciada deguda a 'acció de frenat. També cal remarcar que en aquest cas el punt de partida és ja de 20 quilometres per hora i no 0 com en el cas anterior.

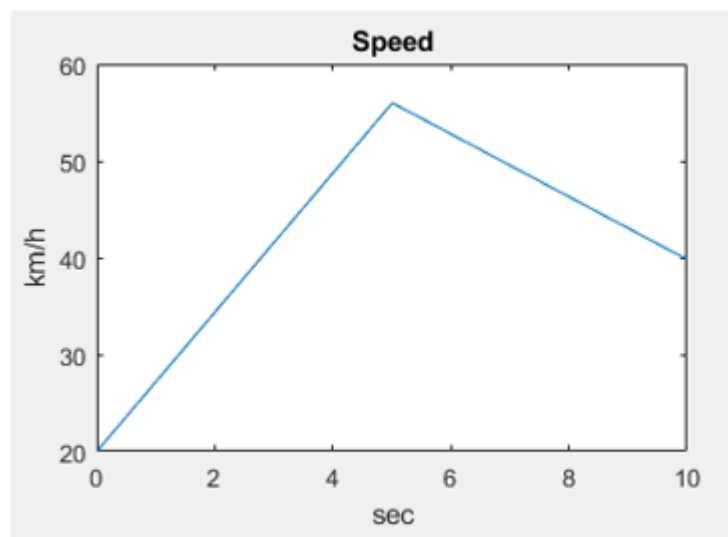


Figura 58: Gràfica velocitat cas 2

Seguidament comentarem les principals diferències que trobem entre una configuració i l'altra per a aquesta mateixa situació de conducció, tenint com en el cas anterior, a

l'esquerra els resultats de la simulació amb la configuració de tracció davantera i a la dreta la de tracció posterior.

Pel que fa al moviment sobre l'eix vertical del centre de gravetat del vehicle, com en el cas anterior trobem dues gràfiques totalment oposades en les que s'aprecia un canvi important degut al pas de l'acceleració a la frenada. La variació de la posició del centre de gravetat en l'eix z és un comportament degut a la suspensió del vehicle i al comportament que té en funció de l'eix que transmet la força motriu.

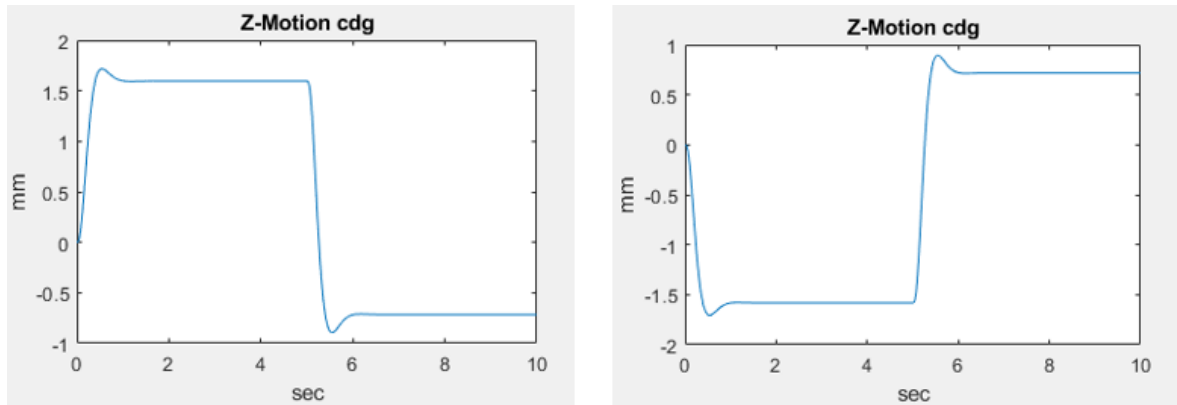


Figura 59: Gràfiques z-motion cas 2

Seguidament podem observar les diferències entre les velocitats de les diferents rodes, tant davanteres com posteriors. Podríem dir que les gràfiques simètriques entre les rodes de davant i de darrere d'una configuració i l'altra. El que s'aprecia és que en les rodes motrius de cada una de les configuracions trobem un petit esglaó en l'instant de canvi d'acceleració a frenada, fet que pot ser produït per el petit lliscament que tenien les rodes durant l'acceleració i que a continuació podem veure gràficament.

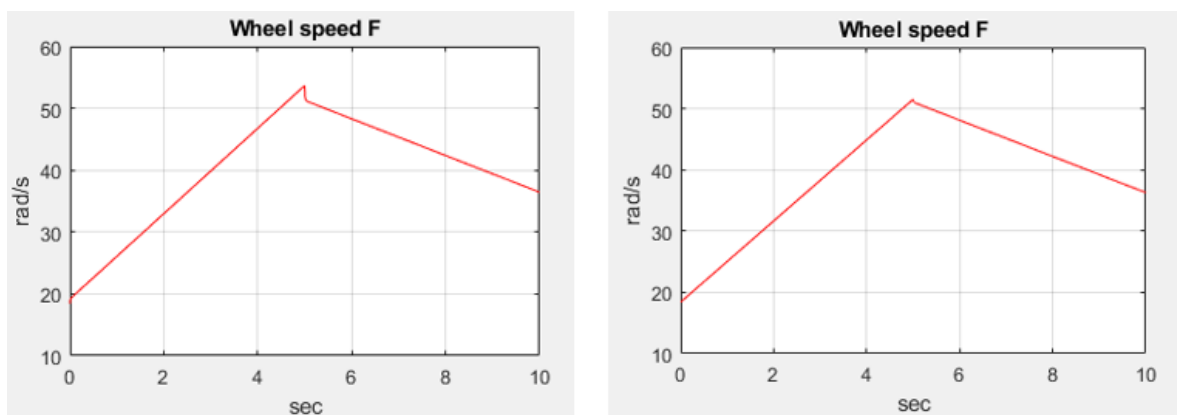


Figura 60: Gràfiques velocitat rodes davanteres cas 2

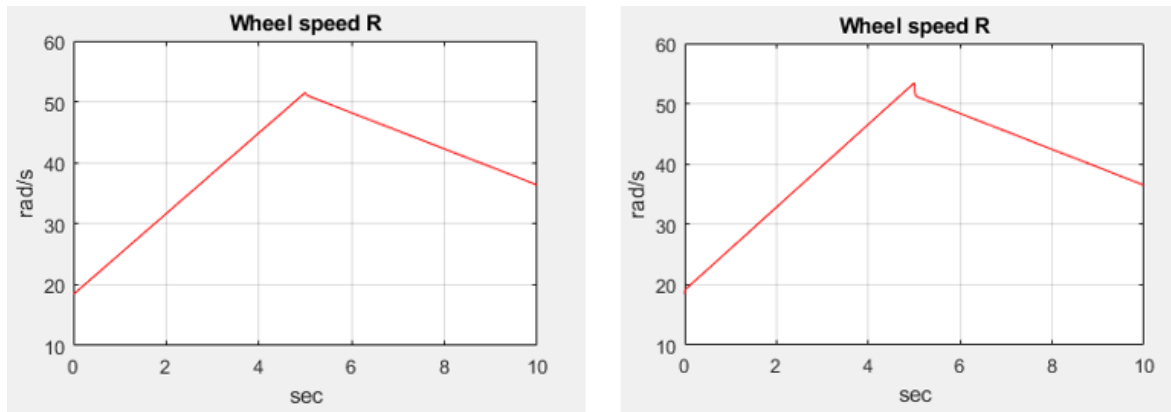


Figura 61: Gràfiques velocitat rodes posteriors cas 2

Com acabem de comentar, les rodes tractores pateixen un lleuger lliscament durant l'acceleració, mentre que en la frenada aquest és molt més baix però present tant en les rodes davanteres com en les posteriors.

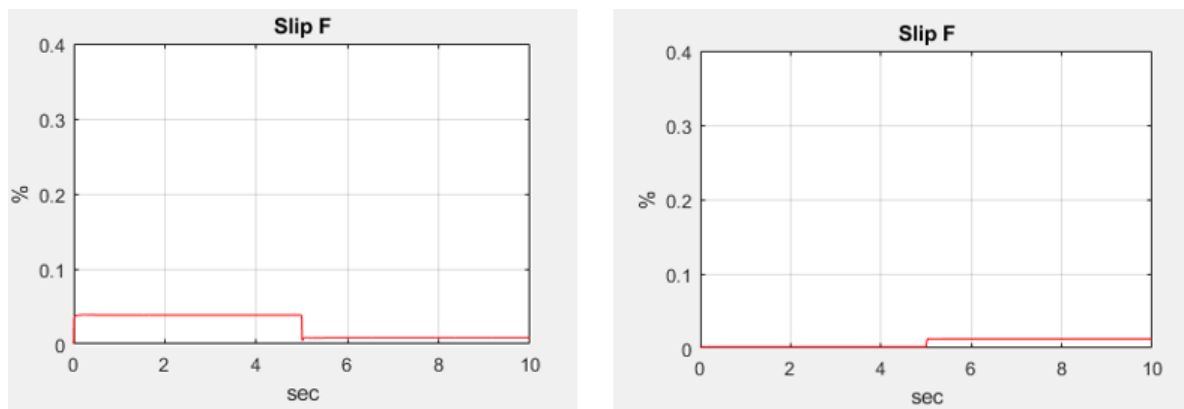


Figura 62: Gràfiques lliscament rodes davanteres cas 2

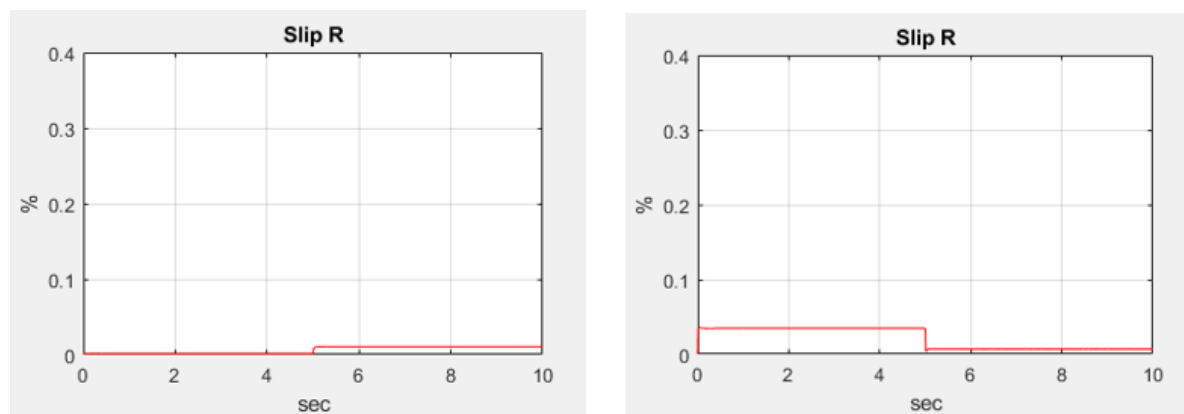


Figura 63: Gràfiques lliscament rodes posteriors cas 2



Finalment podem veure que, igual que en el Cas 1, la càrrega sobre les rodes és més alta sempre sobre les rodes que proporcionen la tracció al vehicle degut principalment al desplaçament del centre de gravetat provocat per la distribució dels elements de tracció.

Un dels fenòmens que si és interessant de comentar és que en les rodes davanteres podem veure un esglaó ascendent en el moment d'inici de la frenada, degut a la inèrcia del vehicle en la direcció de moviment, en canvi en les rodes posteriors i en totes dues configuracions podem observar que aquest esglaó és decreixent, degut igualment a aquesta transmissió de càrrega en el moment de frenada.

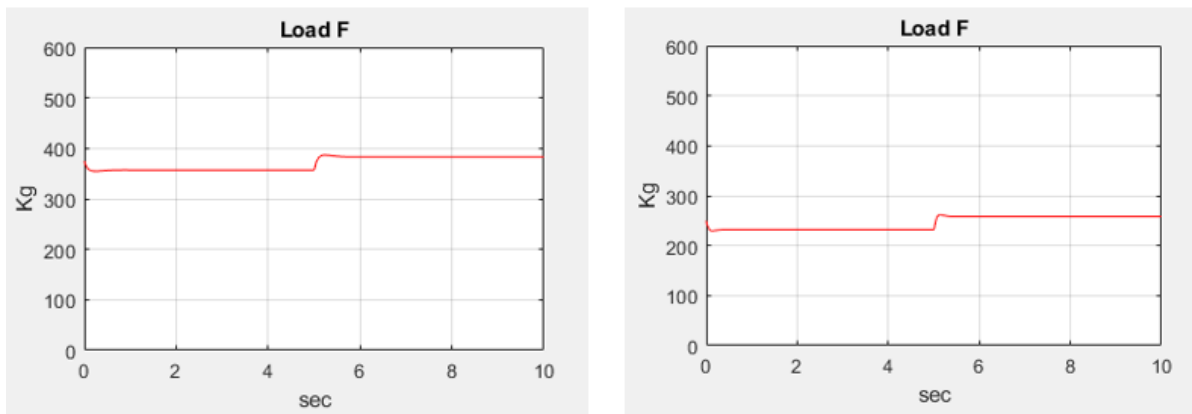


Figura 64: Gràfiques càrregues rodes davanteres cas 2

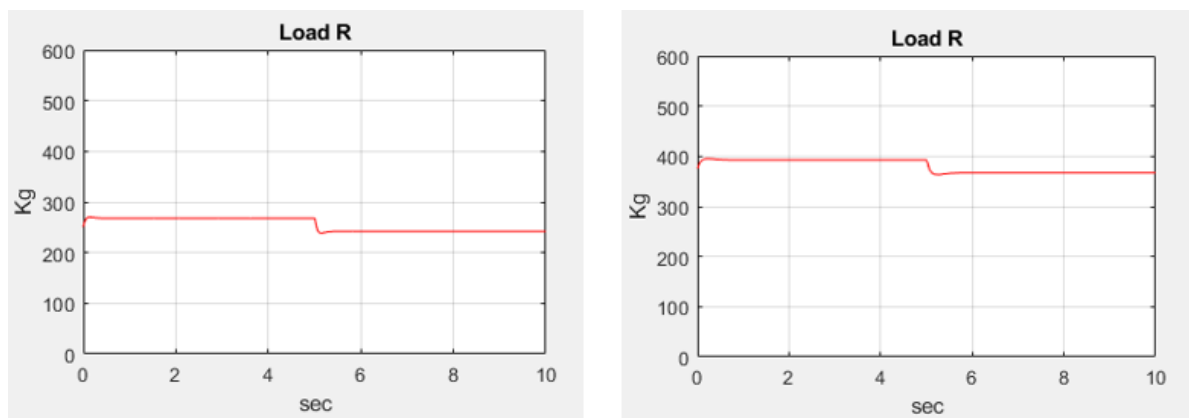


Figura 65: Gràfiques càrregues rodes posteriors cas 2

### 4.3. Cas 3: Corba a velocitat constant

En aquest tercer cas l'objectiu és el de veure la trajectòria i comportament del vehicle prenent una corba amb un gir en la direcció equivalent a 15 graus i a una velocitat constant de 20 quilòmetres per hora. Per aconseguir mantenir aquesta velocitat constant es prem l'accelerador amb un punt de gas, equivalent a un 1 per cent del seu recorregut.

En primer lloc executarem la simulació per al cas amb tracció davantera, seguidament el mateix cas amb tracció posterior i finalment compararem els resultats entre ells:

- Vehicle amb tracció davantera:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.01	0
1	-0.01	0
2	-0.01	0
3	-0.01	0
4	-0.01	0.5
5	-0.01	0.5
6	-0.01	0.5
7	-0.01	0.5
8	-0.01	0.5
9	-0.01	0
10	-0.01	0

Informació

	-1	1
Acceleració	Accel.	Frenada
Direcció	Esquerra	Dreta

**EXECUTAR SIMULACIÓ**

**DADES CORRECTES!**

Figura 66: Entrada variables cas 3.1

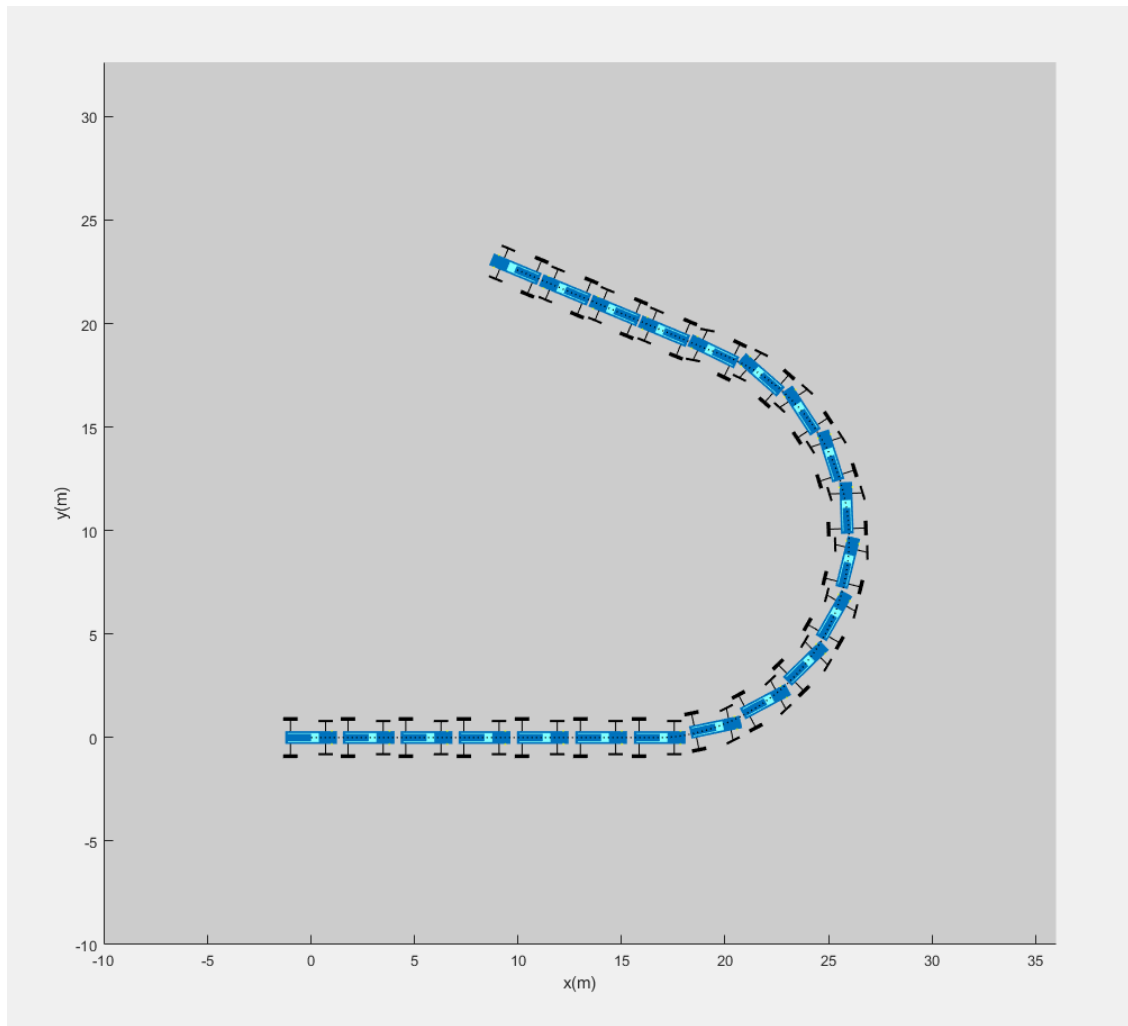


Figura 67: Trajectoria cas 3.1

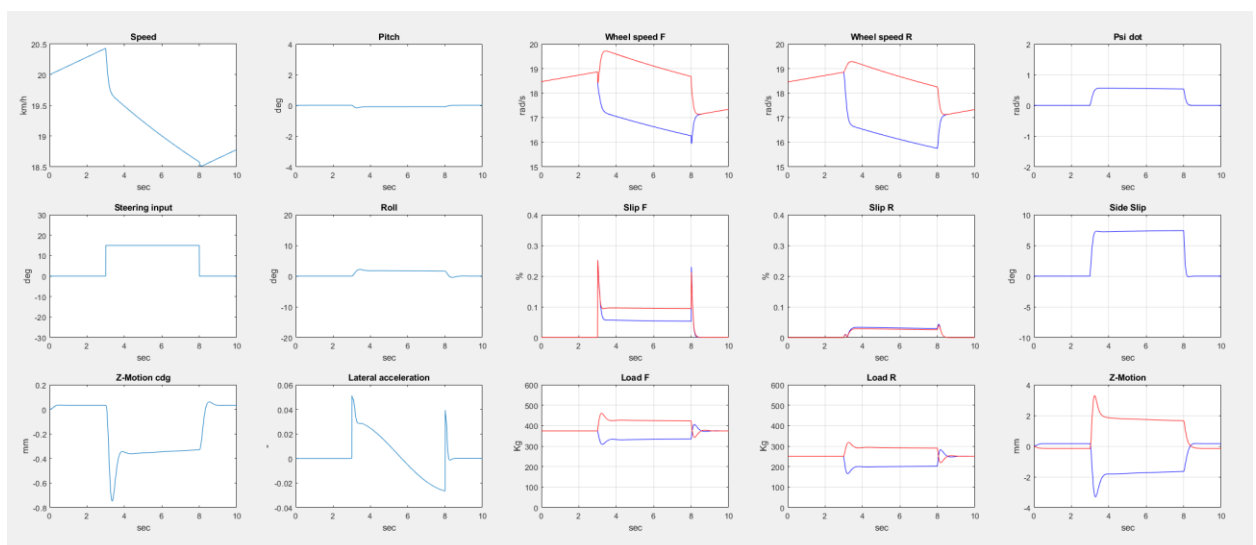


Figura 68: Resultats gràfics cas 3.1

- Vehicle amb tracció posterior:

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.01	0
1	-0.01	0
2	-0.01	0
3	-0.01	0
4	-0.01	0.5
5	-0.01	0.5
6	-0.01	0.5
7	-0.01	0.5
8	-0.01	0.5
9	-0.01	0
10	-0.01	0

Informació

	-1	1
Acceleració	Accel.	Frenada
Direcció	Esquerra	Dreta

**EXECUTAR SIMULACIÓ**

**DADES CORRECTES!**

Figura 69: Entrada variables cas 3.2

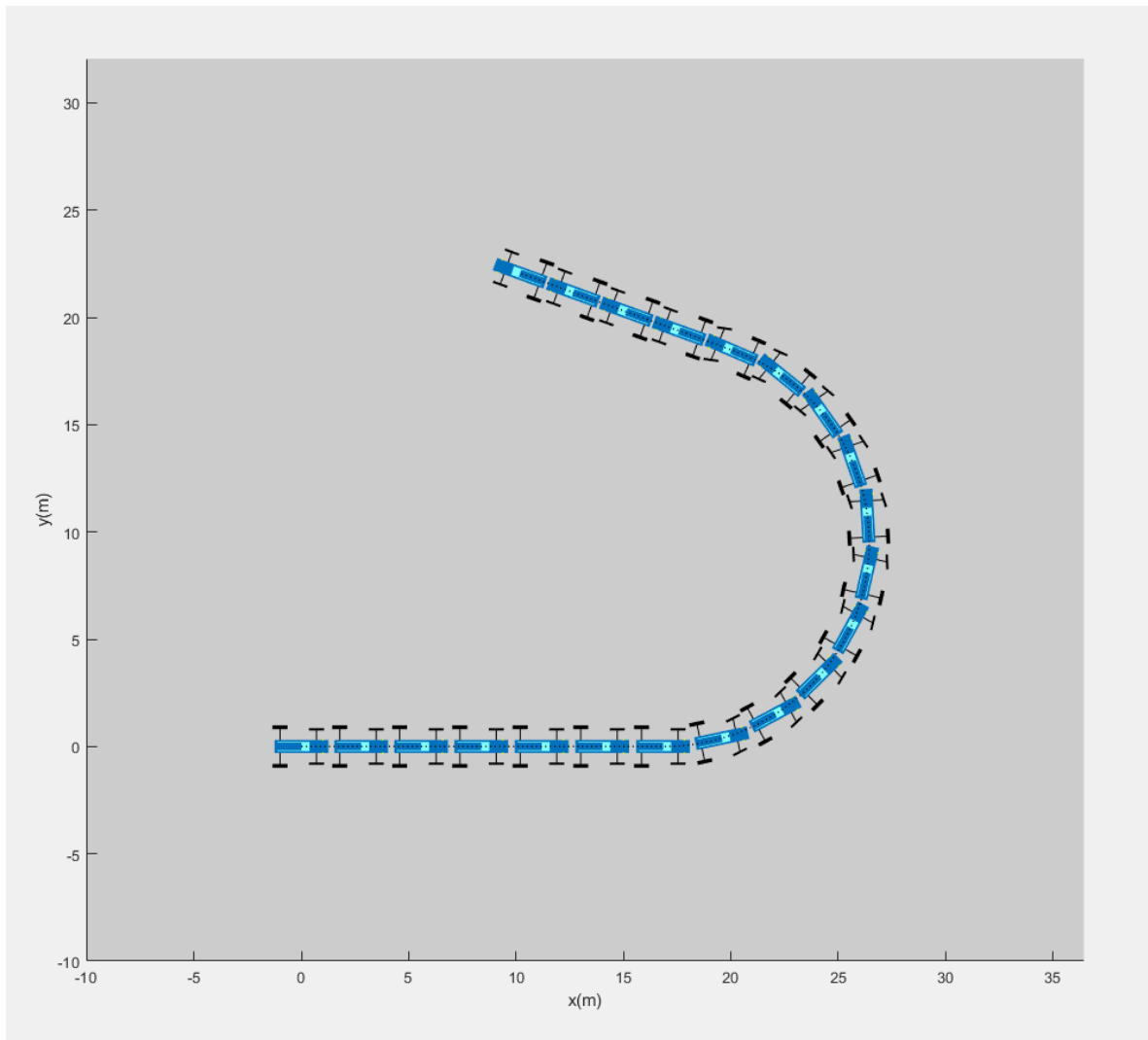


Figura 70: Trajectoria cas 3.2

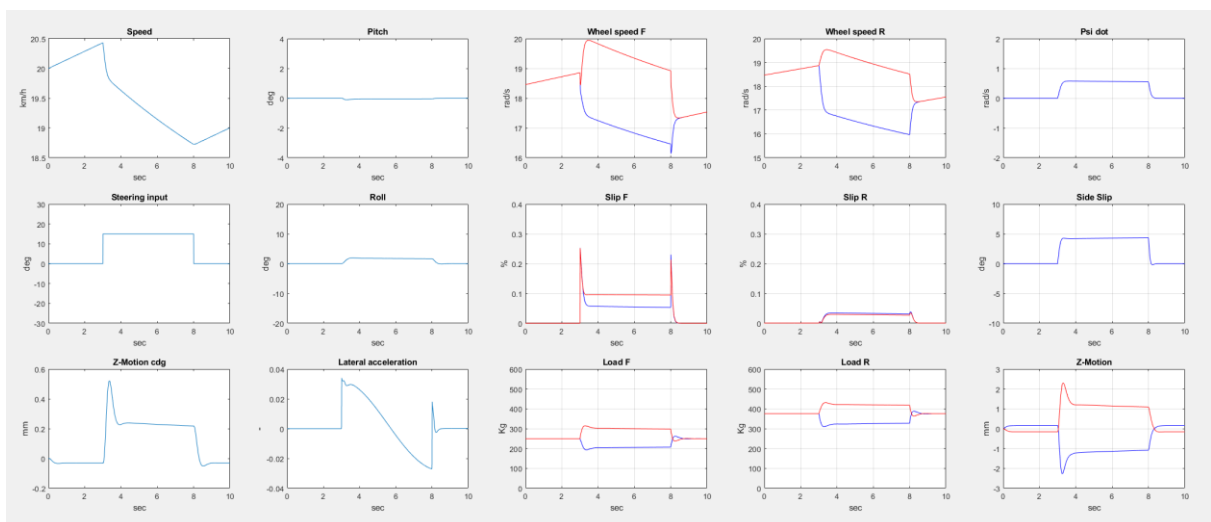


Figura 71: Resultats gràfics cas 3.2

### - Comparativa entre les dues configuracions:

Per aquest tercer cas i després d'haver executat les simulacions per a totes dues configuracions de tracció podem analitzar aquells paràmetres que difereixen en cada una d'aquestes. En aquesta simulació ja intervenen trajectòries no rectilínies, pel que ja prendran importància altres paràmetres que no s'han comentat en els casos anteriors i a més veurem grans diferències entre els comportaments de les rodes d'un costat o de l'altre depenent si es troben al costat interior o exterior de la corba de trajectòria.

En aquest cas, la velocitat del vehicle la qual intentàvem mantenir constant a 20 quilometres per hora prement el pedal de l'accelerador un 1 per cent, podem veure que tenia una lleugera tendència a augmentar fins al moment d'iniciar el gir, on en ambdues direccions baixa en picat durant el primer instant per després seguir baixant de forma més progressiva fins al final del gir. El més destacable però és que la velocitat en el seu punt més baix, és a dir al final del gir, és lleugerament més elevada en el model amb tracció posterior.

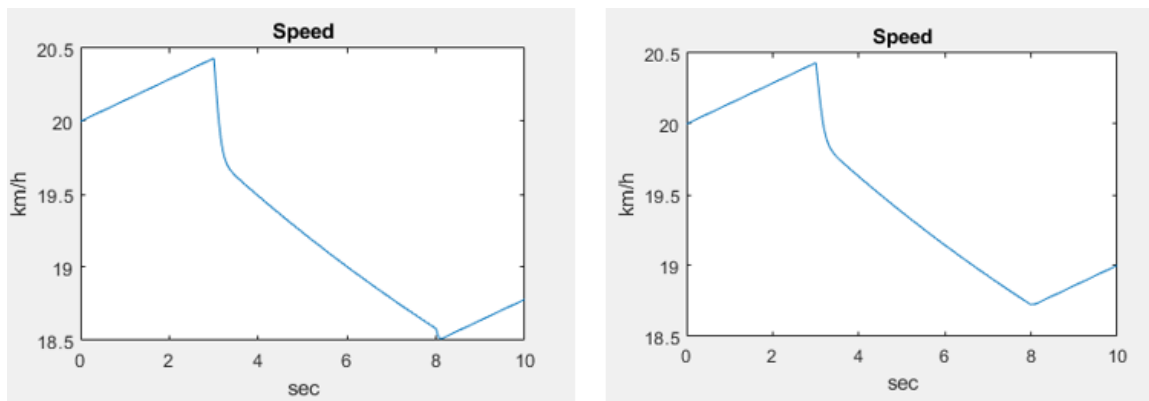


Figura 72: Gràfiques velocitat cas 3

Igual que en els anteriors casos, el moviment del centre de gravetat entre una configuració de tracció i l'altre es reflecteix en gràfiques totalment oposades, en les quals però es poden apreciar els canvis en la conducció, deguts a l'inici i al final del gir.

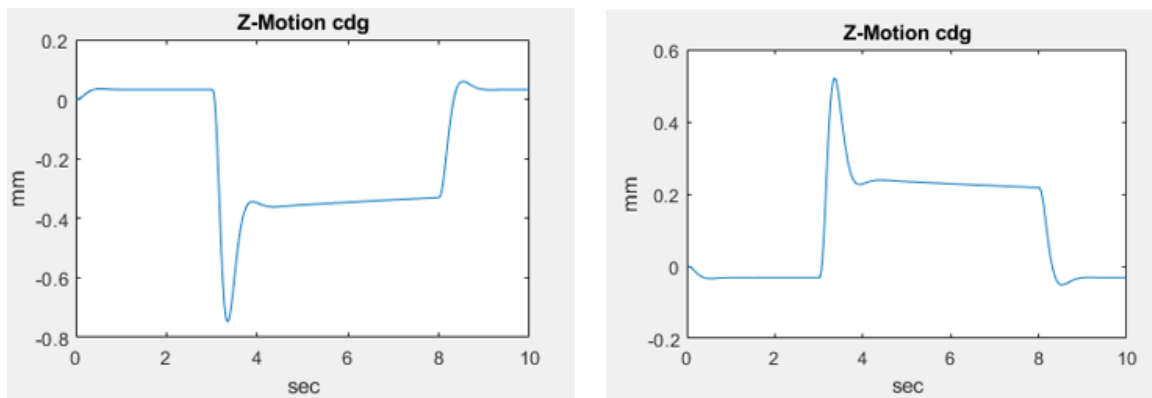


Figura 73: Gràfiques z-motion cas 3

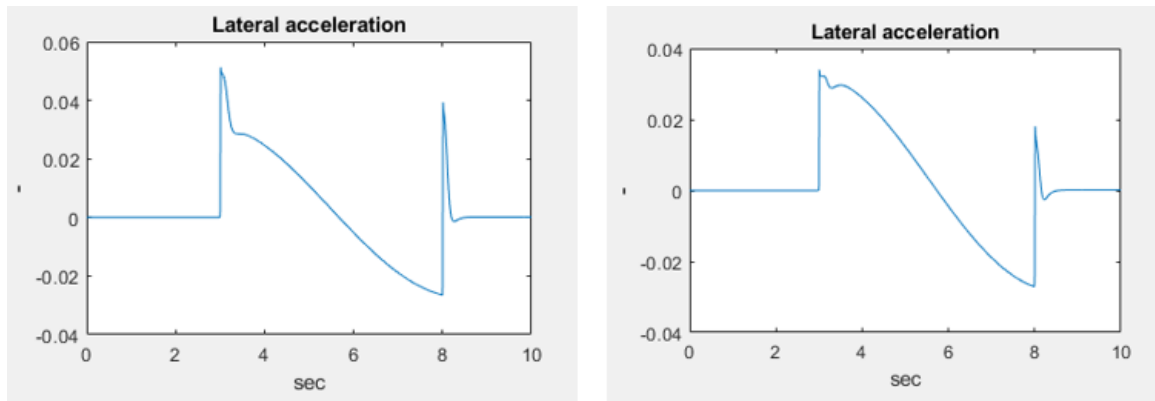


Figura 74: Gràfiques acceleració lateral cas 3

Els paràmetres de velocitat de les rodes presenten una forta desviació entre el costat intern i el costat extern durant el gir. En el vehicle amb tracció posterior les velocitats màximes són més elevades i en ambdós casos, en les rodes davanteres, trobem un pic negatiu en la roda externa a l'inici del gir, i un pic igualment negatiu en el comportament de la roda interior però en aquest cas en l'últim instant del gir. Aquest fenomen és degut al lliscament de les rodes, que a continuació analitzarem i podrem comprovar que el lliscament és molt més elevat en les rodes del tren davanter.

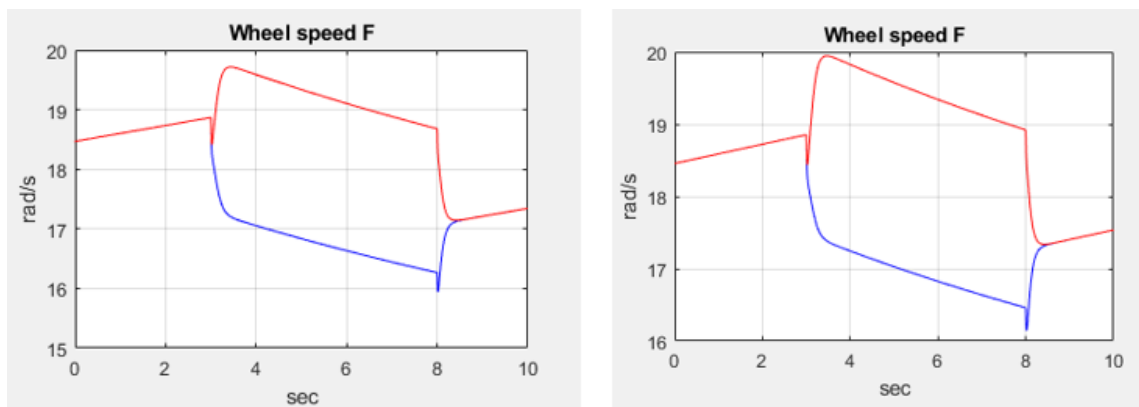


Figura 75: Gràfiques velocitat rodes davanteres cas 3

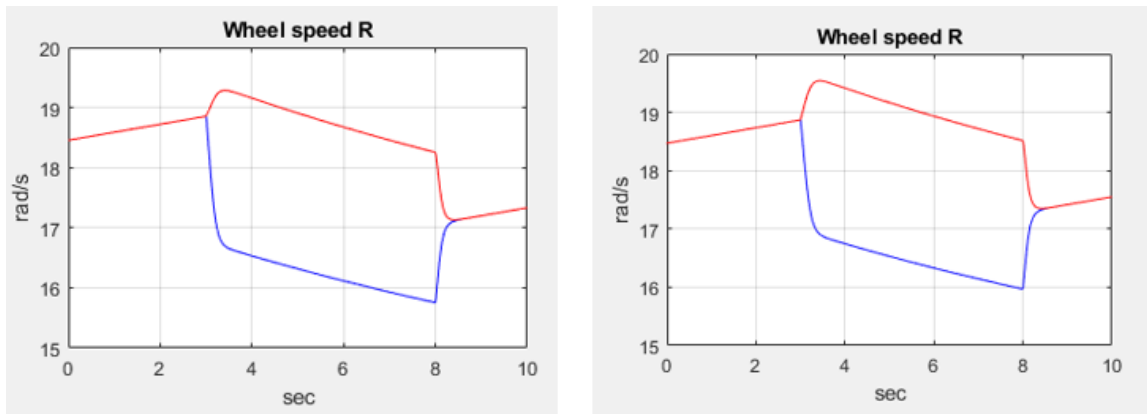


Figura 76: Gràfiques velocitat rodes posteriors cas 3

En aquest tercer cas, el comportament de les rodes a lliscament és força similar entre totes dues configuracions, però aquí sí que veiem contrastos entre les rodes interiors i exteriors de la corba. Particularment en les rodes davanteres podem apreciar valors més alts de lliscament en les rodes del costat exterior de la trajectòria de gir.

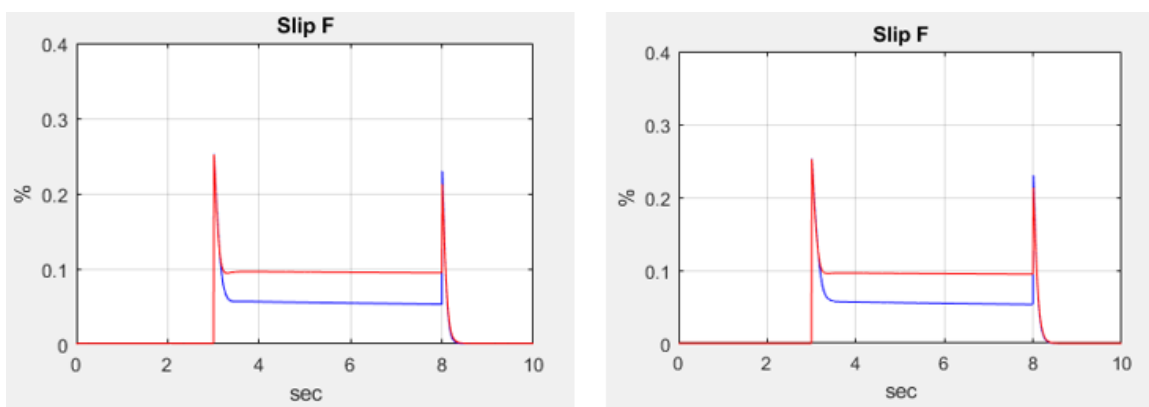


Figura 77: Gràfiques lliscament rodes davanteres cas 3

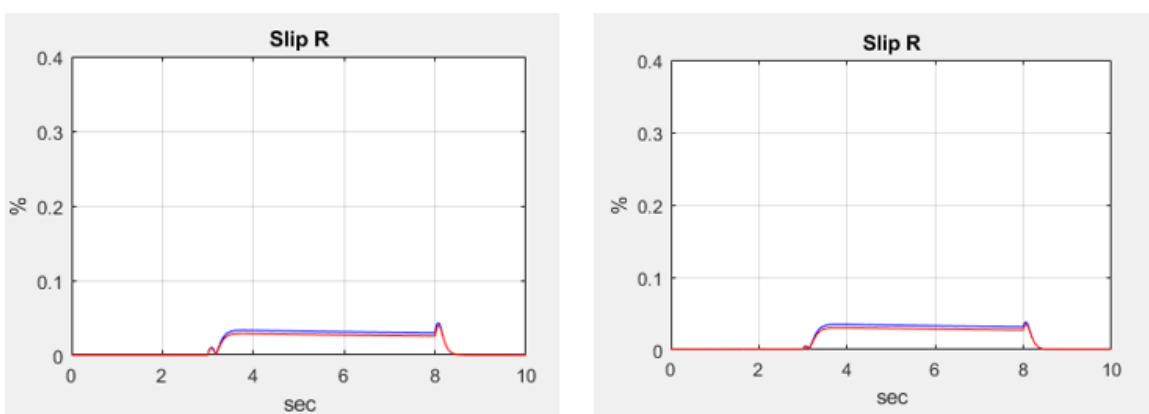


Figura 78: Gràfiques lliscament rodes posteriors cas 3



Pel que fa les càrregues sobre les rodes, el fenomen es similar als casos ja estudiat en el fet de que en les rodes tractores tenim valors més alts de càrrega, tot i així en aquesta simulació podem separar i apreciar les diferències també amb les rodes del costat interior o exterior de la corba, representades en blau i vermell respectivament, i on s'aprecia en tots casos i com era d'esperar valors més elevats de càrrega sobre les rodes del costat extern en els instants en que dura el gir. També es pot apreciar que al final del gir hi ha un intercanvi lleuger de càrregues degut a la recol·locació del vehicle i la força d'inèrcia que aquest portava, fenomen més marcat en el vehicle amb tracció davantera.

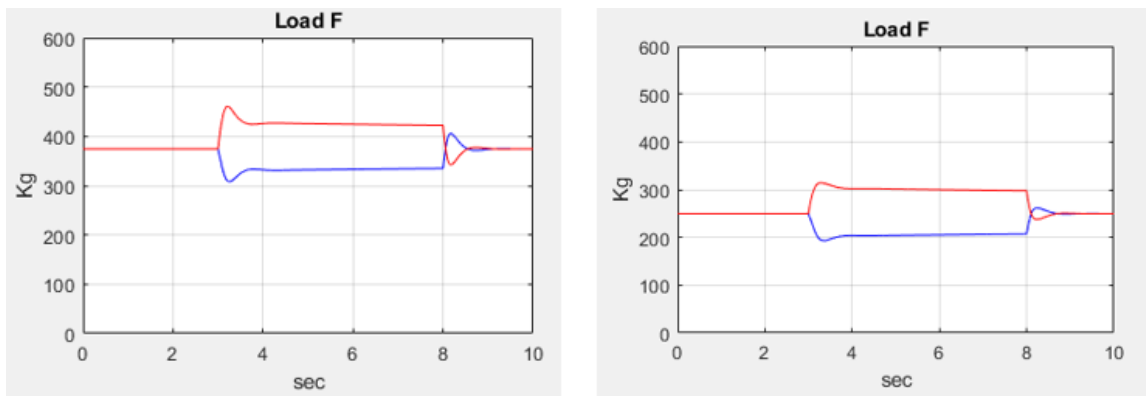


Figura 79: Gràfiques càrregues rodes davanteres cas 3

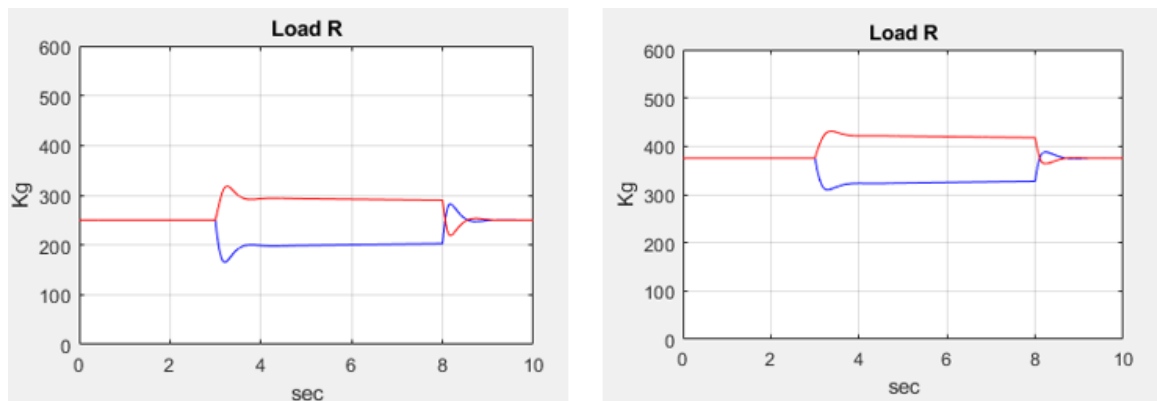


Figura 80: Gràfiques càrregues rodes posteriors cas 3

A continuació podem comparar el lliscament lateral del vehicle en ambdues configuracions. Com era d'esperar tenim lliscament lateral en els instants en que dura el gir, però podem observar que en el vehicle amb tracció davantera aquest valor és més elevat. Això ens quadra amb la representació gràfica de la trajectòria, encara que difícil de visualitzar perquè la diferència és mínima, però en el vehicle amb tracció davantera la trajectòria és lleugerament més oberta degut precisament a aquest lliscament.

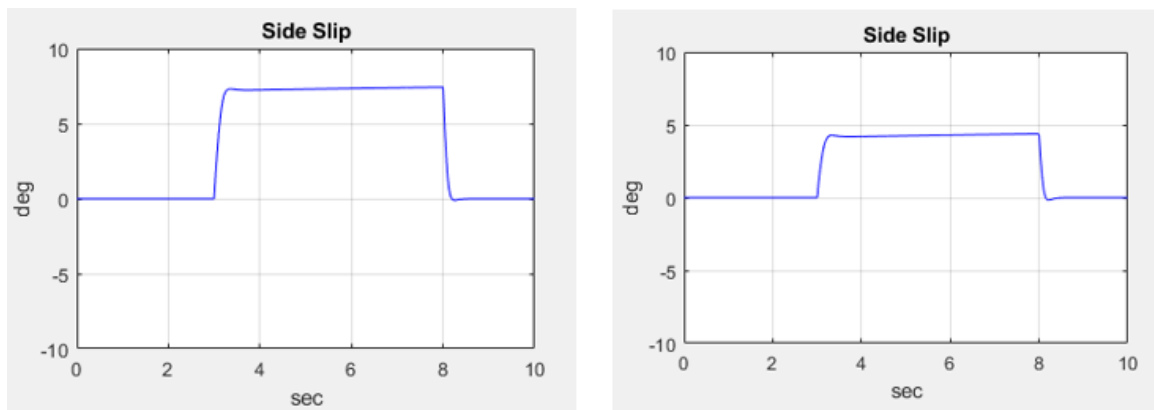


Figura 81: Gràfiques lliscament lateral cas 3

El moviment en l'eix vertical també segueix un patró idèntic però prenen uns valors més elevats en el vehicle amb tracció en l'eix davanter.

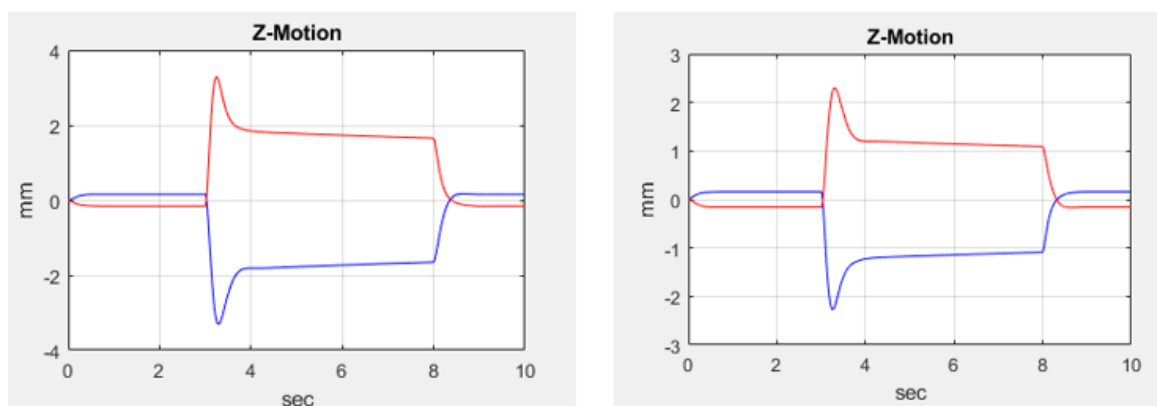


Figura 82: Gràfiques z-motion eix davanter-posterior cas 3

#### 4.4. Cas 4: Corba amb acceleració

En aquest últim cas l'objectiu és el de veure la trajectòria i comportament del vehicle prenent una corba amb un gir en la direcció equivalent a 15 graus, partint d'una velocitat de 30 quilòmetres per hora però en aquest cas amb una acceleració constant equivalent a un 60 per cent de la capacitat del pedal de l'accelerador.

En primer lloc executarem la simulació per al cas amb tracció davantera, seguidament el mateix cas amb tracció posterior i finalment compararem els resultats entre ells:

- Vehicle amb tracció davantera

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.60	0
1	-0.60	0
2	-0.60	0
3	-0.60	0
4	-0.60	0.5
5	-0.60	0.5
6	-0.60	0.5
7	-0.60	0.5
8	-0.60	0.5
9	-0.60	0
10	-0.60	0

Informació

-1
1

Acceleració    Accel.    Frenada

Direcció        Esquerra    Dreta

**EXECUTAR SIMULACIÓ**

**DADES CORRECTES!**

Figura 83: Entrada variables cas 4.1

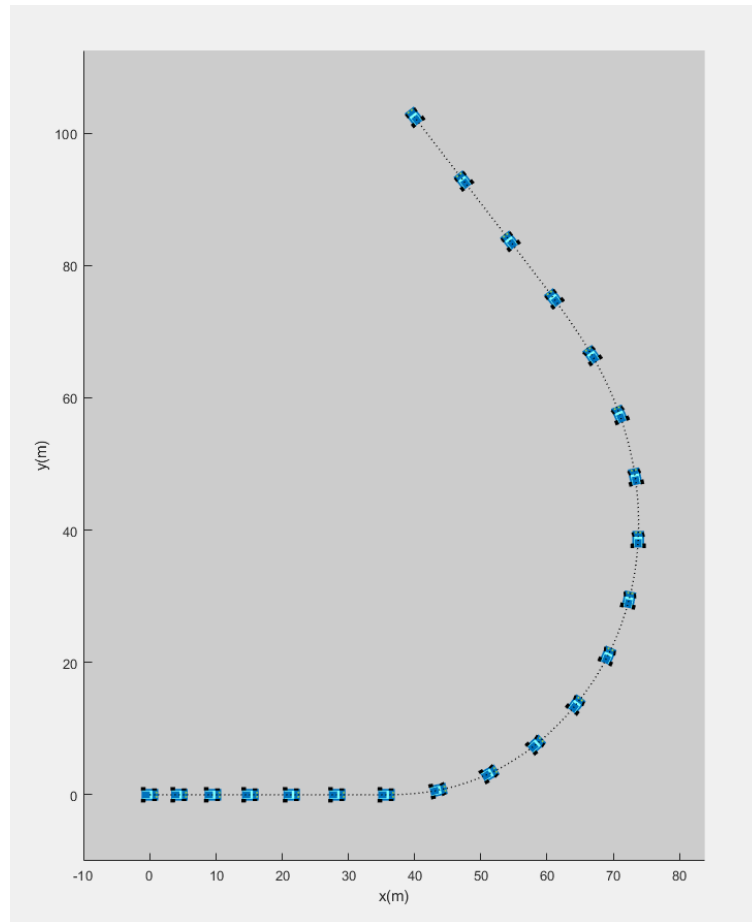


Figura 84: Trajectòria cas 4.1

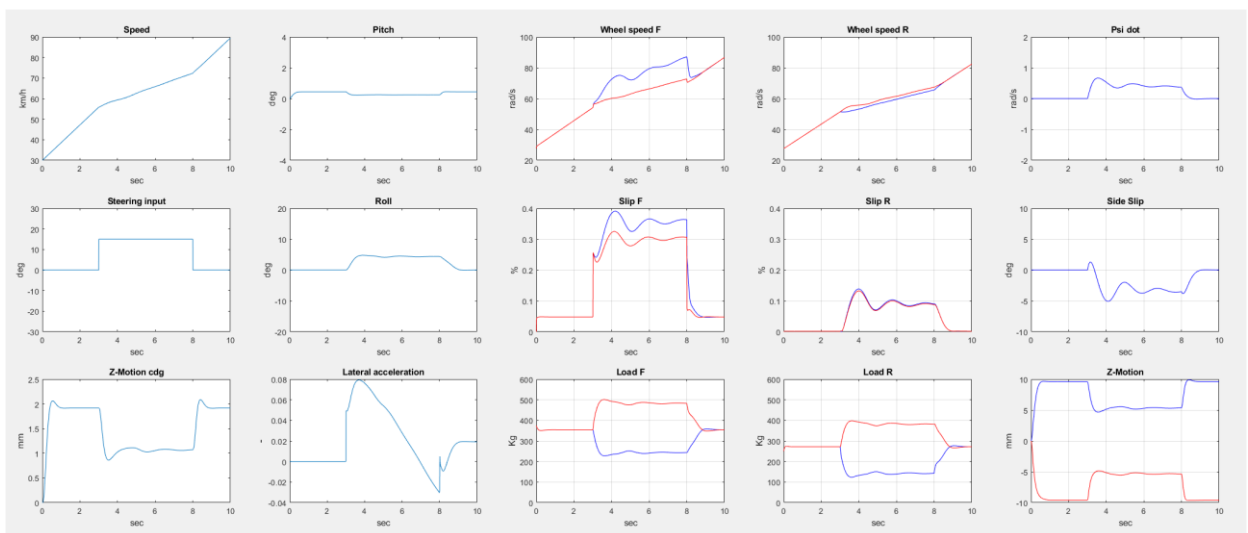


Figura 85: Resultats gràfics cas 4.1

- Vehicle amb tracció posterior

### GEOMETRIA DEL VEHICLE

Massa total:  Kg

Distància de l'eix davanter al cdg:  m

Distància de l'eix posterior al cdg:  m

Radi efectiu dels pneumàtics:  m

### PARÀMETRES DEL MOTOR

Parell davanter:  Nm

Parell posterior:  Nm

### PARÀMETRES DE LA SUSPENSÍO

Kf:  Cf:

Kr:  Cr:

### VELOCITAT INICIAL DEL VEHICLE

Velocitat inicial:  km/h

### ACCIONS DEL CONDUCTOR

Temps	Acceleració	Direcció
0	-0.60	0
1	-0.60	0
2	-0.60	0
3	-0.60	0
4	-0.60	0.5
5	-0.60	0.5
6	-0.60	0.5
7	-0.60	0.5
8	-0.60	0.5
9	-0.60	0
10	-0.60	0

Informació

	<input type="button" value="-1"/>	<input type="button" value="1"/>
<b>Acceleració</b>	Accel.	Frenada
<b>Direcció</b>	Esquerra	Dreta

**EXECUTAR SIMULACIÓ**

**DADES CORRECTES!**

Figura 86: entrada variables cas 4.2

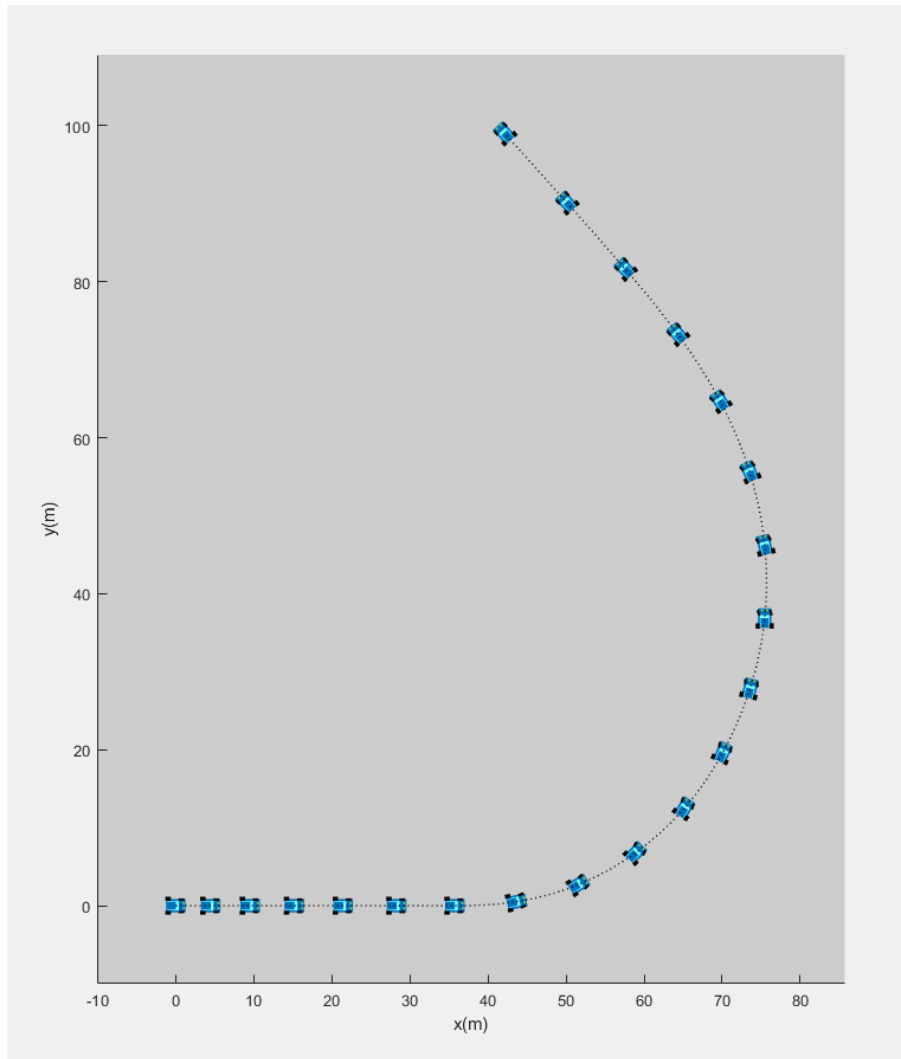


Figura 87: Trajectòria cas 4.2

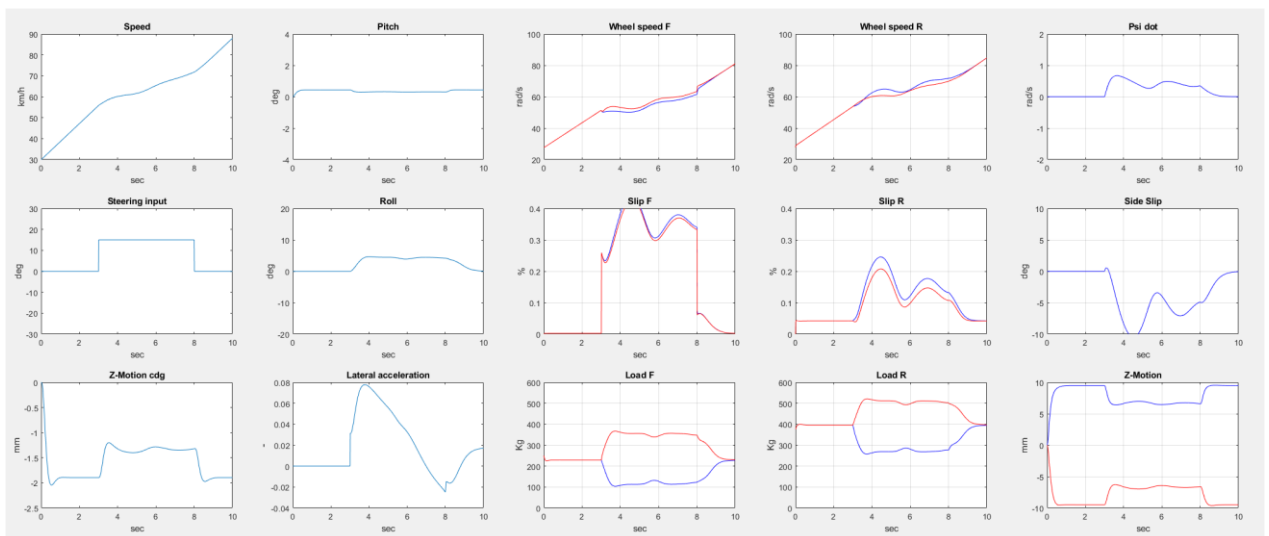


Figura 88: Resultats gràfics cas 4.2

### - Comparativa entre les dues configuracions:

Per aquest últim cas i després d'haver executat les simulacions per a totes dues configuracions de tracció podem analitzar aquells paràmetres que difereixen en cada una d'aquestes. En aquesta simulació, com en l'anterior, intervenen trajectòries no rectilínies, però ara a més a més amb una acceleració durant el gir, fet que provocarà diferències més significatives entre les trajectòries de totes dues configuracions.

Pel que fa a la velocitat del vehicle es pot veure com aquest té una tendència constant a augmentar degut a l'acceleració però durant el gir aquest pendent és menys pronunciat a més de irregular, degut al lliscament dels pneumàtics

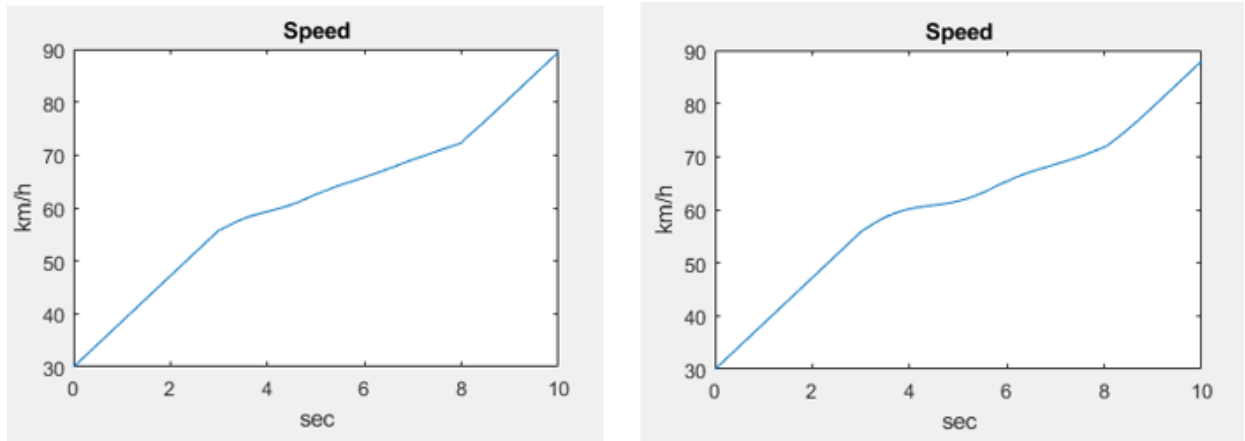


Figura 89: Gràfiques velocitat cas 4

En les velocitats de les rodes davanteres si que s'aprecia una gran diferencia entre el vehicle amb tracció davantera i el de tracció posterior, sobretot degut al comportament irregular de la roda interior de la primera configuració degut a la pèrdua d'adherència. En el vehicle amb tracció posterior veiem que la velocitat també és irregular però molt més propera entre totes dues rodes.

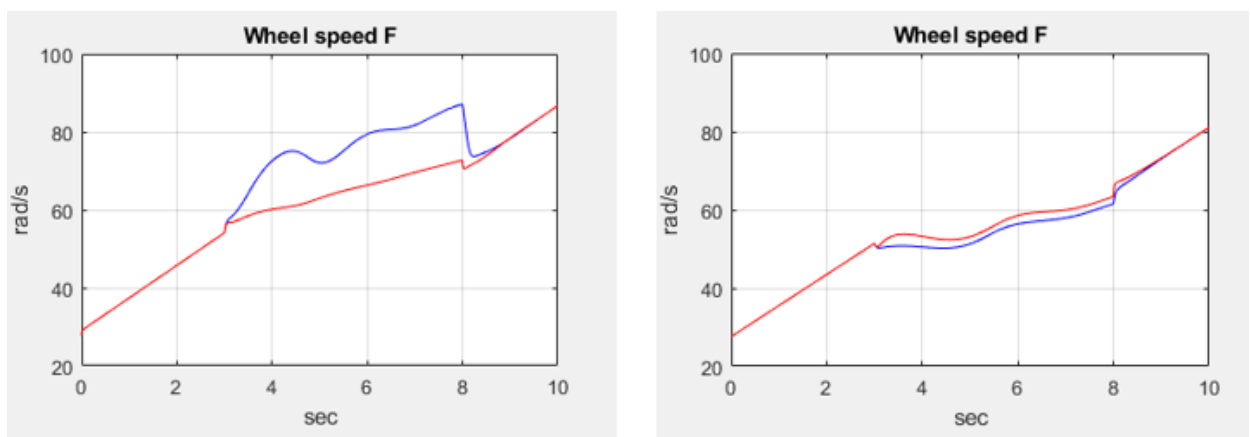


Figura 90: Gràfiques velocitat rodes davanteres cas 4

En les rodes posteriors en canvi, tot i seguir una corba més contínua, el vehicle amb tracció posterior presenta més irregularitats.

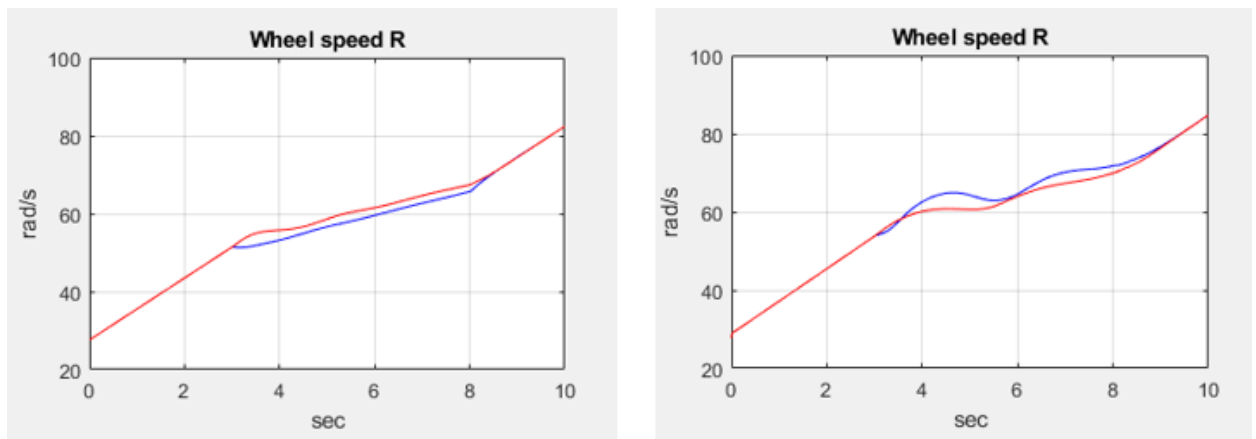


Figura 91: Gràfiques velocitat rodes posteriors cas 4

A continuació podem comprovar que les discrepàncies vistes en les velocitats de les diferents rodes venen directament produïdes pels lliscament corresponents. En general tenim valors més alts de lliscament en les rodes davanteres però on les diferències entre rodes externes és més significativa és en les davanteres en el cas del vehicle amb tracció en l'eix davanter i en les posteriors en el vehicle amb les rodes posteriors com a tractores.

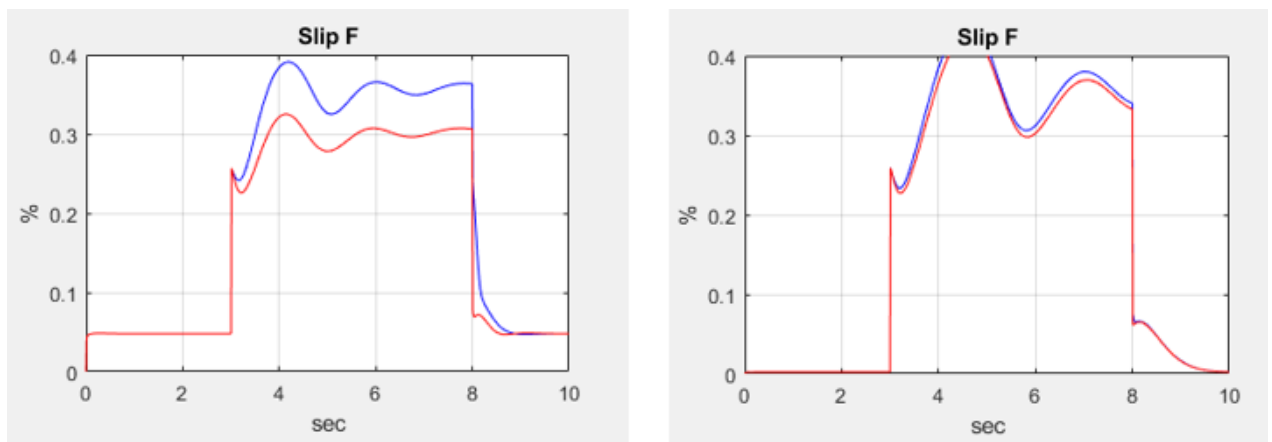


Figura 92: Gràfiques lliscament rodes davanteres cas 4

En valors absoluts podem apreciar també que el vehicle amb tracció en l'eix posterior pateix un lliscament més elevat que el vehicle amb tracció davantera.



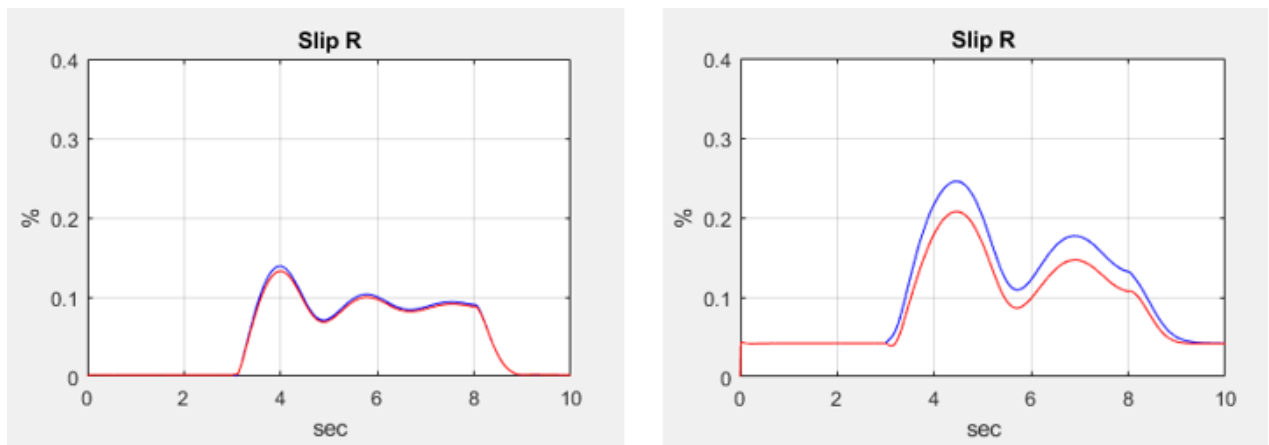


Figura 93: Gràfiques lliscament rodes posteriors cas 4

Pel que fa a les càrregues, aquestes segueixen la tendència esperada, augmentant la càrrega considerablement sobre les rodes exteriors durant el gir

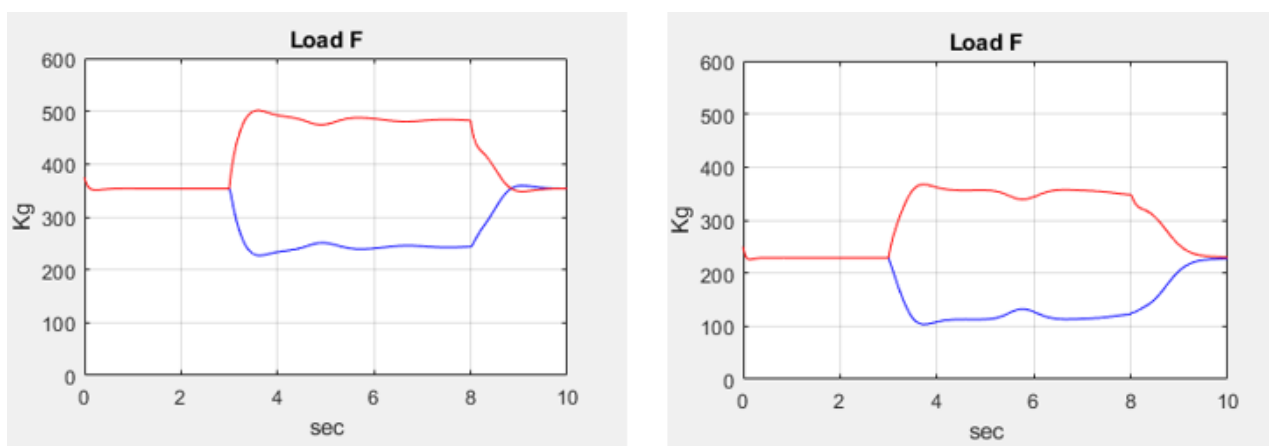


Figura 94: Gràfiques càrregues rodes davanteres cas 4

En les rodes posteriors igual que en les davanteres s'experimenta la diferència de càrregues entre les rodes exteriors i interiors, a més veiem que tant davant com darrera tenim resultats amb una forma força semblant tot i que en la segona configuració sembla que les càrregues s'estabilitzen de forma més progressiva a diferència de la configuració de tracció davantera on el canvi de càrregues és molt més sobtat i amb un petit sobrepic en ambdós sentits, produït per l'efecte dels amortidors.

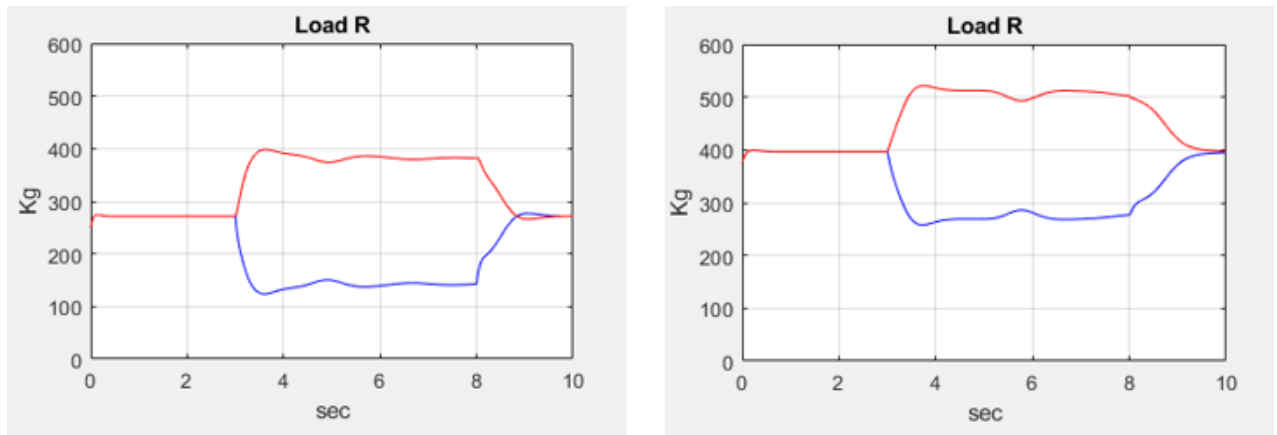


Figura 95: Gràfiques càrregues rodes posteriors cas 4

El lliscament del vehicle i concordant amb el lliscament individual de les rodes veiem que pren valors molt més elevats en el vehicle amb tracció posterior fet que confirma el comentat anteriorment i demostra que el vehicle amb aquesta configuració té tendència ser més inestable als revolts o precisa d'una conducció més experta.

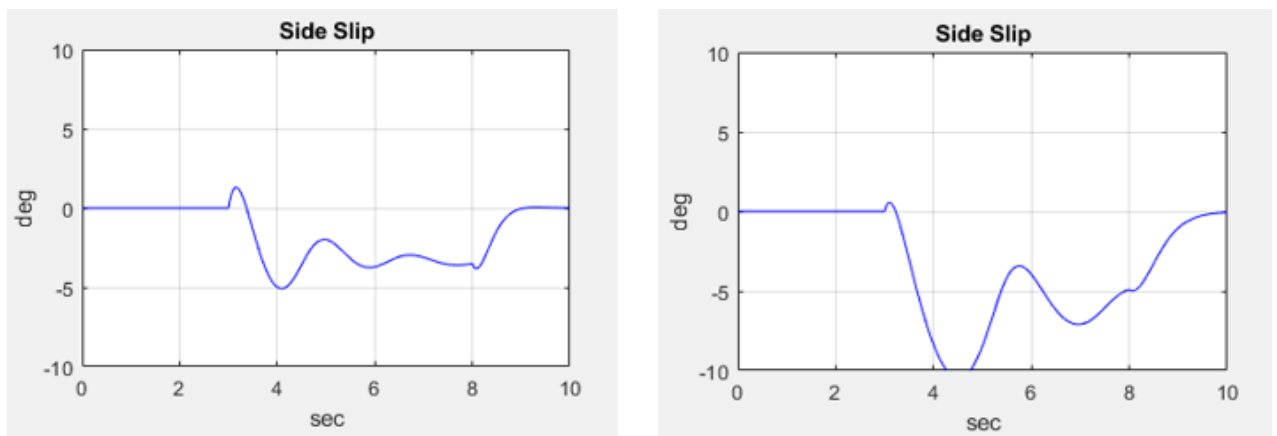


Figura 96: Gràfiques lliscament lateral cas 4

El moviment en l'eix vertical del front davant respecte el posterior veiem que en ambdós casos el valor disminueix durant el gir ja que les càrregues degudes a l'acceleració que provoquen el moviment en l'eix vertical queden minvades per les càrregues laterals degudes a la trajectòria corba.

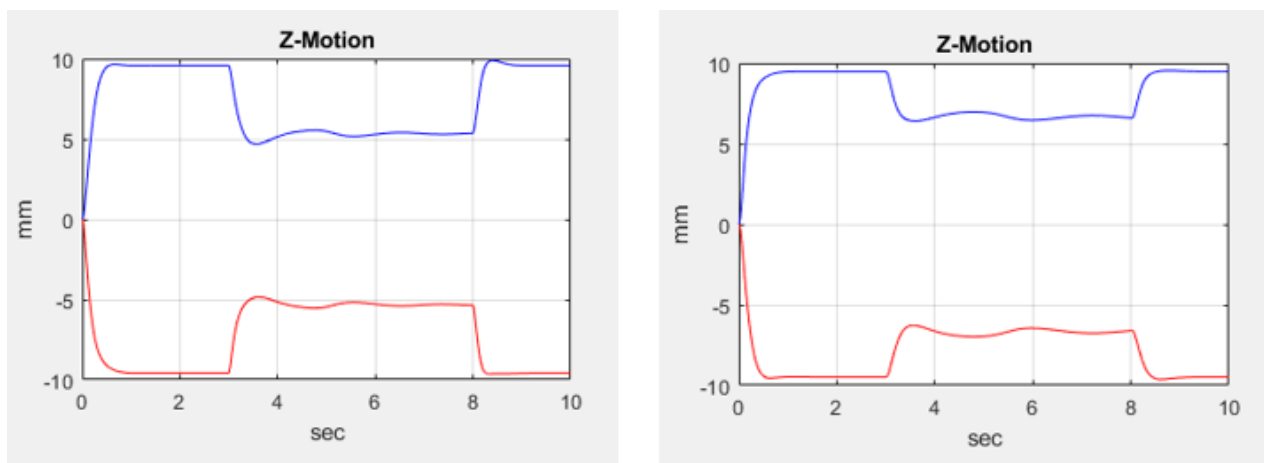


Figura 97: Gràfiques z-motion eix davanter-posterior cas 4

## 5. PRESSUPOST

El pressupost per a aquest estudi està compost bàsicament per les hores de feina destinades i el cost del programari informàtic utilitzat:

<b>Cost d'enginyeria</b>			
<b>Concepte</b>	<b>Preu (€/h)</b>	<b>Hores</b>	<b>Import (€)</b>
Disseny i programació	25	300	7.500,00 €
Simulació	25	50	1.250,00 €
<b>Total</b>			<b>8.750,00 €</b>

<b>Cost d'amortitzacions</b>			
<b>Concepte</b>	<b>N (mesos)</b>	<b>Cost (€/mes)</b>	<b>Import</b>
Llicència software informàtic	6	65	390,00 €
<b>Total</b>			<b>390,00 €</b>

<b>Resum dels pressupostos</b>	
Cost d'enginyeria	8.750,00 €
Cost d'amortitzacions	390,00 €
Pressupost total abans d'impostos	9.140,00 €
IVA 21%	1.919,40 €
<b>Pressupost total</b>	<b>11.059,40 €</b>

El pressupost d'execució per a la modificació i adequació del programari de simulació dinàmica de vehicles i ell seu estudi i anàlisi puja a la quantitat sense IVA de (9.140,00 €):

NOU MIL CENT QUARANTA Euros.

## 6. CONCLUSIONS

En primer lloc, es tenien expectatives de modificació dels models actuals de Simulació de la dinàmica de vehicles amb Matlab molt altes, que a mesura que es va anar aprofundint en el projecte es van anar modificant per enfocar o donar més importància a altres aspectes com és l'anàlisi i l'estudi.

Tot i així, l'objectiu principal del projecte crec que s'ha assolit en un grau força alt, el qual consistia en l'adequació del programari ja existent per tal d'oferir una interfície de treball molt més amable i senzilla per a l'estudi de casos amb l'alumnat en les classes pràctiques de l'assignatura d'enginyeria del transport.

Els punts més crítics eren els de la interfície amb l'usuari, tant l'entrada de dades com en la visualització de resultats i en aquest àmbit s'ha aconseguit millorar considerablement. Per una banda, creant una interfície gràfica per a la introducció de les variables d'entrada, en la qual només ens apareixen els paràmetres que realment es modifiquen constantment per a l'anàlisi de casos d'estudi. És una interfície de molt fàcil ús però no ens priva de poder fer modificacions manualment sobre el codi original per a canviar paràmetres més específics i d'ús no tan habitual.

A més disposa d'un avís de detecció de dades incorrectament entrades abans de l'execució del model de simulació complet, fet que ens pot estalviar temps de càlcul en cas d'errors en l'entrada de variables.

Per altra banda s'han realitzat algunes modificacions en la presentació dels resultats, reduint les finestres de resultats de quatre a dues però conservant la mateixa informació rellevant. Per un costat tenim la finestra que ens mostra la simulació de la trajectòria del vehicle, i per l'altre tenim una gran pantalla on estan concentrats tots els gràfics que ens són realment útils per a l'estudi dels casos i ordenats amb un criteri que en facilita l'anàlisi.

L'altre vessant del projecte és una vessant més teòrica i d'estudi de casos mitjançant els models informàtics. En primer lloc s'ha realitzat un estudi general dels fonaments teòrics més importants que ens permeten crear els models informàtics per a la simulació de la dinàmica en vehicles, per en segon lloc estudiar l'estructura dels actuals models i funcions de *Matlab*, la relació entre ells i relacionar-ho amb la base teòrica comentada.

Finalment un dels punts més interessants ha sigut el d'anàlisi de casos de comportament dinàmic de vehicles utilitzant els models abans descrits. Per una banda s'ha pogut observar tant els punts forts del software com per l'altra, aquells punts on es podria millorar.

De les simulacions realitzades s'han obtingut resultats força bons comparant-los amb els esperats, sobretot en simulacions de moviment rectilini. El punt més important a millorar des del meu punt de vista és la representació gràfica de la trajectòria, que tot i que s'ajusta força a la realitat en la majoria de casos, en altres casos més crítics com és en trajectòries corbes a alta velocitat o en les que intervenen acceleracions i en les quals el vehicle pateix lliscaments, com poden ser subviratges o sobreviratges, aquests no queden representats gràficament amb prou detall en la pantalla de resultat. Tot i així analitzant els gràfics de resultats numèrics, com són les càrregues, velocitats, lliscament, etc. Ens podem fer una bona idea del comportament del vehicle en qualsevol cas.

Finalment una possible implementació que no s'ha tingut capacitat per fer però seria realment interessant seria la d'incorporar una base de dades de vehicles amb les característiques tècniques que afecten als models, per així facilitar encara més l'entorn

d'usuari i simplificar l'estudi de casos, així com implementar una funció que permeti tenir en compte l'inflat dels pneumàtics en el comportament del vehicle.

## **7. RELACIÓ DE DOCUMENTS**

### DOCUMENT NÚM 1: MEMÒRIA

- Annex 1: Instruccions d'ús programa de simulació
- Annex 2: Codis informàtics

## 8. BIBLIOGRAFIA

### Llibres:

- Reza N. Jazar. 2014. *Vehicle Dynamics, Theory and Application*, 2nd Edition. RMIT University, Bundoora, Austràlia.
- Giancarlo Genta. 2006. *Motor Vehicle Dynamics, Modeling and Simulation*. University of Torino, Italy.

### Documentació i manuals:

- Enginyeria del transport, 2018. *Apunts del curs*. Universitat de Girona.
- Web del fabricant de pneumàtics BF Goodrich Tires, 2019. *Guia de conducció*.
- Manuel López Martínez y José Ángel Acosta Rodríguez. 2004. *Manual de Introducción a MATLAB*. Universidad de Sevilla.
- MathWorks. 2018. *Matlab 2018: user manual*.
- M<sup>a</sup> Cristina Casado Fernández. *Manual básico de Matlab*. Servicios informáticos UCM



## ANNEXOS

### ANNEX A: Instruccions d'ús programa de simulació.

Dins la carpeta d'arxius de simulació en format d'arxiu de Matlab, trobarem la relació de models que s'han comentat en la memòria, però de cara a l'usuari final només es treballa amb dos d'ells, seguint una seqüència molt simple:

1. Obrir l'arxiu *interfie\_grafica\_vehicles.m* i executar-lo.
2. Ens apareixerà una nova finestra en forma d'interfície d'entrada de dades. Aquest té unes dades predeterminades per defecte però es modifiquen entrant els valors desitjats per l'usuari.
3. Prémer el botó *Executar simulació* i a continuació ens ha de sortir l'avís de dades correctes en la pròpia pantalla. Si no es així caldrà revisar les dades entrades i tornar a prémer el botó d'executar. Després d'aquest procés es pot tancar o minimitzar la finestra d'entrada de dades.
4. Obrir l'arxiu *FWsim.m* i executar-lo.
5. En uns segons ens apareixeran les pantalles de resultats.
6. Per repetir la simulació o canviar els paràmetres es modificaran en la pantalla d'interfície gràfica i es tornarà a executar el procés.

## ANNEX B: Codis informàtics

### B.1: Codi interfície gràfica

```

function varargout = Interficie_grafica_vehicles(varargin)
% INTERFICIE_GRAFICA_VEHICLES MATLAB code for
Interficie_grafica_vehicles.fig
%     INTERFICIE_GRAFICA_VEHICLES, by itself, creates a new
INTERFICIE_GRAFICA_VEHICLES or raises the existing
%     singleton*.
%
%     H = INTERFICIE_GRAFICA_VEHICLES returns the handle to a new
INTERFICIE_GRAFICA_VEHICLES or the handle to
%     the existing singleton*.
%
%
INTERFICIE_GRAFICA_VEHICLES('CALLBACK',hObject,eventData,handles,...
) calls the local
%     function named CALLBACK in INTERFICIE_GRAFICA_VEHICLES.M with
the given input arguments.
%
%     INTERFICIE_GRAFICA_VEHICLES('Property','Value',...) creates a
new INTERFICIE_GRAFICA_VEHICLES or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before
Interficie_grafica_vehicles_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to
Interficie_grafica_vehicles_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Interficie_grafica_vehicles

% Last Modified by GUIDE v2.5 30-Jul-2019 23:14:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Interficie_grafica_vehicles_OpeningFcn, ...
                  'gui_OutputFcn',  @Interficie_grafica_vehicles_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});

```

```

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Interficie_grafica_vehicles is made
visible.
function Interficie_grafica_vehicles_OpeningFcn(hObject, eventdata,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Interficie_grafica_vehicles
(see VARARGIN)

% Choose default command line output for Interficie_grafica_vehicles
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Interficie_grafica_vehicles wait for user response
(see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Interficie_grafica_vehicles_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function var_parell_Callback(hObject, eventdata, handles)
% hObject    handle to var_parell (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of var_parell as
text
%         str2double(get(hObject,'String')) returns contents of
var_parell as a double

```

```

% --- Executes during object creation, after setting all properties.
function var_parell_CreateFcn(hObject, eventdata, handles)
% hObject    handle to var_parell (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_massa_Callback(hObject, eventdata, handles)
% hObject    handle to var_massa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of var_massa as text
%         str2double(get(hObject,'String')) returns contents of
var_massa as a double

% --- Executes during object creation, after setting all properties.
function var_massa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to var_massa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_dav_Callback(hObject, eventdata, handles)
% hObject    handle to var_dav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of var_dav as text
%         str2double(get(hObject,'String')) returns contents of
var_dav as a double

```

```

% --- Executes during object creation, after setting all properties.
function var_dav_CreateFcn(hObject, eventdata, handles)
% hObject    handle to var_dav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_post_Callback(hObject, eventdata, handles)
% hObject    handle to var_post (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of var_post as text
%         str2double(get(hObject,'String')) returns contents of
var_post as a double

% --- Executes during object creation, after setting all properties.
function var_post_CreateFcn(hObject, eventdata, handles)
% hObject    handle to var_post (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function var_radi_Callback(hObject, eventdata, handles)
% hObject    handle to var_radi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of var_radi as text
%         str2double(get(hObject,'String')) returns contents of
var_radi as a double

% --- Executes during object creation, after setting all properties.
function var_radi_CreateFcn(hObject, eventdata, handles)

```

```
% hObject    handle to var_radi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on slider movement.
function sliderkf_Callback(hObject, eventdata, handles)
% hObject    handle to sliderkf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
```

```
% --- Executes during object creation, after setting all properties.
function sliderkf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sliderkf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
% --- Executes on slider movement.
function sliderkr_Callback(hObject, eventdata, handles)
% hObject    handle to sliderkr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
```

```
% --- Executes during object creation, after setting all properties.
function sliderkr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sliderkr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
% --- Executes on slider movement.
function slidercf_Callback(hObject, eventdata, handles)
% hObject    handle to slidercf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
```

```
% --- Executes during object creation, after setting all properties.
function slidercf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slidercf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
% --- Executes on slider movement.
function slidercr_Callback(hObject, eventdata, handles)
% hObject    handle to slidercr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine
range of slider
```

```
% --- Executes during object creation, after setting all properties.
function slidercr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slidercr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: slider controls usually have a light gray background.
```

```

if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

% --- Executes during object deletion, before destroying properties.
function uitable2_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to uitable2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% --- Executes during object creation, after setting all properties.
function uitable2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uitable2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

function acc0_Callback(hObject, eventdata, handles)
% hObject    handle to acc0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of acc0 as text
%        str2double(get(hObject,'String')) returns contents of acc0
as a double

```

```

% --- Executes during object creation, after setting all properties.
function acc0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function dir0_Callback(hObject, eventdata, handles)
% hObject    handle to dir0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of dir0 as text

```



```
%          str2double(get(hObject,'String')) returns contents of dir0
as a double

% --- Executes during object creation, after setting all properties.
function dir0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function acc1_Callback(hObject, eventdata, handles)
% hObject    handle to acc1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc1 as text
%          str2double(get(hObject,'String')) returns contents of acc1
as a double

% --- Executes during object creation, after setting all properties.
function acc1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dir1_Callback(hObject, eventdata, handles)
% hObject    handle to dir1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir1 as text
%          str2double(get(hObject,'String')) returns contents of dir1
as a double
```

```
% --- Executes during object creation, after setting all properties.
function dir1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function acc2_Callback(hObject, eventdata, handles)
% hObject    handle to acc2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc2 as text
%         str2double(get(hObject,'String')) returns contents of acc2
as a double

% --- Executes during object creation, after setting all properties.
function acc2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dir2_Callback(hObject, eventdata, handles)
% hObject    handle to dir2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir2 as text
%         str2double(get(hObject,'String')) returns contents of dir2
as a double
```

```

% --- Executes during object creation, after setting all properties.
function dir2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function acc3_Callback(hObject, eventdata, handles)
% hObject    handle to acc3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc3 as text
%       str2double(get(hObject,'String')) returns contents of acc3
as a double

% --- Executes during object creation, after setting all properties.
function acc3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dir3_Callback(hObject, eventdata, handles)
% hObject    handle to dir3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir3 as text
%       str2double(get(hObject,'String')) returns contents of dir3
as a double

% --- Executes during object creation, after setting all properties.
function dir3_CreateFcn(hObject, eventdata, handles)

```

```
% hObject    handle to dir3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function acc4_Callback(hObject, eventdata, handles)
% hObject    handle to acc4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of acc4 as text
%        str2double(get(hObject,'String')) returns contents of acc4
as a double
```

```
% --- Executes during object creation, after setting all properties.
function acc4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function dir4_Callback(hObject, eventdata, handles)
% hObject    handle to dir4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of dir4 as text
%        str2double(get(hObject,'String')) returns contents of dir4
as a double
```

```
% --- Executes during object creation, after setting all properties.
function dir4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function acc5_Callback(hObject, eventdata, handles)
% hObject    handle to acc5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of acc5 as text
%       str2double(get(hObject,'String')) returns contents of acc5
as a double
```

```
% --- Executes during object creation, after setting all properties.
function acc5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function dir5_Callback(hObject, eventdata, handles)
% hObject    handle to dir5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of dir5 as text
%       str2double(get(hObject,'String')) returns contents of dir5
as a double
```

```
% --- Executes during object creation, after setting all properties.
function dir5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function acc6_Callback(hObject, eventdata, handles)
% hObject     handle to acc6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of acc6 as text
%     str2double(get(hObject,'String')) returns contents of acc6
as a double
```

```
% --- Executes during object creation, after setting all properties.
function acc6_CreateFcn(hObject, eventdata, handles)
% hObject     handle to acc6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function dir6_Callback(hObject, eventdata, handles)
% hObject     handle to dir6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of dir6 as text
%     str2double(get(hObject,'String')) returns contents of dir6
as a double
```

```
% --- Executes during object creation, after setting all properties.
function dir6_CreateFcn(hObject, eventdata, handles)
% hObject     handle to dir6 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function acc7_Callback(hObject, eventdata, handles)
% hObject handle to acc7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc7 as text
% str2double(get(hObject,'String')) returns contents of acc7
as a double

% --- Executes during object creation, after setting all properties.
function acc7_CreateFcn(hObject, eventdata, handles)
% hObject handle to acc7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dir7_Callback(hObject, eventdata, handles)
% hObject handle to dir7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir7 as text
% str2double(get(hObject,'String')) returns contents of dir7
as a double

% --- Executes during object creation, after setting all properties.
function dir7_CreateFcn(hObject, eventdata, handles)
% hObject handle to dir7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function acc8_Callback(hObject, eventdata, handles)
% hObject      handle to acc8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc8 as text
%          str2double(get(hObject,'String')) returns contents of acc8
as a double

```

```

% --- Executes during object creation, after setting all properties.
function acc8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to acc8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function dir8_Callback(hObject, eventdata, handles)
% hObject      handle to dir8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir8 as text
%          str2double(get(hObject,'String')) returns contents of dir8
as a double

```

```

% --- Executes during object creation, after setting all properties.
function dir8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to dir8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```



```

    set(hObject,'BackgroundColor','white');
end

```

```

function acc9_Callback(hObject, eventdata, handles)
% hObject      handle to acc9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc9 as text
%         str2double(get(hObject,'String')) returns contents of acc9
as a double

```

```

% --- Executes during object creation, after setting all properties.
function acc9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to acc9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function dir9_Callback(hObject, eventdata, handles)
% hObject      handle to dir9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir9 as text
%         str2double(get(hObject,'String')) returns contents of dir9
as a double

```

```

% --- Executes during object creation, after setting all properties.
function dir9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to dir9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function acc10_Callback(hObject, eventdata, handles)
% hObject    handle to acc10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of acc10 as text
%        str2double(get(hObject,'String')) returns contents of acc10
as a double

% --- Executes during object creation, after setting all properties.
function acc10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to acc10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function dir10_Callback(hObject, eventdata, handles)
% hObject    handle to dir10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of dir10 as text
%        str2double(get(hObject,'String')) returns contents of dir10
as a double

% --- Executes during object creation, after setting all properties.
function dir10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dir10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
text2.
function text2_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function edit28_Callback(hObject, eventdata, handles)
% hObject    handle to edit28 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit28 as text
%        str2double(get(hObject,'String')) returns contents of
edit28 as a double

% --- Executes during object creation, after setting all properties.
function edit28_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit28 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit29_Callback(hObject, eventdata, handles)
% hObject    handle to edit29 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit29 as text
%        str2double(get(hObject,'String')) returns contents of
edit29 as a double

% --- Executes during object creation, after setting all properties.
function edit29_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit29 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit30_Callback(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit30 as text
%         str2double(get(hObject,'String')) returns contents of
edit30 as a double

% --- Executes during object creation, after setting all properties.
function edit30_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit31_Callback(hObject, eventdata, handles)
% hObject    handle to edit31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit31 as text
%         str2double(get(hObject,'String')) returns contents of
edit31 as a double

% --- Executes during object creation, after setting all properties.
function edit31_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on mouse press over axes background.
function pantalla_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to pantalla (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in imatge.
function imatge_Callback(hObject, eventdata, handles)
% hObject    handle to imatge (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in botoexecutar.
function botoexecutar_Callback(hObject, eventdata, handles)
% hObject    handle to botoexecutar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
massa = str2double( get(handles.massa, 'string'));
cdgdav = str2double( get(handles.cdgdav, 'string'));
cdgpost = str2double( get(handles.cdgpost, 'string'));
radi = str2double( get(handles.radi, 'string'));
%traccio = str2double( get(handles.traccio, 'string'));
parell_dav = str2double( get(handles.parell_dav, 'string'));
parell_post = str2double( get(handles.parell_post, 'string'));
kf = str2double( get(handles.kf, 'string'));
kr = str2double( get(handles.kr, 'string'));
cf = str2double( get(handles.cf, 'string'));
cr = str2double( get(handles.cr, 'string'));
acc0 = str2double( get(handles.acc0, 'string'));
acc1 = str2double( get(handles.acc1, 'string'));
acc2 = str2double( get(handles.acc2, 'string'));
acc3 = str2double( get(handles.acc3, 'string'));
acc4 = str2double( get(handles.acc4, 'string'));
acc5 = str2double( get(handles.acc5, 'string'));
acc6 = str2double( get(handles.acc6, 'string'));
acc7 = str2double( get(handles.acc7, 'string'));
acc8 = str2double( get(handles.acc8, 'string'));
acc9 = str2double( get(handles.acc9, 'string'));
acc10 = str2double( get(handles.acc10, 'string'));
dir0 = str2double( get(handles.dir0, 'string'));
dir1 = str2double( get(handles.dir1, 'string'));
dir2 = str2double( get(handles.dir2, 'string'));
dir3 = str2double( get(handles.dir3, 'string'));
dir4 = str2double( get(handles.dir4, 'string'));
dir5 = str2double( get(handles.dir5, 'string'));
dir6 = str2double( get(handles.dir6, 'string'));
dir7 = str2double( get(handles.dir7, 'string'));

```

```

dir8 = str2double( get(handles.dir8, 'string'));
dir9 = str2double( get(handles.dir9, 'string'));
dir10 = str2double( get(handles.dir10, 'string'));
velinicial = str2double( get(handles.velinicial, 'string'));

resultat = str2double( get(handles.resultat, 'string'));
resultat = 0;

fileID = fopen('exp.txt','w');
fprintf(fileID,'%d\r\n',massa,cdgdav,cdgpost,radi,parell_dav,parell_
post,kf,kr,cf,cr,acc0,acc1,acc2,acc3,acc4,acc5,acc6,acc7,acc8,acc9,acc10,dir0,dir1,dir2,dir3,dir4,dir5,dir6,dir7,dir8,dir9,dir10,velinicial);
fclose(fileID);

set ( handles.resultat, 'string', num2str(resultat));

if acc0<=1 && acc0>-1 && dir0<=1 && dir0>-1 && acc1<=1 && acc1>-1 &&
dir1<=1 && dir1>-1 &&acc2<=1 && acc2>-1 && dir2<=1 && dir2>-1 &&
acc3<=1 && acc3>-1 && dir3<=1 && dir3>-1 && acc4<=1 && acc4>-1 &&
dir4<=1 && dir4>-1 && acc5<=1 && acc5>-1 && dir5<=1 && dir5>-1 &&
acc6<=1 && acc6>-1 && dir6<=1 && dir6>-1 && acc7<=1 && acc7>-1 &&
dir7<=1 && dir7>-1 && acc8<=1 && acc8>-1 && dir8<=1 && dir8>-1 &&
acc9<=1 && acc9>-1 && dir9<=1 && dir9>-1 && acc10<=1 && acc10>-1 &&
dir10<=1 && dir10>-1
im=imread('ok.jpg');

else
im=imread('ko.jpg');

end
imshow(im);

function kf_Callback(hObject, eventdata, handles)
% hObject    handle to kf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kf as text
%         str2double(get(hObject,'String')) returns contents of kf as
a double

% --- Executes during object creation, after setting all properties.
function kf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to kf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function kr_Callback(hObject, eventdata, handles)
% hObject    handle to kr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kr as text
%         str2double(get(hObject,'String')) returns contents of kr as
a double

% --- Executes during object creation, after setting all properties.
function kr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to kr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function cf_Callback(hObject, eventdata, handles)
% hObject    handle to cf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cf as text
%         str2double(get(hObject,'String')) returns contents of cf as
a double

% --- Executes during object creation, after setting all properties.
function cf_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cf (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

    set(hObject,'BackgroundColor','white');
end

```

```

function cr_Callback(hObject, eventdata, handles)
% hObject    handle to cr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cr as text
%        str2double(get(hObject,'String')) returns contents of cr as
a double

```

```

% --- Executes during object creation, after setting all properties.
function cr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on selection change in traccio.
function traccio_Callback(hObject, eventdata, handles)
% hObject    handle to traccio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns traccio
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
traccio

```

```

% --- Executes during object creation, after setting all properties.
function traccio_CreateFcn(hObject, eventdata, handles)
% hObject    handle to traccio (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```



```

    set(hObject,'BackgroundColor','white');
end

```

```

function parell_Callback(hObject, eventdata, handles)
% hObject    handle to parell (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parell as text
%         str2double(get(hObject,'String')) returns contents of
parell as a double

```

```

% --- Executes during object creation, after setting all properties.
function parell_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parell (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function massa_Callback(hObject, eventdata, handles)
% hObject    handle to massa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of massa as text
%         str2double(get(hObject,'String')) returns contents of massa
as a double

```

```

% --- Executes during object creation, after setting all properties.
function massa_CreateFcn(hObject, eventdata, handles)
% hObject    handle to massa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function cdgdav_Callback(hObject, eventdata, handles)
% hObject    handle to cdgdav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cdgdav as text
%         str2double(get(hObject,'String')) returns contents of
cdgdav as a double

% --- Executes during object creation, after setting all properties.
function cdgdav_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cdgdav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function cdgpost_Callback(hObject, eventdata, handles)
% hObject    handle to cdgpost (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of cdgpost as text
%         str2double(get(hObject,'String')) returns contents of
cdgpost as a double

% --- Executes during object creation, after setting all properties.
function cdgpost_CreateFcn(hObject, eventdata, handles)
% hObject    handle to cdgpost (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function radi_Callback(hObject, eventdata, handles)
% hObject    handle to radi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of radi as text
%        str2double(get(hObject,'String')) returns contents of radi
as a double
```

```
% --- Executes during object creation, after setting all properties.
function radi_CreateFcn(hObject, eventdata, handles)
% hObject    handle to radi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function parell_dav_Callback(hObject, eventdata, handles)
% hObject    handle to parell_dav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parell_dav as
text
%        str2double(get(hObject,'String')) returns contents of
parell_dav as a double
```

```
% --- Executes during object creation, after setting all properties.
function parell_dav_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parell_dav (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```

function parell_post_Callback(hObject, eventdata, handles)
% hObject    handle to parell_post (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of parell_post as
text
%          str2double(get(hObject,'String')) returns contents of
parell_post as a double

% --- Executes during object creation, after setting all properties.
function parell_post_CreateFcn(hObject, eventdata, handles)
% hObject    handle to parell_post (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function velinicial_Callback(hObject, eventdata, handles)
% hObject    handle to velinicial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of velinicial as
text
%          str2double(get(hObject,'String')) returns contents of
velinicial as a double

% --- Executes during object creation, after setting all properties.
function velinicial_CreateFcn(hObject, eventdata, handles)
% hObject    handle to velinicial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

**B.2: Codi funció de simulació**

```

clear all
close all
clc

% DEFINICIÓ VARIABLES
global m lf lr tf tr Jp Jx Jy W B C D Jwf Rwf Jwr Rwr h Kf Kr Cf Cr
maxsteer toef toer
global BrkConstF BrkConstR AccConstF AccConstR

% Importació variables

type exp.txt;
fileID = fopen('exp.txt','r');
formatSpec = '%f';
sizeA = [Inf 1];
var_entrada = fscanf(fileID,formatSpec)
fclose(fileID);

% GEOMETRIA
m = var_entrada(1);%mass in kg
lf = var_entrada(2);
lr = var_entrada(3);
tf = 1.4; tr = 1.4;
Jp = 2741.9; Jx = 1000; Jy = 1000; W = m*9.81;
Jwf = 1.8; Jwr = 1.8;
Rwf = var_entrada(4);
Rwr = var_entrada(4);
B = 7; C = 1.6; D = 0.9; %tyre parameters
h = 0.35; %CoG height
toef = 0* (2*pi/360); %toe-in, units in rad
toer = 0* (2*pi/360);

% PARÀMETRES DE SUSPENSIO

Kf = var_entrada(7);
Kr = var_entrada(8);
Cf = var_entrada(9);
Cr = var_entrada(10);

% PARÀMETRES DEL MOTOR / FRENS
%driving & engine constants

maxsteer = 30; %deg
BrkConstF = 500; %mN
BrkConstR = 400;
AccConstF = var_entrada(5);
AccConstR = var_entrada(6);

% DADES INICIALITZACIÓ
% POSICIÓ INICIAL DEL CDG

```

```

x0 = 0;
y0 = 0;
z0 = h;

% VELOCITAT INICIAL
xdot0 = var_entrada(33)/3.6;
ydot0 = 0;
zdot0 = 0;

%ANGLES INICIALS DEL COTXE

psi0 = 0;
psidot0 = 0;
theta0 = 0;
thetadot0 = 0;
phi0 = 0;
phidot0 = 0;
omegaFL0 = sqrt(xdot0^2 + ydot0^2)/Rwf;
omegaFR0 = sqrt(xdot0^2 + ydot0^2)/Rwf;
omegaRL0 = sqrt(xdot0^2 + ydot0^2)/Rwr;
omegaRR0 = sqrt(xdot0^2 + ydot0^2)/Rwr;

% SIMULACIÓ

e0 = [x0 xdot0 y0 ydot0 z0 zdot0 psi0 psidot0 theta0 thetadot0 phi0
phidot0 omegaFL0 omegaFR0 omegaRL0 omegaRR0]';
tspan = [0:0.01:10];
options = [];
[t,e] = ode45('FWeom',tspan,e0,options);

%=====
%=====
%=====

x = e(:,1);
xdot = e(:,2);
y = e(:,3);
ydot = e(:,4);
z = e(:,5);
zdot = e(:,6);
psi = e(:,7);
psidot = e(:,8);
theta = e(:,9);
thetadot = e(:,10);
phi = e(:,11);
phidot = e(:,12);
omegaFL = e(:,13);
omegaFR = e(:,14);
omegaRL = e(:,15);
omegaRR = e(:,16);

for i = 1:length(t)
    delta = steer_fun(t(i));

```

```

    deltaFL(i) = delta(1);
    deltaFR(i) = delta(2);
    deltaRL(i) = delta(3);
    deltaRR(i) = delta(4);
    clear delta
    torque = accbrk_fun(t(i), omegaFL(i), omegaFR(i), omegaRL(i),
omegaRR(i));
    TFL(i) = torque(1);
    TFR(i) = torque(2);
    TRL(i) = torque(3);
    TRR(i) = torque(4);
    clear torque

    V(i) = sqrt(xdot(i)^2 + ydot(i)^2);

    rho(i) = atan2(ydot(i), xdot(i));
    beta(i) = rho(i) - psi(i);

%=====
%   Vij
%=====
VFLx(i) = V(i)*cos(beta(i) - deltaFL(i)) +
psidot(i)*lf*sin(deltaFL(i)) - psidot(i)*(tf/2)*cos(deltaFL(i));
VFLy(i) = V(i)*sin(beta(i) - deltaFL(i)) +
psidot(i)*lf*cos(deltaFL(i)) - psidot(i)*(tf/2)*sin(deltaFL(i));

VFRx(i) = V(i)*cos(beta(i) - deltaFR(i)) +
psidot(i)*lf*sin(deltaFR(i)) - psidot(i)*(tf/2)*sin(deltaFR(i));
VFRy(i) = V(i)*sin(beta(i) - deltaFR(i)) +
psidot(i)*lf*cos(deltaFR(i)) + psidot(i)*(tf/2)*sin(deltaFR(i));

VRLx(i) = V(i)*cos(beta(i) + deltaRL(i)) +
psidot(i)*lr*sin(deltaRL(i)) - psidot(i)*(tr/2)*cos(deltaRL(i));
VRLy(i) = V(i)*sin(beta(i) + deltaRL(i)) -
psidot(i)*lr*cos(deltaRL(i)) - psidot(i)*(tr/2)*sin(deltaRL(i));

VRRx(i) = V(i)*cos(beta(i) - deltaRR(i)) -
psidot(i)*lr*sin(deltaRR(i)) + psidot(i)*(tr/2)*cos(deltaRR(i));
VRRy(i) = V(i)*sin(beta(i) - deltaRR(i)) -
psidot(i)*lr*cos(deltaRR(i)) - psidot(i)*(tr/2)*sin(deltaRR(i));

%=====
%   Slip
%=====
slip = slip_fun(VFLx(i), VFLy(i), VFRx(i), VFLy(i), VRLx(i), VRLy(i),
VRRx(i), VRRy(i), omegaFL(i), omegaFR(i), omegaRL(i), omegaRR(i));

sFLx(i) = slip(1);
sFLy(i) = slip(2);
sFL(i) = slip(3);

sFRx(i) = slip(4);

```

```

sFRy(i) = slip(5);
sFR(i)  = slip(6);

sRLx(i) = slip(7);
sRLy(i) = slip(8);
sRL(i)  = slip(9);

sRRx(i) = slip(10);
sRRy(i) = slip(11);
sRR(i)  = slip(12);

clear slip

%=====
%   Friction
%=====

friction = friction_fun(sFLx(i),sFLy(i),sFL(i), sFRx(i), sFRy(i),
sFR(i), sRLx(i), sRLy(i), sRL(i), sRRx(i), sRRy(i), sRR(i), z(i),
zdot(i), theta(i), thetadot(i), phi(i), phidot(i));

fFLx(i) = friction(1);
fFLy(i) = friction(2);
fFL(i)  = friction(3);

fFRx(i) = friction(4);
fFRy(i) = friction(5);
fFR(i)  = friction(6);

fRLx(i) = friction(7);
fRLy(i) = friction(8);
fRL(i)  = friction(9);

fRRx(i) = friction(10);
fRRy(i) = friction(11);
fRR(i)  = friction(12);

fFLz(i) = friction(13);
fFRz(i) = friction(14);
fRLz(i) = friction(15);
fRRz(i) = friction(16);

clear friction

end
snapshots
plots
%plots_B
%plots_C

```

### B.3: Codi funció de les equacions de moviment

```
function de = FWeom(t,e)
```



```

global m lf lr Jp Jx Jy tf tr W Jwf Jwr Rwf Rwr

global deltaFL deltaFR deltaRL deltaRR

x = e(1);
xdot = e(2);
y = e(3);
ydot = e(4);
z = e(5);
zdot = e(6);
psi = e(7);
psidot = e(8);
theta = e(9);
thetadot = e(10);
phi = e(11);
phidot = e(12);
omegaFL = e(13);
omegaFR = e(14);
omegaRL = e(15);
omegaRR = e(16);

V = sqrt(xdot^2 + ydot^2);
rho = atan2(ydot,xdot);
beta = rho-psi;

%=====
%   Input
% delta, Moment Pneumàtic
%=====

delta = steer_fun(t);
    deltaFL = delta(1);
    deltaFR = delta(2);
    deltaRL = delta(3);
    deltaRR = delta(4);
torque = accbrk_fun(t,omegaFL,omegaFR, omegaRL, omegaRR);
    TFL = torque(1);
    TFR = torque(2);
    TRL = torque(3);
    TRR = torque(4);

%=====
%   Vij
% Càlcul velocitats
%=====

rF = sqrt(lf^2 + (tf/2)^2);
angF = atan((tf/2)/lf);
rR = sqrt(lr^2 + (tr/2)^2);
angR = atan((tr/2)/lr);

```

```

VFLx = V*cos(beta - deltaFL) + psidot*rF*cos(pi/2 - deltaFL + angF);
VFLy = V*sin(beta - deltaFL) + psidot*rF*sin(pi/2 - deltaFL + angF);

VFRx = V*cos(beta - deltaFR) + psidot*rF*cos(pi/2 - deltaFR - angF);
VFRy = V*sin(beta - deltaFR) + psidot*rF*sin(pi/2 - deltaFR - angF);

VRLx = V*cos(beta - deltaRL) - psidot*rR*cos(pi/2 - deltaRL - angR);
VRLy = V*sin(beta - deltaRL) - psidot*rR*sin(pi/2 - deltaRL - angR);

VRRx = V*cos(beta - deltaRR) - psidot*rR*cos(pi/2 - deltaRL + angR);
VRRy = V*sin(beta - deltaRR) - psidot*rR*sin(pi/2 - deltaRL + angR);

%=====
%   Slip
% Angle de patinatge
%=====

slip = slip_fun(VFLx, VFLy, VFRx, VFRy, VRLx,VRLy, VRRx, VRRy,
omegaFL, omegaFR, omegaRL, omegaRR);

sFLx = slip(1);
sFLy = slip(2);
sFL  = slip(3);

sFRx = slip(4);
sFRy = slip(5);
sFR  = slip(6);

sRLx = slip(7);
sRLy = slip(8);
sRL  = slip(9);

sRRx = slip(10);
sRRy = slip(11);
sRR  = slip(12);

%=====
%   Friction
% Força de fricció
%=====

friction = friction_fun(sFLx, sFLy, sFL, sFRx, sFRy, sFR, sRLx,
sRLy, sRL, sRRx, sRRy, sRR, z, zdot, theta, thetadot, phi, phidot);

fFLx = friction(1);
fFLy = friction(2);
fFL  = friction(3);

fFRx = friction(4);
fFRy = friction(5);

```

```

fFR = friction(6);

fRLx = friction(7);
fRLy = friction(8);
fRL = friction(9);

fRRx = friction(10);
fRRy = friction(11);
fRR = friction(12);

fFLz = friction(13);
fFRz = friction(14);
fRLz = friction(15);
fRRz = friction(16);

%=====
% EOM
% Equacions de moviment
%=====
t;

de1 = xdot;
de2 = (fFLx*cos(deltaFL+psi) - fFLy*sin(deltaFL+psi) +
fFRx*cos(deltaFR+psi)...
      - fFRy*sin(deltaFR+psi) + fRLx*cos(psi+deltaRL) -
fRLy*sin(psi+deltaRL)...
      + fRRx*cos(deltaRR+psi) - fRRy*sin(deltaRR+psi))/m;
de3 = ydot;
de4 = (fFLx*sin(deltaFL+psi) + fFLy*cos(deltaFL+psi) +
fFRx*sin(deltaFR+psi)...
      + fFRy*cos(deltaFR+psi) + fRLx*sin(psi+deltaRL) +
fRLy*cos(psi+deltaRL)...
      + fRRx*sin(psi+deltaRR) + fRRy*cos(psi+deltaRR))/m;
de5 = zdot;
de6 = (fFLz + fFRz + fRLz + fRRz - W)/m;
de7 = psidot;
de8 = ((-fFLx*cos(deltaFL) + fFLy*sin(deltaFL) + fFRx*cos(deltaFR) -
fFRy*sin(deltaFR))*z...
      + (fFLx*sin(deltaFL) + fFLy*cos(deltaFL) + fFRx*sin(deltaFR) +
fFRy*cos(deltaFR))*lf)...
      + (-fRLx*cos(deltaRL) + fRLy*sin(deltaRL) + fRRx*cos(deltaRR) -
fRRy*sin(deltaRR))*z...
      + (-fRLx*sin(deltaRL) - fRLy*cos(deltaRL) - fRRx*sin(deltaRR) -
fRRy*cos(deltaRR))*z)/Jp;
de9 = thetadot;
de10 = ((fFLx*cos(deltaFL) - fFLy*sin(deltaFL)+fFRx*cos(deltaFR) -
fFRy*sin(deltaFR))*z ...
      + (fFLz+fFRz)*lf +
(fRLx*cos(deltaRL)+fRLy*sin(deltaRL)+fRRx*cos(deltaRR)-
fRRy*sin(deltaRR))*z - (fRLz+fRRz)*lr)/Jy;
de11 = phidot;
de12 = ((fFLx*sin(deltaFL) + fFLy*cos(deltaFL)+ fFRx*sin(deltaFR)+
fFRy*cos(deltaFR))*z ...
      + (fFLz-fFRz)*(tf/2) + (-fRLx*sin(deltaRL) + fRLy*cos(deltaRL)+
fRRx*sin(deltaRR)+ fRRy*cos(deltaRR))*z + (fRLz-fRRz)*(tr/2))/Jx;

```

```
de13 = (TFL - fFLx*Rwf) / Jwf;  
de14 = (TFR - fFRx*Rwf) / Jwf;  
de15 = (TRL - fRLx*Rwr) / Jwr;  
de16 = (TRR - fRRx*Rwr) / Jwr;
```

```
de = [de1 de2 de3 de4 de5 de6 de7 de8 de9 de10 de11 de12 de13 de14  
de15 de16]';
```

**B.4: Codi funció d'acceleració i frenada**

```
function T = accbrk_fun(t, omegaFL, omegaFR, omegaRL, omegaRR)

global BrkConstF BrkConstR AccConstF AccConstR

% Importació variables
type exp.txt;
fileID = fopen('exp.txt','r');
formatSpec = '%f';
sizeA = [Inf 1];
var_entrada = fscanf(fileID,formatSpec)
fclose(fileID);

% Acceleració / Frenada

if t<=0
    uaccbrk = var_entrada (11); %valor en % de l'accelerador

elseif t>0 && t<=1
    uaccbrk = var_entrada (12);

elseif t>1 && t<=2
    uaccbrk = var_entrada (13);

elseif t>2 && t<=3
    uaccbrk = var_entrada (14);

elseif t>3 && t<=4
    uaccbrk = var_entrada (15);

elseif t>4 && t<=5
    uaccbrk = var_entrada (16);

elseif t>5 && t<=6
    uaccbrk = var_entrada (17);

elseif t>6 && t<=7
    uaccbrk = var_entrada (18);

elseif t>7 && t<=8
    uaccbrk = var_entrada (19);

elseif t>8 && t<=9
    uaccbrk = var_entrada (20);

elseif t>9 && t<=10
    uaccbrk = var_entrada (21);

end
```

```
%Criteris de signe per la funció uaccbrk; >0 frena, <0 accelera
if uaccbrk >=0
    TFL = -uaccbrk*BrkConstF;
    TFR = -uaccbrk*BrkConstF;
    TRL = -uaccbrk*BrkConstR;
    TRR = -uaccbrk*BrkConstR;
elseif uaccbrk<0
    TFL = -uaccbrk*AccConstF;
    TFR = -uaccbrk*AccConstF;
    TRL = -uaccbrk*AccConstR;
    TRR = -uaccbrk*AccConstR;
end

T = [TFL TFR TRL TRR]';
```

**B.5: Codi funció de la direcció**

```
function delta = steer_fun(t)

global maxsteer toef toer

% Importació variables
type exp.txt;
fileID = fopen('exp.txt','r');
formatSpec = '%f';
sizeA = [Inf 1];
var_entrada = fscanf(fileID,formatSpec)
fclose(fileID);

if t<=0
    udelta = var_entrada (22);

elseif t>0 && t<=1
    udelta = var_entrada (23);

elseif t>1 && t<=2
    udelta = var_entrada (24);

elseif t>2 && t<=3
    udelta = var_entrada (25);

elseif t>3 && t<=4
    udelta = var_entrada (26);

elseif t>4 && t<=5
    udelta = var_entrada (27);

elseif t>5 && t<=6
    udelta = var_entrada (28);

elseif t>6 && t<=7
    udelta = var_entrada (29);

elseif t>7 && t<=8
    udelta = var_entrada (30);

elseif t>8 && t<=9
    udelta = var_entrada (31);

elseif t>9 && t<=10
    udelta = var_entrada (32);

end

% udelta = 0 si és una recta
```

```
% Criteris de signe udelta >0 gira en un sentit, udelta <0 en  
l'altre
```

```
delta = maxsteer*udelta*pi/180;
```

```
deltaFL = delta - toef;
```

```
deltaFR = delta + toef;
```

```
deltaRL = -toer;
```

```
deltaRR = toer;
```

```
delta = [deltaFL deltaFR deltaRL deltaRR]';
```



**B.6: Codi funció de lliscament**

```
function sfun = slip_fun(VFLx, VFLy, VFRx, VFRy, VRLx, VRLy, VRRx,
VRRy, omegaFL, omegaFR, omegaRL, omegaRR)
```

```
global Rwf Rwr
```

```
%Front left wheel
```

```
if omegaFL<0 & VFLx<0 & abs(omegaFL)*Rwf>abs(VFLx)
sFLx = 1 - VFLx/(omegaFL*Rwf);
sFLy = VFLy/abs(omegaFL*Rwf);
elseif omegaFL<0 & VFLx<0 & abs(omegaFL)*Rwf<abs(VFLx)
sFLx = -1 + (omegaFL*Rwf)/VFLx;
sFLy = VFLy/abs(VFLx);
elseif omegaFL>0 & VFLx>0 & abs(omegaFL)*Rwf>abs(VFLx)
sFLx = -1 + VFLx/(omegaFL*Rwf);
sFLy = VFLy/abs(omegaFL*Rwf);
elseif omegaFL>0 & VFLx>0 & abs(omegaFL)*Rwf<abs(VFLx)
sFLx = 1 - (omegaFL*Rwf)/VFLx;
sFLy = VFLy/abs(VFLx);
elseif omegaFL<0 & VFLx>=0
sFLx = 1;
sFLy = VFLy/abs(omegaFL);
elseif omegaFL>0 & VFLx<=0
sFLx = -1;
sFLy = VFLy/abs(omegaFL);
elseif omegaFL==0 & abs(VFLx)>0;
sFLx = sign(VFLx);
sFLy = VFLy/abs(VFLx);
elseif abs(omegaFL)*Rwf==abs(VFLx) & abs(VFLx)>0
sFLx = 0;
sFLy = VFLy/abs(VFLx);
elseif omegaFL==0 & VFLx == 0;
    sFLx = 0;
    sFLy = 0;
end
```

```
sFL = sqrt(sFLx^2 + sFLy^2);
```

```
%Front right wheel
```

```
if omegaFR<0 & VFRx<0 & abs(omegaFR)*Rwf>abs(VFRx)
sFRx = 1 - VFRx/(omegaFR*Rwf);
sFRy = VFRy/abs(omegaFR*Rwf);
elseif omegaFR<0 & VFRx<0 & abs(omegaFR)*Rwf<abs(VFRx)
sFRx = -1 + (omegaFR*Rwf)/VFRx;
sFRy = VFRy/abs(VFRx);
elseif omegaFR>0 & VFRx>0 & abs(omegaFR)*Rwf>abs(VFRx)
sFRx = -1 + VFRx/(omegaFR*Rwf);
sFRy = VFRy/abs(omegaFR*Rwf);
elseif omegaFR>0 & VFRx>0 & abs(omegaFR)*Rwf<abs(VFRx)
sFRx = 1 - (omegaFR*Rwf)/VFRx;
sFRy = VFRy/abs(VFRx);
elseif omegaFR<0 & VFRx>=0
sFRx = 1;
```

```

sFRy = VFRy/abs(omegaFR);
elseif omegaFR>0 & VFRx<=0
sFRx = -1;
sFRy = VFRy/abs(omegaFR);
elseif omegaFR==0 & abs(VFRx)>0;
sFRx = sign(VFRx);
sFRy = VFRy/abs(VFRx);
elseif abs(omegaFR)*Rwf==abs(VFRx) & abs(VFRx)>0
sFRx = 0;
sFRy = VFRy/abs(VFRx);
elseif omegaFR==0 & VFRx == 0;
    sFRx = 0;
    sFRy = 0;
end

sFR = sqrt(sFRx^2 + sFRy^2);

%Rear left wheel
if omegaRL<0 & VRLx<0 & abs(omegaRL)*Rwr>abs(VRLx)
sRLx = 1 - VRLx/(omegaRL*Rwr);
sRLy = VRLy/abs(omegaRL*Rwr);
elseif omegaRL<0 & VRLx<0 & abs(omegaRL)*Rwr<abs(VRLx)
sRLx = -1 + (omegaRL*Rwr)/VRLx;
sRLy = VRLy/abs(VRLx);
elseif omegaRL>0 & VRLx>0 & abs(omegaRL)*Rwr>abs(VRLx)
sRLx = -1 + VRLx/(omegaRL*Rwr);
sRLy = VRLy/abs(omegaRL*Rwr);
elseif omegaRL>0 & VRLx>0 & abs(omegaRL)*Rwr<abs(VRLx)
sRLx = 1 - (omegaRL*Rwr)/VRLx;
sRLy = VRLy/abs(VRLx);
elseif omegaRL<0 & VRLx>=0
sRLx = 1;
sRLy = VRLy/abs(omegaRL);
elseif omegaRL>0 & VRLx<=0
sRLx = -1;
sRLy = VRLy/abs(omegaRL);
elseif omegaRL==0 & abs(VRLx)>0;
sRLx = sign(VRLx);
sRLy = VRLy/abs(VRLx);
elseif abs(omegaRL)*Rwr==abs(VRLx) & abs(VRLx)>0
sRLx = 0;
sRLy = VRLy/abs(VRLx);
elseif omegaRL==0 & VRLx == 0;
    sRLx = 0;
    sRLy = 0;
end

sRL = sqrt(sRLx^2 + sRLy^2);

%Rear right wheel
if omegaRR<0 & VRRx<0 & abs(omegaRR)*Rwr>abs(VRRx)
sRRx = 1 - VRRx/(omegaRR*Rwr);
sRRy = VRRy/abs(omegaRR*Rwr);
elseif omegaRR<0 & VRRx<0 & abs(omegaRR)*Rwr<abs(VRRx)

```

```

sRRx = -1 + (omegaRR*Rwr)/VRRx;
sRRy = VRRy/abs(VRRx);
elseif omegaRR>0 & VRRx>0 & abs(omegaRR)*Rwr>abs(VRRx)
sRRx = -1 + VRRx/(omegaRR*Rwr);
sRRy = VRRy/abs(omegaRR*Rwr);
elseif omegaRR>0 & VRRx>0 & abs(omegaRR)*Rwr<abs(VRRx)
sRRx = 1 - (omegaRR*Rwr)/VRRx;
sRRy = VRRy/abs(VRRx);
elseif omegaRR<0 & VRRx>=0
sRRx = 1;
sRRy = VRRy/abs(omegaRR);
elseif omegaRR>0 & VRRx<=0
sRRx = -1;
sRRy = VRRy/abs(omegaRR);
elseif omegaRR==0 & abs(VRRx)>0;
sRRx = sign(VRRx);
sRRy = VRRy/abs(VRRx);
elseif abs(omegaRR)*Rwr==abs(VRRx) & abs(VRRx)>0
sRRx = 0;
sRRy = VRRy/abs(VRRx);
elseif omegaRR==0 & VRRx == 0;
    sRRx = 0;
    sRRy = 0;
end

sRR = sqrt(sRRx^2 + sRRy^2);

sfun = [sFLx sFLy sFL sFRx sFRy sFR sRLx sRLy sRL sRRx sRRy sRR]';

```

**B.7: Codi funció de fricció**

```
function friction = friction_fun(sFLx, sFLy, sFL, sFRx, sFRy, sFR,
sRLx, sRLy, sRL, sRRx, sRRy, sRR, z, zdot, theta, thetadot, phi,
phidot)
```

```
global B C D W lf lr h tf tr Kf Cf Kr Cr
```

```
%FRONT LEFT
```

```
if sFL==0;
    muFL = 0; muFLx = 0; muFLy = 0;
else
    muFL = D*sin(C*atan(B*sFL));
    muFLx = -(sFLx/sFL)*muFL;
    muFLy = -(sFLy/sFL)*muFL;
end
```

```
%FRONT RIGHT
```

```
if sFR==0;
    muFR = 0; muFRx = 0; muFRy = 0;
else
    muFR = D*sin(C*atan(B*sFR));
    muFRx = -(sFRx/sFR)*muFR;
    muFRy = -(sFRy/sFR)*muFR;
end
```

```
%REAR LEFT
```

```
if sRL==0;
    muRL = 0; muRLx = 0; muRLy = 0;
else
    muRL = D*sin(C*atan(B*sRL));
    muRLx = -(sRLx/sRL)*muRL;
    muRLy = -(sRLy/sRL)*muRL;
end
```

```
%REAR RIGHT
```

```
if sRR==0;
    muRR = 0; muRRx = 0; muRRy = 0;
else
    muRR = D*sin(C*atan(B*sRR));
    muRRx = -(sRRx/sRR)*muRR;
    muRRy = -(sRRy/sRR)*muRR;
end
```

```
%CALCULATION OF LOAD (f_(i,j)z)
```

```
%AZ on each wheel
```

```
AZFL = (z-h) + lf*sin(theta) + tf*0.5*sin(phi);
AZFR = (z-h) + lf*sin(theta) - tf*0.5*sin(phi);
AZRL = (z-h) - lr*sin(theta) + tr*0.5*sin(phi);
AZRR = (z-h) - lr*sin(theta) - tr*0.5*sin(phi);
```

```

% AZdot on each wheel
AZFLdot = zdot + lf*thetadot*cos(theta) + tf*0.5*phidot*cos(phi);
AZFRdot = zdot + lf*thetadot*cos(theta) - tf*0.5*phidot*cos(phi);
AZRLdot = zdot - lr*thetadot*cos(theta) + tr*0.5*phidot*cos(phi);
AZRRdot = zdot - lr*thetadot*cos(theta) - tr*0.5*phidot*cos(phi);

%fFi,j Z
fFLz = 0.5*(W*lr/(lf+lr)) - AZFL*Kf - AZFLdot*Cf;
fFRz = 0.5*(W*lr/(lf+lr)) - AZFR*Kf - AZFRdot*Cf;
fRLz = 0.5*(W*lf/(lf+lr)) - AZRL*Kr - AZRLdot*Cr;
fRRz = 0.5*(W*lf/(lf+lr)) - AZRR*Kr - AZRRdot*Cr;

%Forces fij, x,y on each wheel
fFL = fFLz*muFL; fFLx = fFLz*muFLx; fFLy = fFLz*muFLy;
fFR = fFRz*muFR; fFRx = fFRz*muFRx; fFRy = fFRz*muFRy;

fRL = fRLz*muRL; fRLx = fRLz*muRLx; fRLy = fRLz*muRLy;
fRR = fRRz*muRR; fRRx = fRRz*muRRx; fRRy = fRRz*muRRy;

friction = [fFLx fFLy fFL fFRx fFRy fFR fRLx fRLy fRL fRRx fRRy fRR
fFLz fFRz fRLz fRRz]';

```

**B.8: Funció Snapshots o representació gràfica de la trajectòria**

```

% Animation
L = 2.5;           % Vehicle length
w1 = 0.8;         % Front wheel lateral distance from CG
w2 = 0.9;         % Back wheel lateral distance from CG
l1 = 0.7;         % Front wheel longitudinal distance from CG
l2 = 1;           % Back wheel longitudinal distance from CG
lw = 0.7;         % Wheel diameter
hl1 = 0.4*L;      % Headlights long. distance
hlw = 0.4*w1;     % Headlights lat. distance

animfig = figure;
set(animfig,'DoubleBuffer','on','position',[0 0 400 664]);
delta = (deltaFL+deltaFR)/2;

% Plot the Road
%=====
XYaxes = axes;
hold on
%=====

set(XYaxes,'color',[ 0.8 0.8 0.8 ]);
axis equal
axis([min(x)-10 max(x)+10 min(y)-10 max(y)+10]);

rA_b = [L/2 0]';
rB_b = [-L/2 0]';
rC_b = [0 0]';
rD_b = [-0.9*L/2 0]';
rE_b = [0.3*L/2 0]';
rHL1_b = [hl1 hlw]';
rHL2_b = [hl1 -hlw]';
r1_b = [l1 w1]';
r2_b = [l1 -w1]';
r3_b = [-l2 w2]';
r4_b = [-l2 -w2]';
r1f_w = [lw/2 0]';
r1b_w = [-lw/2 0]';
r2f_w = [lw/2 0]';
r2b_w = [-lw/2 0]';
r3f_b = [-l2+lw/2 w2]';
r3b_b = [-l2-lw/2 w2]';
r4f_b = [-l2+lw/2 -w2]';
r4b_b = [-l2-lw/2 -w2]';

% Initial Position
xc_in = x(1);
yc_in = y(1);
psi_in = psi(1);
%delta_in = delta(1);
delta_in = 0;

RIB_in = [cos(psi_in) -sin(psi_in);sin(psi_in) cos(psi_in)];

```

```

RBW_in = [cos(delta_in) -sin(delta_in);sin(delta_in) cos(delta_in)];
RIW_in = RIB_in*RBW_in;

rA = RIB_in*rA_b + [xc_in yc_in]';
rB = RIB_in*rB_b + [xc_in yc_in]';
rC = RIB_in*rC_b + [xc_in yc_in]';
rD = RIB_in*rD_b + [xc_in yc_in]';
rE = RIB_in*rE_b + [xc_in yc_in]';
rHL1 = RIB_in*rHL1_b + [xc_in yc_in]';
rHL2 = RIB_in*rHL2_b + [xc_in yc_in]';
r1 = RIB_in*r1_b + [xc_in yc_in]';
r2 = RIB_in*r2_b + [xc_in yc_in]';
r3 = RIB_in*r3_b + [xc_in yc_in]';
r4 = RIB_in*r4_b + [xc_in yc_in]';
r1f = RIB_in*(RBW_in*r1f_w + [l1 w1]') + [xc_in yc_in]';
r1b = RIB_in*(RBW_in*r1b_w + [l1 w1]') + [xc_in yc_in]';
r2f = RIB_in*(RBW_in*r2f_w + [l1 -w1]') + [xc_in yc_in]';
r2b = RIB_in*(RBW_in*r2b_w + [l1 -w1]') + [xc_in yc_in]';
r3f = RIB_in*r3f_b + [xc_in yc_in]';
r3b = RIB_in*r3b_b + [xc_in yc_in]';
r4f = RIB_in*r4f_b + [xc_in yc_in]';
r4b = RIB_in*r4b_b + [xc_in yc_in]';

wheelaxis1 = line([r1(1) r2(1)], [r1(2)
r2(2)], 'Color', 'Black', 'EraseMode', 'Normal');
wheelaxis2 = line([r3(1) r4(1)], [r3(2)
r4(2)], 'Color', 'Black', 'EraseMode', 'Normal');
Body = line([rA(1) rB(1)], [rA(2)
rB(2)], 'Linewidth', 8.5, 'EraseMode', 'Normal');
Body1 = line([rD(1) rE(1)], [rD(2)
rE(2)], 'Linewidth', 6, 'EraseMode', 'Normal', 'Color', [ 0.502 1.000
1.000 ]);
Body2 = line([rC(1) rD(1)], [rC(2)
rD(2)], 'Linewidth', 5, 'EraseMode', 'Normal');
%CG =
line(xc_in, yc_in, 'Linewidth', 2, 'Marker', 'o', 'Color', 'Red', 'EraseMode',
'Normal');
wheel1 = line([r1f(1) r1b(1)], [r1f(2)
r1b(2)], 'Linewidth', 1.5, 'Color', 'Black', 'EraseMode', 'Normal');
wheel2 = line([r2f(1) r2b(1)], [r2f(2)
r2b(2)], 'Linewidth', 1.5, 'Color', 'Black', 'EraseMode', 'Normal');
wheel3 = line([r3f(1) r3b(1)], [r3f(2)
r3b(2)], 'Linewidth', 2.5, 'Color', 'Black', 'EraseMode', 'Normal');
wheel4 = line([r4f(1) r4b(1)], [r4f(2)
r4b(2)], 'Linewidth', 2.5, 'Color', 'Black', 'EraseMode', 'Normal');
HLight1 =
line([rHL1(1)], [rHL1(2)], 'color', 'yellow', 'marker', '.', 'markersize',
2);
HLight2 =
line([rHL2(1)], [rHL2(2)], 'color', 'yellow', 'marker', '.', 'markersize',
2);

tfinal = t(length(t));
if tfinal>0

```

```

tanim = 0:0.5:tfinal;
else
tanim = 0:-0.5:tfinal;
end

% tfinal = min(t);
% tanim = 0:-0.5:tfinal

for i = 1:length(tanim)
    xc_i = interp1(t,x,tanim(i));
    yc_i = interp1(t,y,tanim(i));
    psi_i = interp1(t,psi,tanim(i));
    delta_i = interp1(t,delta,tanim(i));

    RIB_i = [cos(psi_i) -sin(psi_i);sin(psi_i) cos(psi_i)];
    RBW_i = [cos(delta_i) -sin(delta_i);sin(delta_i) cos(delta_i)];

    rA = RIB_i*rA_b + [xc_i yc_i]';
    rB = RIB_i*rB_b + [xc_i yc_i]';
    rC = RIB_i*rC_b + [xc_i yc_i]';
    rD = RIB_i*rD_b + [xc_i yc_i]';
    rE = RIB_i*rE_b + [xc_i yc_i]';
    rHL1 = RIB_i*rHL1_b + [xc_i yc_i]';
    rHL2 = RIB_i*rHL2_b + [xc_i yc_i]';
    r1 = RIB_i*r1_b + [xc_i yc_i]';
    r2 = RIB_i*r2_b + [xc_i yc_i]';
    r3 = RIB_i*r3_b + [xc_i yc_i]';
    r4 = RIB_i*r4_b + [xc_i yc_i]';
    r1f = RIB_i*(RBW_i*r1f_w + [l1 w1]') + [xc_i yc_i]';
    r1b = RIB_i*(RBW_i*r1b_w + [l1 w1]') + [xc_i yc_i]';
    r2f = RIB_i*(RBW_i*r2f_w + [l1 -w1]') + [xc_i yc_i]';
    r2b = RIB_i*(RBW_i*r2b_w + [l1 -w1]') + [xc_i yc_i]';
    r3f = RIB_i*r3f_b + [xc_i yc_i]';
    r3b = RIB_i*r3b_b + [xc_i yc_i]';
    r4f = RIB_i*r4f_b + [xc_i yc_i]';
    r4b = RIB_i*r4b_b + [xc_i yc_i]';

    wheelaxis1 = line([r1(1) r2(1)], [r1(2)
    r2(2)], 'Color', 'Black', 'EraseMode', 'Normal');
    wheelaxis2 = line([r3(1) r4(1)], [r3(2)
    r4(2)], 'Color', 'Black', 'EraseMode', 'Normal');
    Body = line([rA(1) rB(1)], [rA(2)
    rB(2)], 'Linewidth', 8.5, 'EraseMode', 'Normal');
    Body1 = line([rD(1) rE(1)], [rD(2)
    rE(2)], 'Linewidth', 6, 'EraseMode', 'Normal', 'Color', [ 0.502 1.000
    1.000 ]);
    Body2 = line([rC(1) rD(1)], [rC(2)
    rD(2)], 'Linewidth', 5, 'EraseMode', 'Normal');
    %CG =
    line(xc_in,yc_in, 'Linewidth', 2, 'Marker', 'o', 'Color', 'Red', 'EraseMode',
    'Normal');
    wheel1 = line([r1f(1) r1b(1)], [r1f(2)
    r1b(2)], 'Linewidth', 1.5, 'Color', 'Black', 'EraseMode', 'Normal');

```



```
wheel2 = line([r2f(1) r2b(1)], [r2f(2)
r2b(2)], 'Linewidth', 1.5, 'Color', 'Black', 'EraseMode', 'Normal');
wheel3 = line([r3f(1) r3b(1)], [r3f(2)
r3b(2)], 'Linewidth', 2.5, 'Color', 'Black', 'EraseMode', 'Normal');
wheel4 = line([r4f(1) r4b(1)], [r4f(2)
r4b(2)], 'Linewidth', 2.5, 'Color', 'Black', 'EraseMode', 'Normal');
HLight1 =
line([rHL1(1)], [rHL1(2)], 'color', 'yellow', 'marker', '.', 'markersize',
2);
HLight2 =
line([rHL2(1)], [rHL2(2)], 'color', 'yellow', 'marker', '.', 'markersize',
2);

    drawnow
    pause(0.02)

end

%movie(M)
%movie2avi(M, '2Danim2', 'compression', 'Cinepak', 'quality', 100, 'fps', 1
2)

hold on

trajectory = plot(x,y, 'linestyle', ':', 'linewidth', 1);
set(trajectory, 'color', 'black')
xlabel('x(m)')
ylabel('y(m)')
```

**B.9: Funció de representació gràfica de resultats**

```

figure2 = figure;
set(figure2,'DoubleBuffer','on','position',[410 40 800 600]);

subplot(3,5,1)
axis([0,max(t),0,50])
plot(t,V*3.6)
xlabel('sec')
ylabel('km/h')
title('Speed')

subplot(3,5,11)
axis([0,max(t),30,50])
plot(t,(z-h)*1000)
xlabel('sec')
ylabel('mm')
title('Z-Motion cdg')

subplot(3,5,6)
plot(t,(deltaFL+deltaFR)/2*180/pi)
axis([0,max(t),-30,30])
xlabel('sec')
ylabel('deg')
title('Steering input')

subplot(3,5,2)
plot(t,theta*180/pi)
axis([0,max(t),-4,4])
xlabel('sec')
ylabel('deg')
title('Pitch')

subplot(3,5,7)
title('Roll')
plot(t,phi*180/pi)
axis([0,max(t),-20,20])
xlabel('sec')
ylabel('deg')
title('Roll')

subplot(3,5,12)
plot(t,[0;diff(ydot)])
%axis([0,max(t),-0.02,0.02])
xlabel('sec')
ylabel('-')
title('Lateral acceleration')

subplot(3,5,3)
axis([min(t),max(t),0,150])
plot(t,omegaFL,'b')
hold on
plot(t,omegaFR,'r')
xlabel('sec')

```

```
ylabel('rad/s')
title('Wheel speed F')
grid on

subplot(3,5,4)
axis([min(t),max(t),0,150])
plot(t,omegaRL,'b')
hold on
plot(t,omegaRR,'r')
xlabel('sec')
ylabel('rad/s')
title('Wheel speed R')
grid on

subplot(3,5,13)
plot(t,fFLz/9.81,'b')
hold on
plot(t,fFRz/9.81,'r')
axis([min(t),max(t),0,600])
xlabel('sec')
ylabel('Kg')
title('Load F')
grid on

subplot(3,5,14)
plot(t,fRLz/9.81,'b')
hold on
plot(t,fRRz/9.81,'r')
axis([min(t),max(t),0,600])
xlabel('sec')
ylabel('Kg')
title('Load R')
grid on

subplot(3,5,8)
title('Slip F')
plot(t,sFL,'b')
hold on
plot(t,sFR,'r')
axis([min(t),max(t),0,0.4])
xlabel('sec')
ylabel('%')
title('Slip F')
grid on

subplot(3,5,9)
title('Slip F')
plot(t,sRL,'b')
hold on
plot(t,sRR,'r')
axis([min(t),max(t),0,0.4])
xlabel('sec')
ylabel('%')
title('Slip R')
grid on
```

```
subplot(3,5,5)
plot(t,psidot,'b')
axis([min(t),max(t),-2,2])
xlabel('sec')
ylabel('rad/s')
title('Psi dot')
grid on

subplot(3,5,15)
axis([min(t),max(t),-20,20])
plot(t,((z-h)+lf*sin(theta))*1000,'b')
hold on
plot(t,((z-h)-lr*sin(theta))*1000,'r')
xlabel('sec')
ylabel('mm')
title('Z-Motion')
grid on

subplot(3,5,10)
plot(t,beta*180/pi,'b')
axis([min(t),max(t),-10,10])
xlabel('sec')
ylabel('deg')
title('Side Slip')
grid on
```