



EPS

Escola Politècnica
Superior

Projecte/Treball Fi de Carrera

Estudi: Enginyeria Tècn. Ind. Electrònica Ind. Pla 2002

Títol: Integració i estudi d'un conjunt de propulsors per a un robot submarí

Document: 1.Memòria. Annex A. Càlculs i Programa

Alumne: Albert Soler Hernández

Director/Tutor: Marc Carreras i Pérez

Departament: Electrònica, Informàtica i Automàtica

Àrea: ATC

Convocatòria (mes/any): febrer/2009

Volum 2/2

INDEX

A.1 MODEL DEL PROPULSOR	2
A.1.1 FORÇA EFECTUADA RESPECTE VELOCITAT DE GIR	2
A.1.2 VELOCITAT DE GIR RESPECTE A CONSIGNA	9
A.1.2.2 TRAM DE 55 A 90.....	14
A.1.2.3 TRAM DE -90 A -55	16
A.1.2.4 PUNTS DE TALL ENTRE LES RECTES	17
A.1.3 FORÇA EFECTUADA RESPECTE A CONSIGNA	19
A.1.4 CONSIGNA RESPECTE A FORÇA EFECTUADA	22
A.1.5 CONSUM ELÈCTRIC.....	23
A.1.6 COMPORTAMENT TRANSITORI.....	24
A.2.CÀLCUL DEL FLOTADOR.....	25
A.3 MANUAL DE MANTENIMENT	27
A.3.1 OPERACIONS DE MANTENIMENT PER ELS PROPULSORS.....	27
A.3.2 OPERACIONS DE MANTENIMENT RECOMANADES EN EL VEHICLE	27
A.4 CODI INFORMÀTIC	29
A.4.1 PANTALLA-H. CAPÇALERES	30
A.4.2 PANTALLA-C. CODI PANTALLES	33

A.1 MODEL DEL PROPULSOR

En aquest apartat es detallen els càlculs realitzats per a obtenir el model matemàtic en règim estacionari del propulsor. Concretament interessa trobar el valor de la força aportada per el propulsor en funció de la consigna donada a aquest.

El càlcul consta de 3 passos. En primer lloc obtenir la relació entre la força efectuada per el propulsor respecte a la seva velocitat de gir. En segon lloc s'obté la velocitat de gir, en règim estacionari, respecte a la consigna donada al propulsor. Finalment, en tercer lloc es relacionen les dues funcions obtingudes per a obtenir la relació entre consigna i força efectuada per el propulsor.

A.1.1 FORÇA EFECTUADA RESPECTE VELOCITAT DE GIR

Les dades recollides en l'assaig són les següents:

Velocitat (RPM)	Força mesurada (N)	Força mesurada (kgf.)	Força real (kgf.)
-887	-42	-4,29	-10,54
-840	-35	-3,57	-8,79
-818	-34	-3,47	-8,53
-780	-29	-2,96	-7,28
-723	-26	-2,65	-6,53
-580	-18,5	-1,89	-4,64
-516	-15	-1,53	-3,77
-390	-8	-0,82	-2,01
-292	-4,5	-0,46	-1,13
-230	-3	-0,31	-0,75
-144	-1,5	-0,15	-0,38
0	0	0,00	0,00
234	2	0,20	0,50
276	4	0,41	1,00
330	5	0,51	1,26
461	12	1,22	3,01
567	18	1,84	4,52
640	24	2,45	6,02
736	31,8	3,24	7,98
753	33	3,37	8,28
780	33	3,37	8,28
815	35	3,57	8,79
879	42	4,29	10,54

Taula 1 Assaig de força

En la primera columna es mostra la velocitat de gir mesurada pel propulsor, En la segona i tercera la força mesurada en l'assaig (en Newton i kilogram-força respectivament) i en la quarta columna la força real efectuada per el motor, que s'obté d'aplicar la relació de palanca del banc de proves (valor de 2.246).

A continuació es mostren gràficament aquestes mesures:

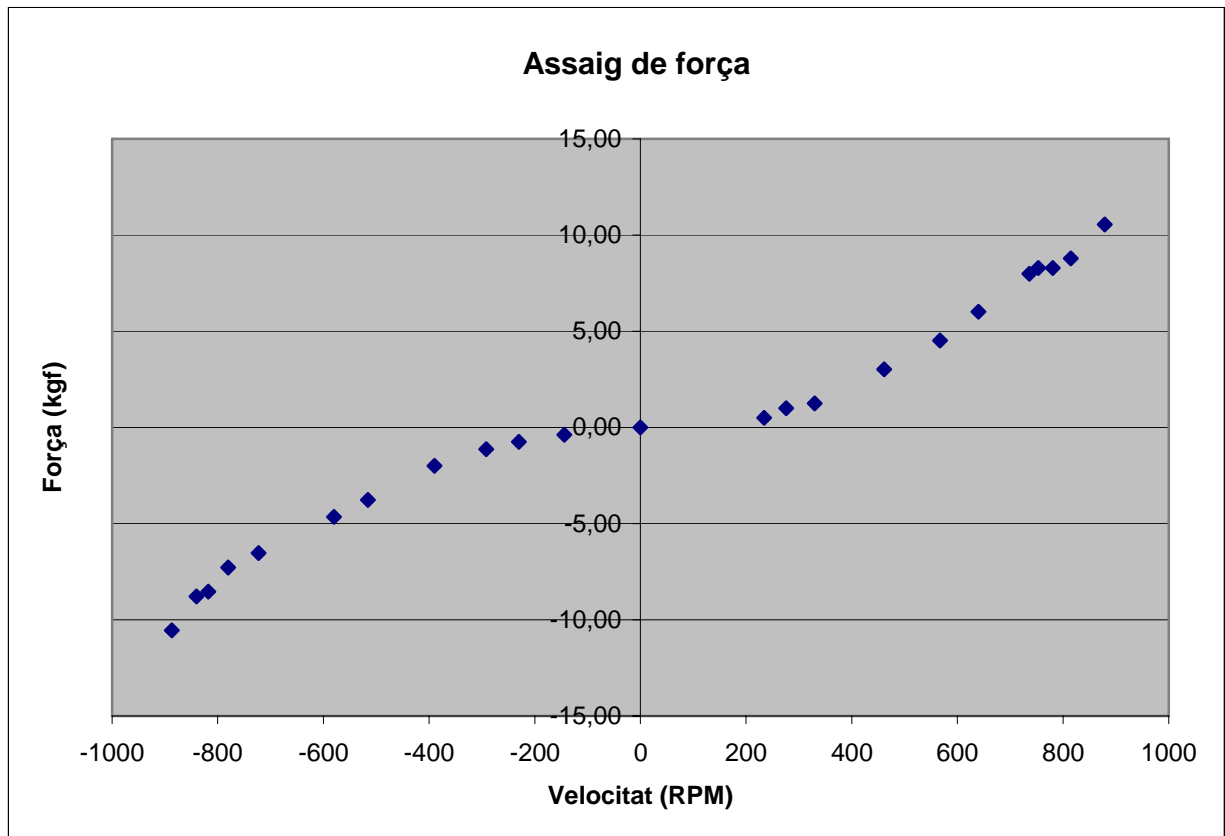


Figura 1 Assaig de força

A continuació es busca una funció quadràtica que s'ajusti a aquests punts. Es transformarà aquesta en una funció lineal posant a l'eix horitzontal el valor de la velocitat al quadrat. Amb els valors obtinguts obtindrem per regressió lineal una recta. Amb el pendent d'aquesta recta i com que sabem que la funció passa per el punt (0,0) obtindrem un valor que anomenarem C_t que ens definirà aquesta corba.

Per a acceptar l'aproximació com a bona es fixa un valor mínim del coeficient de determinació (r^2) de 0.99. La forma quadràtica en base a aquest valor ens donarà la funció matemàtica que estem buscant, és a dir, la força efectuada per el propulsor respecte a la velocitat de gir de l'hèlix.

Com es desprèn del full de característiques les forces efectuades en cada sentit són diferents, es buscarà un valor C_i per a cada sentit de gir. En el cas que obtinguéssim valors molt similars s'empraria un valor mig per als dos sentits.

Linealitzant els valors obtinguts i obtenim la següent taula:

MOTOR 1	
V^2	Força real (kgf.)
786769	-10,54
705600	-8,79
669124	-8,53
608400	-7,28
522729	-6,53
336400	-4,64
266256	-3,77
152100	-2,01
85264	-1,13
52900	-0,75
20736	-0,38
0	0,00
54756	0,50
76176	1,00
108900	1,26
212521	3,01
321489	4,52
409600	6,02
541696	7,98
567009	8,28
608400	8,28
664225	8,79
772641	10,54

Taula 2 Valors linealitzats

Les figures 2 i 3 mostren gràficament els valors obtinguts en l'assaig:

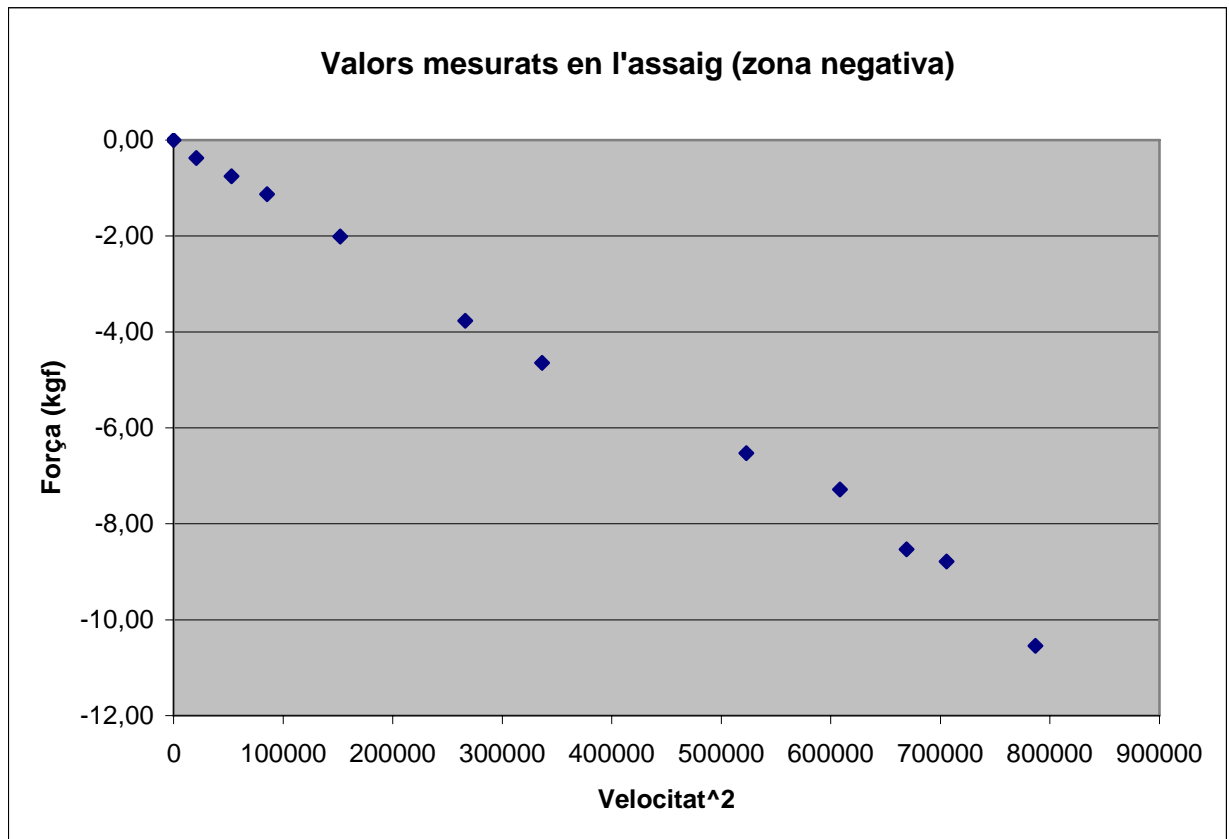


Figura 2 Valors linealitzats zona negativa

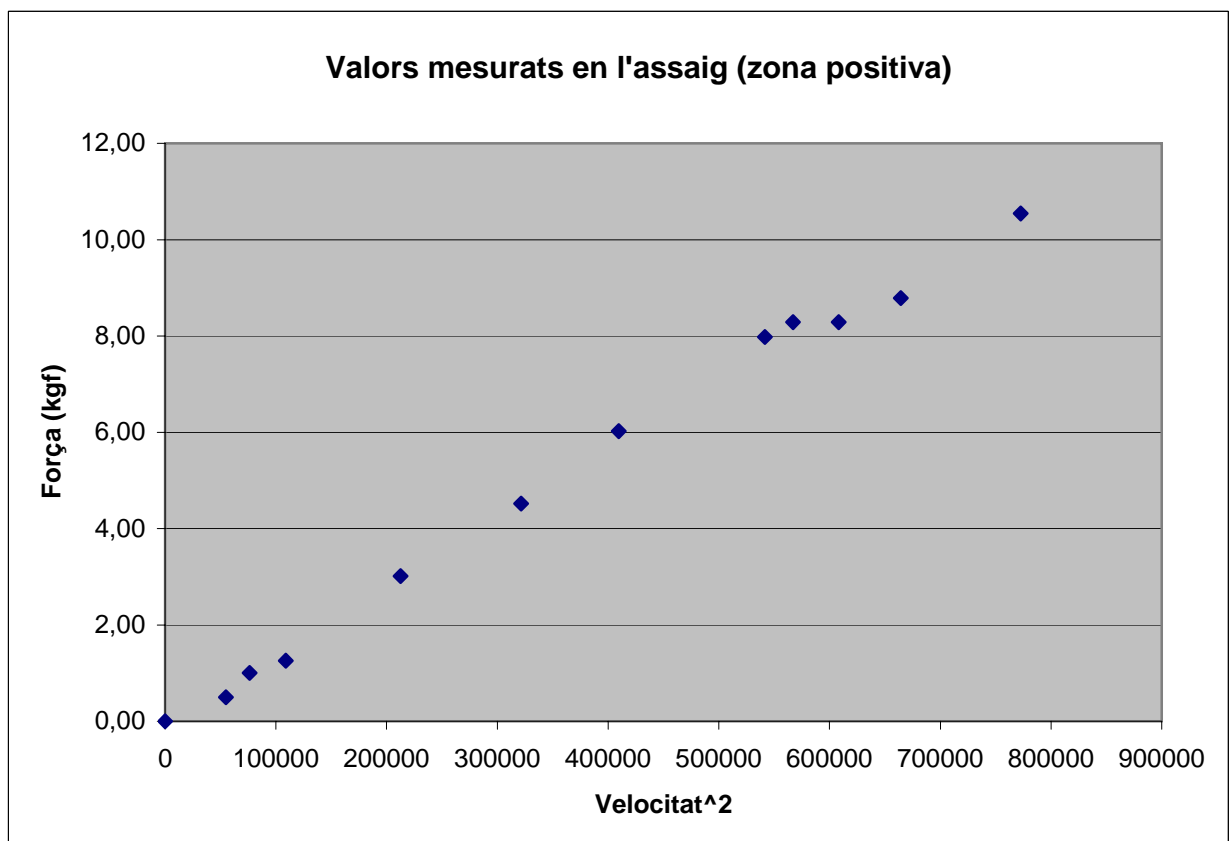


Figura 3 Valors linealitzats zona positiva

Busquem la recta de regressió de cada un dels trams. Obtindrem una recta del tipus:

$$y = b_0 + b_1x \quad (\text{Eq. 1})$$

on:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} \quad (\text{Eq. 2})$$

$$b_0 = \bar{y} - b_1 \bar{x} \quad (\text{Eq. 3})$$

En el cas negatiu:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = -1.262 \cdot 10^{-5} \quad (\text{Eq. 4})$$

$$b_0 = \bar{y} - b_1 \bar{x} = -0.1035 \quad (\text{Eq. 5})$$

$$y = b_0 + b_1x = -0.1035 - 1.262 \cdot 10^{-5} x \quad (\text{Eq. 6})$$

En el cas positiu:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = 1.399 \cdot 10^{-5} \quad (\text{Eq. 7})$$

$$b_0 = \bar{y} - b_1 \bar{x} = -0.0403 \quad (\text{Eq. 8})$$

$$y = b_0 + b_1x = -0.0403 + 1.399 \cdot 10^{-5} x \quad (\text{Eq. 9})$$

Per a acceptar com a bona aquesta aproximació busquem el coeficient de determinació (R^2) que ha de ser com a mínim de 099. L'obtenim de la següent manera:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} \quad (\text{Eq. 10})$$

On e es la diferència entre el valor mesurat i el valor esperat.

En el cas negatiu:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0.9949 \quad (\text{Eq. 11})$$

En el cas positiu:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0.9904 \quad (\text{Eq. 12})$$

Per tant acceptem que les rectes de regressió s'assemblen significativament a les dades mesurades.

Com que volem que les corbes passin per el punt (0,0) despreciam el valor dels paràmetres b_0 . Cometem en el pitjor dels casos un error de uns 100 grams força que es un valor insignificant respecte a la precisió de la mesura. Per tant :

$$C_{\text{tneg}} = -1.262 \cdot 10^{-5} \quad (\text{Eq. 13})$$

$$C_{\text{tpos}} = 1.399 \cdot 10^{-5} \quad (\text{Eq. 14})$$

Tot i ser dos valors pròxims no són iguals per aixó diferenciarem la funció en 2 parts segons sigui positiu o negatiu el sentit de gir.

La funció quadràtica en cada cas quedarà així:

$$\left\{ \begin{array}{l} y = C_{\text{tneg}} x^2 = -1.262 \cdot 10^{-5} \cdot x^2 \rightarrow x \leq 0 \\ y = C_{\text{tpos}} x^2 = 1.399 \cdot 10^{-5} \cdot x^2 \rightarrow x > 0 \end{array} \right. \quad (\text{Eq. 15})$$

o el que és el mateix:

$$\left\{ \begin{array}{l} F = -1.262 \cdot 10^{-5} \cdot \omega^2 \rightarrow x \leq 0 \\ F = 1.399 \cdot 10^{-5} \cdot \omega^2 \rightarrow x > 0 \end{array} \right. \quad (\text{Eq. 17})$$

$$(\text{Eq. 18})$$

Sobreposem la corba teòrica amb les dades obtingudes i veiem que aquesta s'ajusta visualment a les dades obtingudes.

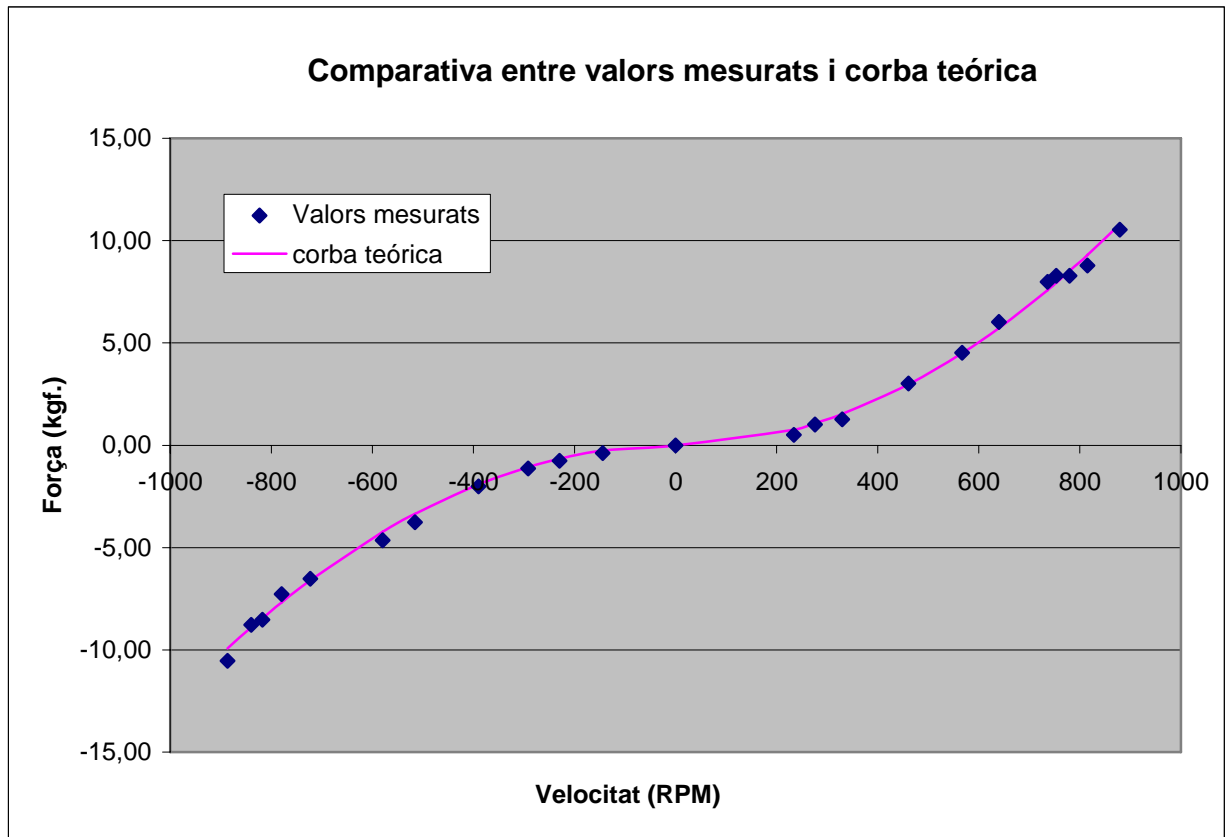


Figura 4 Comparativa

A.1.2 VELOCITAT DE GIR RESPECTE A CONSIGNA

Les dades per a cada motor recollides en l'assaig son les següents:

Consigna	RPM (M1)	RPM (M2)	RPM (M3)	RPM (M4)	RPM (M5)
-100	-889	-861	-899	-915	-850
-95	-896	-874	-902	-921	-856
-90	-887	-862	-905	-918	-852
-85	-830	-812	-840	-850	-811
-80	-760	-750	-772	-786	-740
-75	-691	-680	-705	-713	-668
-70	-619	-612	-625	-637	-598
-65	-540	-530	-547	-557	-520
-60	-465	-430	-460	-472	-436
-55	-380	-376	-380	-392	-362
-50	-350	-350	-340	-350	-339
-45	-312	-312	-305	-315	-307
-40	-280	-280	-274	-284	-276
-35	-250	-250	-240	-259	-248
-30	-216	-221	-204	-223	-216
-25	-184	-190	-174	-189	-183
-20	-151	-156	-141	-158	-156
-15	-117	-119	-120	-120	-117
-10	-72	-79	-57	-78	0
-5	-30	0	0	-39	0
0	0	0	0	0	0
5	21	0	0	0	0
10	72	0	62	73	67
15	116	111	102	116	111
20	156	146	146	150	131
25	189	180	178	183	184
30	222	213	208	220	217
35	250	240	240	250	239
40	280	278	270	284	276
45	310	304	301	312	300
50	340	336	333	340	331
55	380	365	376	384	360
60	460	447	459	465	431
65	540	520	537	546	510
70	617	606	615	625	591
75	691	676	703	706	662
80	763	745	774	777	735
85	830	810	841	847	805
90	896	860	894	909	850
95	897	863	900	911	855
100	896	863	893	909	850

Taula 3 Assaig de velocitat

A continuació es representen aquestes dades gràficament:

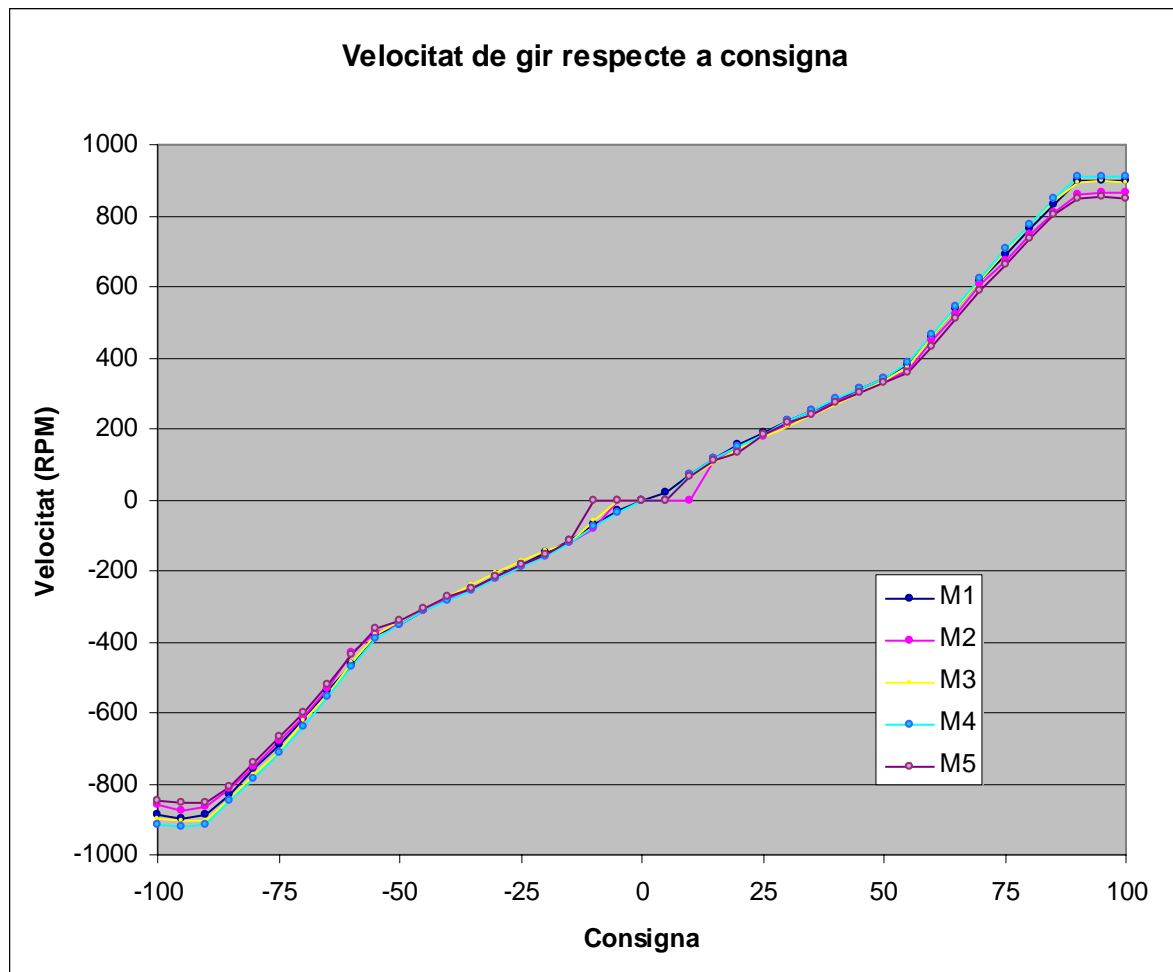


Figura 5 Assaig de velocitat

Es poden apreciar 5 zones diferenciades, una zona més o menys rectilínea al centre dues zones amb un pendent més pronunciat més cap a l'exterior i unes zones horitzontals en els extrems.

Les zones exteriors de saturació són degudes probablement al fet que els motors són alimentats a 42 volts i no a 48 volts. En aquests punts es produeix una saturació. Per a evitar això es procurarà treballar entre els valors de -90 i 90.

En la zona central es veuen alguns punts anòmals. Aquests punts són deguts al mal ajustatge de les parts mecàniques de l'helix. Amb una mica de lubricació desapareixeran. Aquests punts es consideren dades anòmales i no es tindran en compte.

A continuació s'estudiarà cada un dels trams rectilinis i es buscarà una funció rectilínea que s'ajusti a elles.

A.1.2.1 TRAM DE 0 A 55

Les dades en aquest tram són aquestes:

consigna	Velocitat (RPM)				
0	0	0	0	0	0
5	21				
10	72	62	73	67	
15	116	111	102	116	111
20	156	146	146	150	131
25	189	180	178	183	184
30	222	213	208	220	217
35	250	240	240	250	239
40	280	278	270	284	276
45	310	304	301	312	300
50	340	336	333	340	331
55	380	365	376	384	360

Taula 4 Valors de 0 a 55

Busquem la recta de regressió lineal:

$$y = b_0 + b_1x \quad (\text{Eq. 19})$$

on :

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} \quad (\text{Eq. 20})$$

$$b_0 = \bar{y} - b_1\bar{x} \quad (\text{Eq. 21})$$

En aquest cas:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = 6.72 \quad (\text{Eq. 22})$$

$$b_0 = \bar{y} - b_1\bar{x} = 6.67 \quad (\text{Eq. 23})$$

$$y = b_0 + b_1x = 6.67 + 6.72x \quad (\text{Eq. 24})$$

El coeficient de determinació (r^2) es:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0,995 \quad (\text{Eq. 25})$$

Com que la recta ha de passar per (0,0) ometem el valor de b_0 i considerem que l'error de 6.72 rpm s despreciable.

La recta queda dons d'aquesta manera:

$$y = 6.72x \quad (\text{Eq. 26})$$

B.1.2.2 Tram de -55 a 0

Seguim el mateix mètode que en el cas anterior.

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = 6.78 \quad (\text{Eq. 27})$$

$$b_0 = \bar{y} - b_1 \bar{x} = 8.84 \quad (\text{Eq. 28})$$

$$y = b_0 + b_1 x = 8.84 + 6.78x \quad (\text{Eq. 29})$$

El coeficient de determinació (R^2) és:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0,9952 \quad (\text{Eq. 30})$$

Com que la recta ha de passar per (0,0) ometem el valor de b_0 i considerem que l'error de 8.84 rpm és despreciable.

La recta queda dons d'aquesta manera:

$$y = 6.78x \quad (\text{Eq. 31})$$

Com que aquesta recta i l'anterior són molt similars optem per a considerar-les iguals i en considerarem el valor mig, quedant la funció així:

$$y = \left(\frac{6.78 + 6.72}{2} \right) x = 6.75x \quad (\text{Eq. 32})$$

Anomenem les variables i obtenim:

$$\omega = 6.75c \quad (\text{Eq. 33})$$

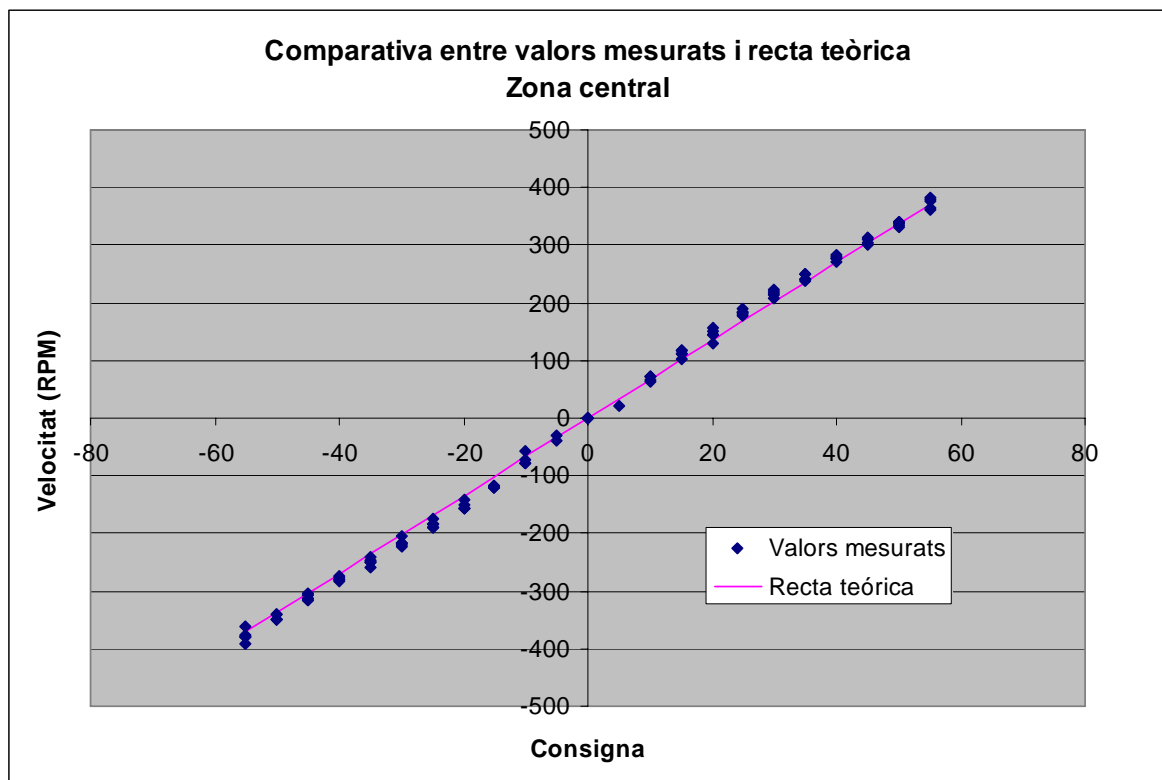
On:

C: consigna

ω : Velocitat de gir

Sobreposem la recta teòrica amb les dades obtingudes i veiem que aquesta s'ajusta visualment a les dades obtingudes.

La figura 6 mostra gràficament la similitud entre els valors mesurats i la recta teòrica



Fi

gura 6 Comparativa zona central

A.1.2.2 TRAM DE 55 A 90

Les dades en aquest tram són aquestes:

consigna	Velocitat (RPM)				
55	380	365	376	384	360
60	460	447	459	465	431
65	540	520	537	546	510
70	617	606	615	625	591
75	691	676	703	706	662
80	763	745	774	777	735
85	830	810	841	847	805
90	896	860	894	909	850

Taula 4 Valors de 55 a 90

Busquem la recta de regressió lineal:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = 14.75 \quad (\text{Eq. 34})$$

$$b_0 = \bar{y} - b_1 \bar{x} = -429 \quad (\text{Eq. 35})$$

$$y = b_0 + b_1 x = -429 + 14.75x \quad (\text{Eq. 36})$$

El coeficient de determinació (R^2) és:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0,9892 \quad (\text{Eq. 37})$$

Aquest valor és molt pròxim a 0.99 per aixó considerem bona la aproximació. Aquesta aproximació és:

$$\omega = 14.75c - 429 \quad (\text{Eq. 38})$$

On:

c: consigna

ω : Velocitat de gir

La figura 7 mostra gràficament la similitud entre els valors mesurats i la recta teòrica

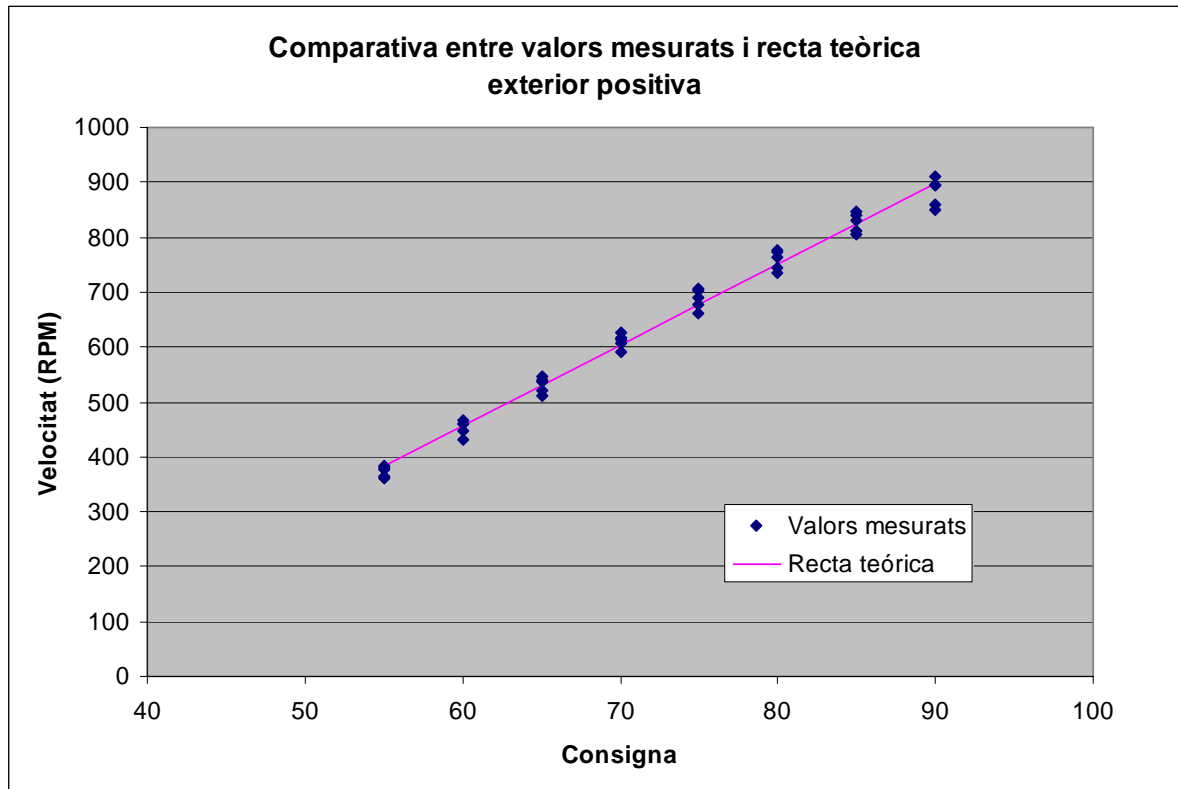


Figura 7 Comparativa zona exterior positiva

A.1.2.3 TRAM DE -90 A -55

Les dades en aquest tram són aquestes:

consigna	Velocitat (RPM)				
-90	-887	-862	-905	-918	-852
-85	-830	-812	-840	-850	-811
-80	-760	-750	-772	-786	-740
-75	-691	-680	-705	-713	-668
-70	-619	-612	-625	-637	-598
-65	-540	-530	-547	-557	-520
-60	-465	-430	-460	-472	-436
-55	-380	-376	-380	-392	-362

Taula 5 Valors de -90 a 55

Busquem la recta de regressió lineal:

$$b_1 = \frac{\sum x \cdot y - \bar{y} \sum x}{\sum x^2 - \bar{x} \sum x} = 14.68 \quad (\text{Eq. 39})$$

$$b_0 = \bar{y} - b_1 \bar{x} = 421 \quad (\text{Eq. 40})$$

$$y = b_0 + b_1 x = 421 + 14.68 x \quad (\text{Eq. 41})$$

El coeficient de determinació (R^2) es:

$$R^2 = \frac{1 - \sum e^2}{\sum (y - \bar{y})^2} = 0,9878 \quad (\text{Eq. 42})$$

Aquest valor es molt pròxim a 0.99 per aixó considerem bona la aproximació. Aquesta aproximació és:

$$\omega = 14.68c + 421 \quad (\text{Eq. 43})$$

On:

C: consigna

ω : Velocitat de gir

La figura 8 mostra gràficament la similitud entre els valors mesurats i la recta teòrica

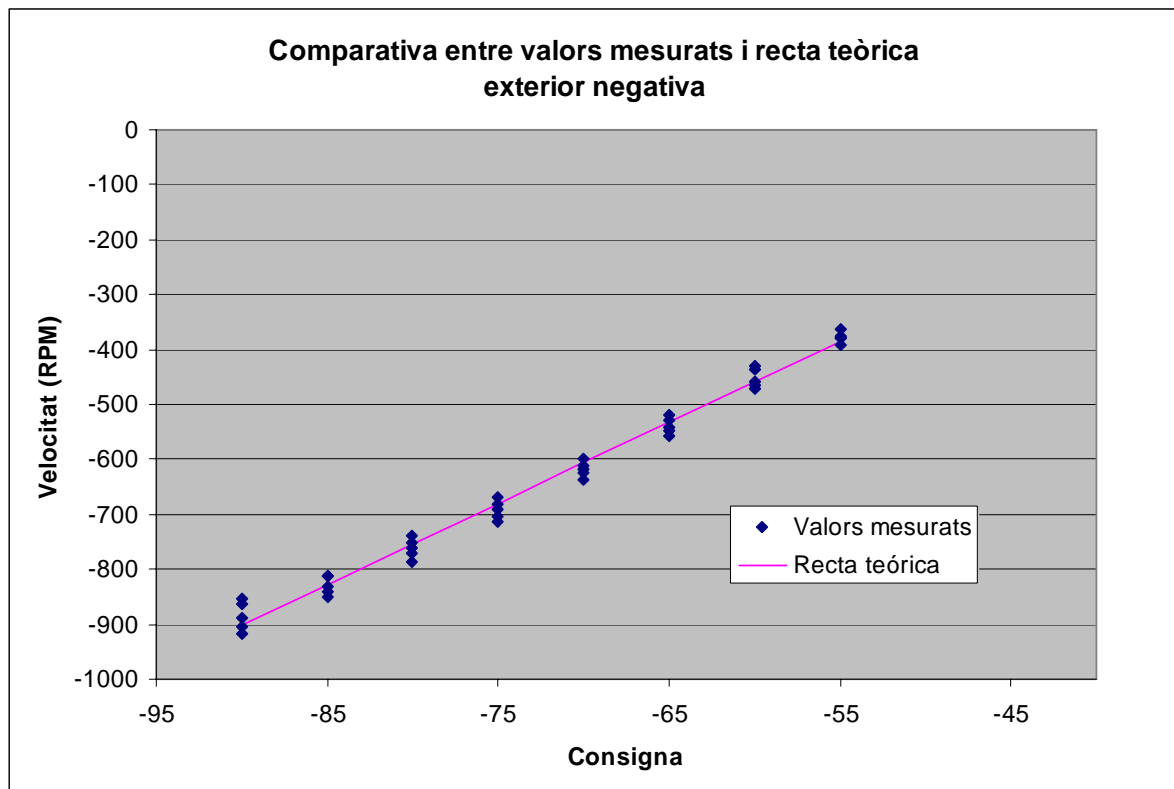


Figura 8 Comparativa zona exterior negativa

A.1.2.4 PUNTS DE TALL ENTRE LES RECTES

Hem observat gràficament que els punts de tall entre les rectes trobades ronda per 55 i -55, però no sabem exactament on. En aquest apartat es busquen els punts de tall per a poder obtenir una funció continua.

Per el punt de tall positiu plantegem aquest sistema d'equacions:

$$\begin{cases} \omega = 6.75c \\ \omega = 14.75c - 429 \end{cases} \Rightarrow \begin{cases} \omega = 361.97 \\ c = 53.625 \end{cases} \quad \begin{matrix} \text{(Eq. 44)} \\ \text{(Eq. 45)} \end{matrix}$$

Per el punt de tall negatiu plantegem aquest altre sistema:

$$\begin{cases} \omega = 6.75c \\ \omega = 14.68c + 421 \end{cases} \Rightarrow \begin{cases} \omega = 358.35 \\ c = -53.089 \end{cases} \quad \begin{matrix} \text{(Eq. 46)} \\ \text{(Eq. 47)} \end{matrix}$$

Amb aquests punts de tall ja posem plantejar la funció que ens relaciona la velocitat de gir amb la consigna:

$$\left\{ \begin{array}{l} \omega = 14.68c + 421 \rightarrow -90 < c < -53.089 \end{array} \right. \quad (\text{Eq. 48})$$

$$\left\{ \begin{array}{l} \omega = 6.75c \rightarrow -53.089 \leq c \leq 53,625 \end{array} \right. \quad (\text{Eq. 49})$$

$$\left\{ \begin{array}{l} \omega = 14.75c - 429 \rightarrow 53,625 < c < 90 \end{array} \right. \quad (\text{Eq. 50})$$

La figura 9 mostra gràficament la similitud entre els valors mesurats i les funcions obtingudes

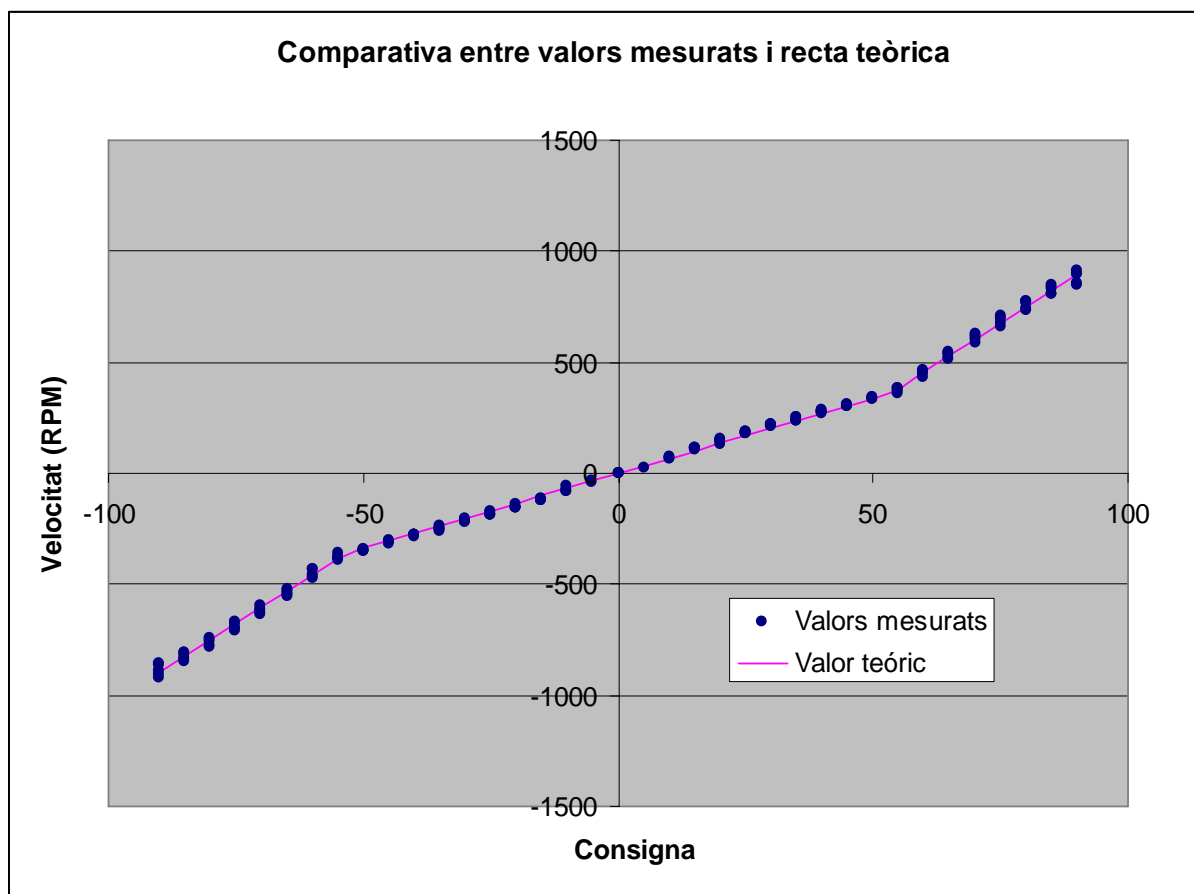


Figura 9 Comparativa general

A.1.3 FORÇA EFECTUADA RESPECTE A CONSIGNA

Busquem relacionar les dues equacions obtingudes anteriorment. Seguirem els següents passos:

$$\begin{cases} F(w) \\ w(c) \end{cases} \Rightarrow F(c) = F(w(c)) \quad (\text{Eq. 51})$$

Seguirem aquest procés per a cada un dels trams que hem obtingut. Diferenciem 4 trams: De -90 a -53.089, de -53.089 a 0, de 0 a 53.625 i de 53.625 a 90

De -90 a -53.089

$$\begin{cases} F(w) = -1.262 \cdot 10^{-5} w^2 \\ w(c) = 14.68c + 421 \end{cases} \Rightarrow F(c) = F(w(c)) = -1.262 \cdot 10^{-5} (14.68c + 421)^2 \quad (\text{Eq. 52})$$

$$F(c) = -0.0027196c^2 - 0.1559903c - 2.2367814 \quad (\text{Eq. 53})$$

De -53.089 a 0

$$\begin{cases} F(w) = -1.262 \cdot 10^{-5} w^2 \\ w(c) = 6.75c \end{cases} \Rightarrow F(c) = F(w(c)) = -1.262 \cdot 10^{-5} (6.75c)^2 \quad (\text{Eq. 54})$$

$$F(c) = -0.000575c^2 \quad (\text{Eq. 55})$$

B.1.3.3 De 0 a 53.625

$$\begin{cases} F(w) = 1.399 \cdot 10^{-5} w^2 \\ w(c) = 6.75c \end{cases} \Rightarrow F(c) = F(w(c)) = 1.399 \cdot 10^{-5} (6.75c)^2 \quad (\text{Eq. 56})$$

$$F(c) = -0.0006374c^2 \quad (\text{Eq. 57})$$

De 53.625 a 90

$$\begin{cases} F(w) = 1.399 \cdot 10^{-5} w^2 \\ w(c) = 14.75c - 429 \end{cases} \Rightarrow F(c) = F(w(c)) = 1.399 \cdot 10^{-5} (14.75c - 429)^2 \quad (\text{Eq. 58})$$

$$F(c) = 0.0030437c^2 - 0.1770504c + 2.5747336 \quad (\text{Eq. 59})$$

És possible que els punts de tall hagin variat a causa d'arrodoniments i ajustament de les corbes, a continuació es calculen els punts de tall entre les ultimes funcions obtingudes.

$$\begin{cases} F(c) = -0.0027196c^2 - 0.1559903c - 2.2367814 & \text{(Eq. 60)} \end{cases}$$

$$\begin{cases} F(c) = -0.000575c^2 & \text{(Eq. 61)} \end{cases}$$

$$-0.0027196c^2 - 0.1559903c - 2.2367814 = -0.000575c^2 \quad \text{(Eq. 62)}$$

$$c = -19,7 \quad \text{(Eq. 63)}$$

$$c = -52,7 \quad \text{(Eq. 64)}$$

El primer punt de tall es -52,7

El segon punt de tall és 0. Per a obtenir el tercer plantegem:

$$\begin{cases} F(c) = 0.0030437c^2 - 0.1770504c + 2.5747336 & \text{(Eq. 65)} \end{cases}$$

$$\begin{cases} F(c) = 0.0006374c^2 & \text{(Eq. 66)} \end{cases}$$

$$0.0030437c^2 - 0.1770504c + 2.5747336 = 0.0006374c^2 \quad \text{(Eq. 67)}$$

$$c = -19,95 \quad \text{(Eq. 68)}$$

$$c = -53.62 \quad \text{(Eq. 69)}$$

El tercer punt de tall és 53,62.

Les equacions que ens relacionen la força aportada per el propulsor respecte a la consigna són:

$$\begin{cases} F(c) = -0.0027196c^2 - 0.1559903c - 2.2367814 \rightarrow -90 < c < -52.7 & \text{(Eq. 70)} \end{cases}$$

$$\begin{cases} F(c) = -0.000575c^2 \rightarrow -52.7 \leq c < 0 & \text{(Eq. 71)} \end{cases}$$

$$\begin{cases} F(c) = 0.0006374c^2 \rightarrow 0 \leq c < 53.62 & \text{(Eq. 72)} \end{cases}$$

$$\begin{cases} F(c) = 0.0030437c^2 - 0.1770504c + 2.5747336 \rightarrow 53.62 \leq c < 90 & \text{(Eq. 73)} \end{cases}$$

La figura 10 mostra gràficament aquesta funció:

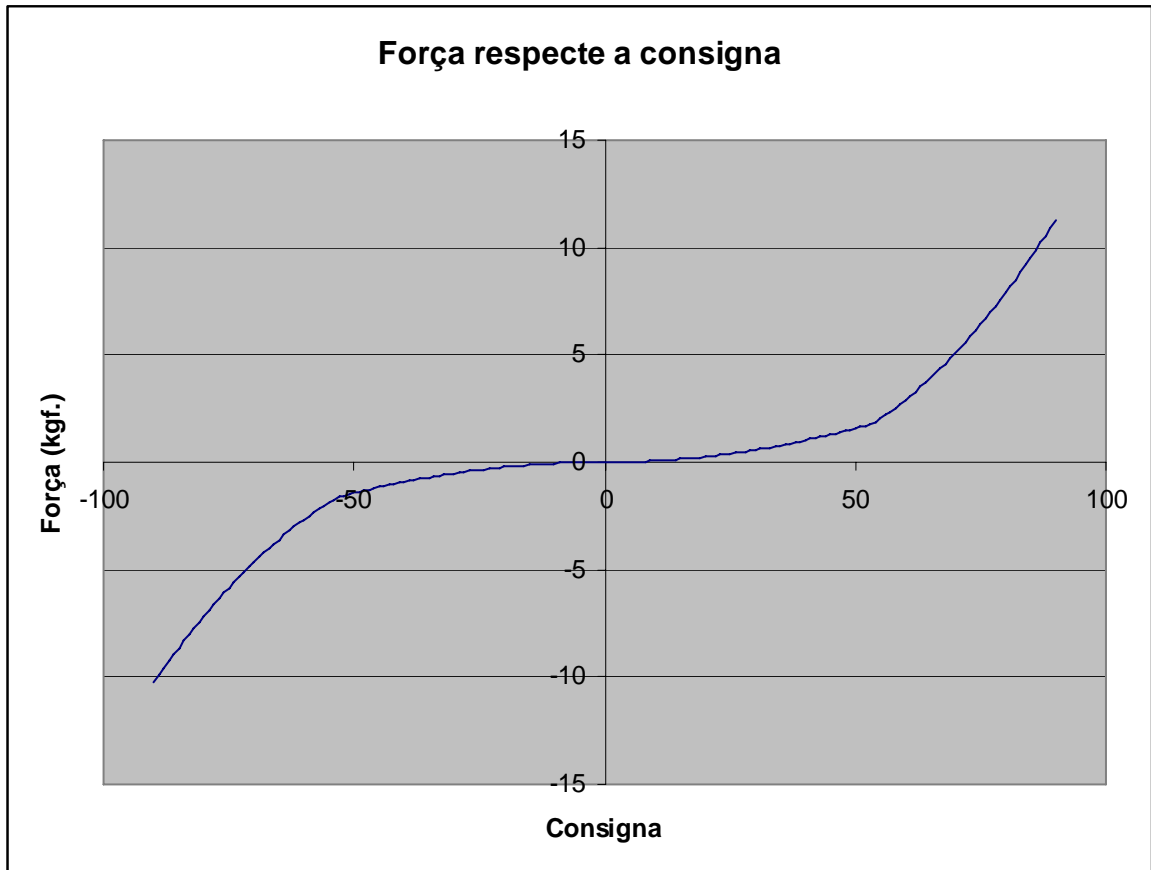


Figura 10 Força respecte a consigna

A.1.4 CONSIGNA RESPECTE A FORÇA EFECTUADA

Aquesta funció s'obté de la inversa de la funció anterior. Queda de la següent forma:

$$C(f) = -0.156 - \frac{\sqrt{-0.0108f}}{0.054392} \rightarrow f < -1.6 \quad (\text{Eq. 74})$$

$$C(f) = \sqrt{\frac{f}{-0.000575}} \rightarrow -1.6 \leq f < 0 \quad (\text{Eq. 75})$$

$$C(f) = \sqrt{\frac{f}{0.0006374}} \rightarrow 0 \leq f < 1.8 \quad (\text{Eq. 76})$$

$$C(f) = -0.177 + \frac{\sqrt{0.01218}}{0.006088} \rightarrow f \geq -1.6 \quad (\text{Eq. 77})$$

La figura 11 mostra gràficament aquesta funció.

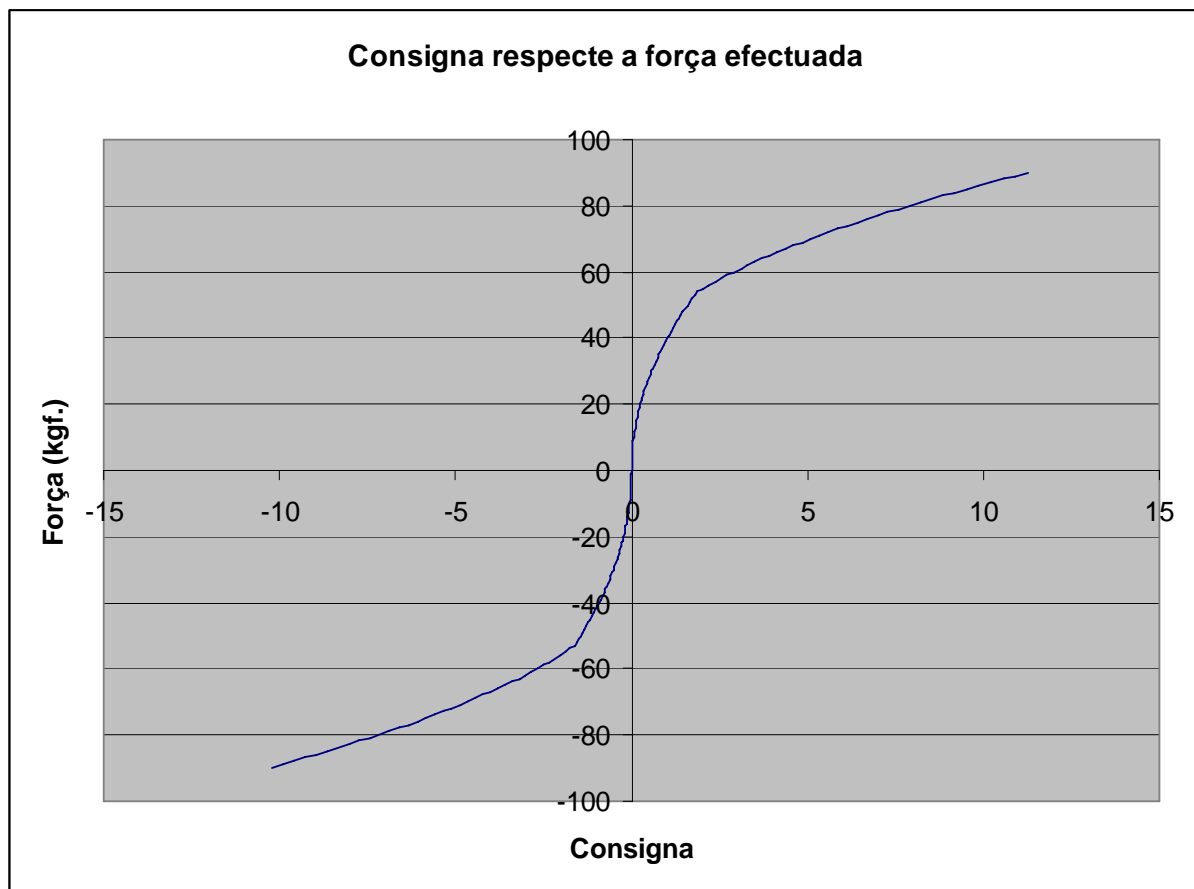


Figura 11 Consigna respecte a força

A.1.5 CONSUM ELÈCTRIC

La taula 6 mostra els consums elèctrics a diverses velocitats de gir. La Figura 12 mostra aquestes dades gràficament:

Velocitat (rpm)	Intensitat (A)	Velocitat (rpm)	Intensitat (A)
897	5,4	-30	0,1
897	5,4	-72	0,1
896	5,4	-117	0,1
830	4	-151	0,2
763	3,5	-184	0,2
691	2,7	-216	0,3
617	1,9	-250	0,3
540	1,4	-280	0,4
460	1	-312	0,5
380	0,7	-350	0,5
340	0,6	-380	0,7
310	0,5	-465	1
280	0,4	-540	1,3
250	0,3	-619	1,9
222	0,3	-691	2,6
189	0,2	-760	3,5
156	0,2	-830	4,4
116	0,15	-889	5,4
72	0,1	-889	5,4
21	0,1	-889	5,4
0	0		

Taula 7 Assaig d'intensitat

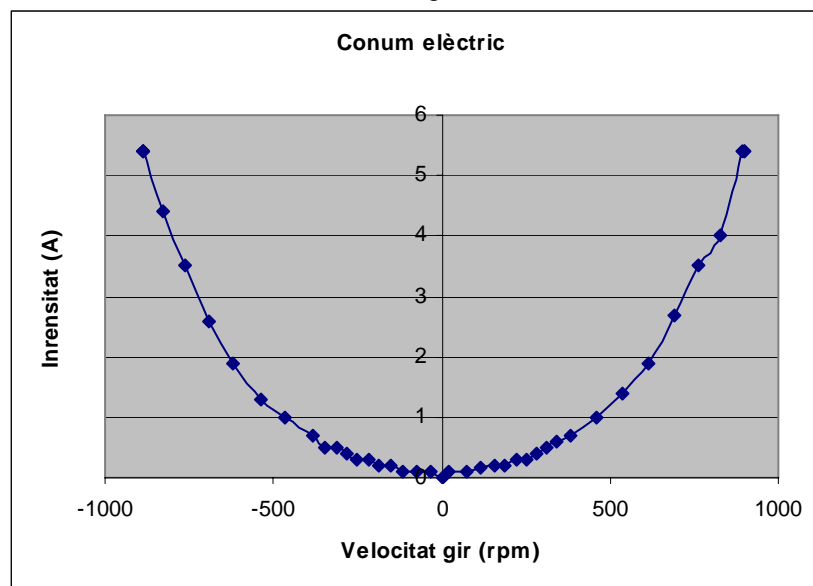


Figura 12 Assaig d'intensitat

De les dades es despren la següent aproximació.

$$I(v) = 4 \cdot 10^{-12} v^4 - 2 \cdot 10^{-10} v^3 + 9 \cdot 10^{-5} v \quad (\text{Eq. 78})$$

On:

I: Intensitat en ampers

v: Velocitat de gir en rpm.

A.1.6 COMPORTAMENT TRANSITORI

Els temps de reacció respecte a escalons són molt curts, s'aproxima el sistema a un sistema instantani. No es consideren fenòmens transitoris, tot i que aixó no eximeix d'implantar un regulador en llaç tancat en cada propulsor.

A.2.CÀLCUL DEL FLOTADOR

En aquest apartat es detalla el procés de càlcul del volum del flotador i les dimensions d'aquest per a tenir una flotabilitat d'aproximadament 3kg. Per a obtenir la flotabilitat neutra es compensarà amb 3Kg de plom repartits uniformement per l'estructura del vehicle.

El pes del vehicle submergit és de 16,5Kp. S'ha de trobar el volum del flotador que compensi aquest pes. Es sap que la densitat del material es de 0.43 Kg/dm². S'ha de tenir en compte el pes del propi flotador.

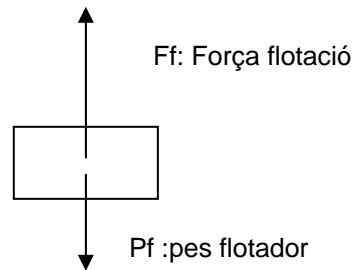


Figura 13 Balanç de forces

Es pretén tenir una flotabilitat d'uns 3 kp. Per tant la força de flotació que ha d'aportar el flotador ha de ser de 13.5Kp. Cal que:

$$13.5 = F_f - P_f \quad (\text{Eq. 79})$$

$$F_f = 1 \cdot v \quad (\text{Eq. 80})$$

$$P_f = 0.43 \cdot v \quad (\text{Eq. 81})$$

$$16.5 = v - 0.43 \cdot v = 0.57 \cdot v \rightarrow v = \frac{13.5}{0.57} = 23.68 \text{dm}^2 \quad (\text{Eq. 82})$$

Es parteix d'un prisma rectangular de 60x80x7 cm. El volum inicial es:

$$v_1 = 60 \cdot 80 \cdot 7 = 33600 \text{cm}^2 = 33.6 \text{dm}^2 \quad (\text{Eq. 83})$$

Es retallaran uns xamfrans de 45° i 5cm de costat en la part davantera i posterior. El volum del flotador amb aquests xamfrans es:

$$v_2 = 33600 - 2 \cdot \left(\frac{6 \cdot 6}{2} \right) \cdot 60 = 31440 \text{ cm}^2 = 31.44 \text{ dm}^2 \quad (\text{Eq. 84})$$

Es considera necessari un forat de 30 cm de diàmetre al centre per a facilitar la circulació de l'aigua impulsada per el propulsor vertical. El volum del flotador amb aquest forat serà:

$$v_3 = 31440 - (\pi \cdot 15^2) \cdot 7 = 26494 \text{ cm}^2 = 26.49 \text{ dm}^2 \quad (\text{Eq. 85})$$

Es pretén repartir la resta de forats per la superfície del flotador. Els forats seran de 7.2 cm de diàmetre. El nombre de forats a fer serà de:

$$v_4 = 26.49 - 23.68 = 2.81 \text{ dm}^2 \quad (\text{Eq. 86})$$

$$v_f = (\pi \cdot 3.6^2) \cdot 7 = 284.9 \text{ cm}^2 = 0.2849 \text{ dm}^2 \quad (\text{Eq. 87})$$

$$nf = \frac{2.81}{0.2849} = 9.86 \quad (\text{Eq. 88})$$

S'opta per posar 10 forats repartits uniformement. La posició dels forats es representa en el plànol corresponent.

A.3 MANUAL DE MANTENIMENT

A continuació s'enumeren alguns consells per a efectuar les operacions de manteniment del vehicle i propulsors.

Abans que res, és convenient desconnectar l'alimentació del vehicle abans de realitzar operacions de manteniment.

A.3.1 OPERACIONS DE MANTENIMENT PER ELS PROPULSORS

Verificar cada vegada abans d'utilitzar el vehicle l'estat de l'hèlix de cada propulsor. Verificar l'estat de les pales i l'absència de joc entre l'hèlix i l'eix. Verificar el correcte ajustament del cargol que subjecta el topall frontal.

En cas d'apreciar danys en l'hèlix aquesta ha de ser substituïda. El manual dels propulsors indica la manera correcta de fer aquesta operació.

EINES, MATERIALS I RECANVIS NECESSARIS:

- Claus Allen (mides comuns)
- Oli lubricant (Aqua shield o similar)
- Segellant Loctite 222 o similar
- Hèlix del propulsor (Part number P01780)

A.3.2 OPERACIONS DE MANTENIMENT RECOMANADES EN EL VEHICLE

Abans de cada us cal:

Verificar la rigidesa de l'estructura. En cas d'apreciar moviment efectuar un reapretat general dels cargols

Verificar la rigidesa de l'acoblament entre estructura i motors. En cas de joc, realitzar un apretat general de cargols.

Verificar l'estat de les hèlix dels propulsors. En cas d'estar en mal estat procedir a substituir-les el més aviat possible. Veure Documentació tècnica del propulsor.

Verificar el gir dels propulsors fora de l'aigua

Després de cada ús cal:

Penjar el vehicle i deixar assecar el conjunt. Esbandir amb aigua dolça en cas d'aigua salada o molt clorada.

Realitzar una inspecció ocular general del vehicle. En cas d'haver-hi elements estranys com algues, sorres o residus netejar.

Trimestralment cal:

Efectuat un apretat general de cargols. Parell recomanat 12Nm per els cargols de l'estructura. 8 Nm per cargols caixes elèctriques.

Ajustar la posició dels peus de l'estructura.

Verificació de l'estanquitat de la caixa elèctrica. Obrir la caixa amb el vehicle sec i observar una possible presència d'aigua. Revisió de la junta de la tapa.

Substitució de tots els elements que presentin corrosió visible.

A.4 CODI INFORMÀTIC

A continuació s'exposa el codi informàtic de la interfície gràfica. Aquest codi es troba en el format adequat (font i binari) en el CD annex.

El codi es compon de 4 arxius:

Pantalla.cws : Arxiu de projecte

Pantalla.uir : conté les pantalles i els seus elements

Pantalla.h : Capçaleres automàtiques

Pantalla.c : Conté el codi de les pantalles.

A.4.1 PANTALLA-H. CAPÇALERES

```

/*****
/* LabWindows/CVI User Interface Resource (UIR) Include File          */
/* Copyright (c) National Instruments 2006. All Rights Reserved.    */
/*                                                                    */
/* WARNING: Do not add to, delete from, or otherwise modify the contents */
/*       of this include file.                                         */
*****/

#include <userint.h>

#ifdef __cplusplus
extern "C" {
#endif

    /* Panels and Controls: */

#define PANTALLA 1
#define PANTALLA_TAB 2 /* callback function: TAB_CHANGE */
#define PANTALLA_B_SORTIR 3 /* callback function: Sortir */
#define PANTALLA_SPLITTER 4
#define PANTALLA_TEXTRESPOTES 5
#define PANTALLA_C_PORT 6 /* callback function: CANVI_PORT */
#define PANTALLA_L_PORT 7
#define PANTALLA_B_PORT 8 /* callback function: OBRIR_PORT */
#define PANTALLA_T_REFRESCA 9 /* callback function: Refresca */
#define PANTALLA_B_GRAPH 10 /* callback function: GRAPH */
#define PANTALLA_B_NetejaTXT 11 /* callback function: NetejaTXT */
#define PANTALLA_B_NetejaGR 12 /* callback function: NetejaGR */
#define PANTALLA_B_Grabar 13 /* callback function: Grabar */
#define PANTALLA_S_NARXIU 14
#define PANTALLA_STRIPCHART 15

    /* tab page panel controls */
#define TABPANEL_N_VELOCITAT 2
#define TABPANEL_N_AMPERIMETRE 3
#define TABPANEL_DECORATION 4
#define TABPANEL_DECORATION_2 5
#define TABPANEL_S_POT_INDIVIDUAL 6
#define TABPANEL_B_STOP_INDIVIDUAL 7 /* callback function: STOP */
#define TABPANEL_B_ENABLE_MOT_IND 8 /* callback function: ENNABLE_MOTORS */
#define TABPANEL_L_ENABLE_MOTOR_IND 9
#define TABPANEL_B_VERSIO 10 /* callback function: REQ_VERSIO */
#define TABPANEL_B_RESET 11 /* callback function: REQ_RESET */
#define TABPANEL_B_GRABAR_EEPROM 12 /* callback function: REQ_GRABAR_EEPROM */
#define TABPANEL_B_NOM 13 /* callback function: REQ_NOM */
#define TABPANEL_B_TEMPS 14 /* callback function: REQ_TEMPS */
#define TABPANEL_B_BRATE 15 /* callback function: REQ_BRATE */
#define TABPANEL_B_EEPROM 16 /* callback function: ENABLE_EEPROM */
#define TABPANEL_L_EEPROM 17
#define TABPANEL_B_NODE_INDIVIDUAL 18 /* callback function: CANVI_ADR */
#define TABPANEL_C_NODE 19
#define TABPANEL_TEXTMSG 20
#define TABPANEL_C_BRATE 21
#define TABPANEL_DECORATION_3 22
#define TABPANEL_DECORATION_4 23

    /* tab page panel controls */
#define TABPANEL_2_SPLITTER 2
#define TABPANEL_2_S_MOT1 3 /* callback function: CH_M */
#define TABPANEL_2_S_MOT2 4
#define TABPANEL_2_S_MOT3 5 /* callback function: CH_M */
#define TABPANEL_2_S_MOT4 6
#define TABPANEL_2_S_MOT5 7
#define TABPANEL_2_B_STOP_CONJUNT 8 /* callback function: STOP */
#define TABPANEL_2_C_RECT 9 /* callback function: CH_RECT */
#define TABPANEL_2_B_ENABLE_FB 10 /* callback function: EnableFB */
#define TABPANEL_2_L_ENABLE_FB 11
#define TABPANEL_2_B_ENABLE_MOT_CONJ 12 /* callback function: ENNABLE_MOT_CONJ */
#define TABPANEL_2_L_ENABLE_MOTOR_CONJ 13
#define TABPANEL_2_T_RPM5 14
#define TABPANEL_2_T_RPM4 15
#define TABPANEL_2_T_RPM3 16

```

```

#define TABPANEL_2_T_RPM1          17
#define TABPANEL_2_T_RPM2          18

/* tab page panel controls */
#define TABPANEL_3_D2                2
#define TABPANEL_3_N_OFFSETV         3
#define TABPANEL_3_N_DG              4
#define TABPANEL_3_D3                5
#define TABPANEL_3_N_FV              6
#define TABPANEL_3_D4                7
#define TABPANEL_3_N_FH              8
#define TABPANEL_3_D_DIRECCIO        9
#define TABPANEL_3_V_AVE             10 /* callback function: AVE */
#define TABPANEL_3_B_AV              11 /* callback function: AV */
#define TABPANEL_3_B_AVD             12 /* callback function: AVD */
#define TABPANEL_3_B_ES              13 /* callback function: ES */
#define TABPANEL_3_B_NOP             14 /* callback function: NOP */
#define TABPANEL_3_B_DR              15 /* callback function: DR */
#define TABPANEL_3_B_REE             16 /* callback function: REE */
#define TABPANEL_3_B_RE              17 /* callback function: RE */
#define TABPANEL_3_B_RED             18 /* callback function: RED */
#define TABPANEL_3_B_GE              19 /* callback function: GE */
#define TABPANEL_3_B_ASC             20 /* callback function: ASC */
#define TABPANEL_3_B_DES             21 /* callback function: DES */
#define TABPANEL_3_B_GD              22 /* callback function: GD */
#define TABPANEL_3_B_ROT             23 /* callback function: ROTE */
#define TABPANEL_3_B_ROT             24 /* callback function: ROTD */
#define TABPANEL_3_B_STOP_CONTROLAT  25 /* callback function: STOP */
#define TABPANEL_3_B_MOU             26 /* callback function: MouControlat */
#define TABPANEL_3_PICTURE           27
#define TABPANEL_3_B_CONFIGF         28 /* callback function: ConfigF */
#define TABPANEL_3_D1                29

/* tab page panel controls */
#define TABPANEL_4_TEXTMSG           2
#define TABPANEL_4_TEXTMSG_2         3
#define TABPANEL_4_TEXTMSG_4         4
#define TABPANEL_4_TEXTMSG_3         5
#define TABPANEL_4_PICTURE           6
#define TABPANEL_4_TEXTMSG_6         7
#define TABPANEL_4_TEXTMSG_7         8
#define TABPANEL_4_TEXTMSG_8         9

/* Menu Bars, Menus, and Menu Items: */

/* (no menu bars in the resource file) */

/* Callback Prototypes: */
int CVICALLBACK ASC(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK AV(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK AVD(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK AVE(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK CANVI_ADR(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK CANVI_PORT(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK CH_M(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK CH_RECT(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK ConfigF(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK DES(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK DR(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK ENABLE_EEPROMW(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK EnableFB(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);

```



```
int CVICALLBACK ENNABLE_MOT_CONJ(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK ENNABLE_MOTORS(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK ES(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK GD(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK GE(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK Grabar(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK GRAPH(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK MouControlat(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK NetejaGR(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK NetejaTXT(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK NOP(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK OBRIR_PORT(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK RE(int panel, int control, int event, void *callbackData, int eventData1, int
eventData2);
int CVICALLBACK RED(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK REE(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK Refresca(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_BRATE(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_GRABAR_EEPROM(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_NOM(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_RESET(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_TEMPS(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK REQ_VERSIO(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);
int CVICALLBACK ROTD(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK ROTE(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK Sortir(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK STOP(int panel, int control, int event, void *callbackData, int eventData1,
int eventData2);
int CVICALLBACK TAB_CHANGE(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2);

#ifdef __cplusplus
}
#endif
```

A.4.2 PANTALLA-C. CODI PANTALLES

```
//-----
//
//      -INTERFICIE DE CONTROL BANC DE PROVES I VEHICLE-
//
//PROJECTE: INTEGRACIÓ I ESTUDI D'UN CONJUT DE PROPULSORS PER A UN ROBOT SUBMARÍ
//AUTOR:   ALBERT SOLER
//DATA:    20/03/2008
//
//-----

//-----INCLUDE-----

#include <formatio.h>
#include <ansi_c.h>
#include <utility.h>
#include <rs232.h>
#include <cvirte.h>
#include <userint.h>
#include "Pantalla.h"
#include "toolbox.h"

//-----DEFINE & GLOBALS-----

#define SETS      1
#define REQS     2
#define REQC    3
#define REQB    4
#define REQT    5
#define REQN    6
#define REQVER  7
#define SETEEPROM 8
#define RESET   9
#define SETADR  10
#define SETBR   11
#define BR57600 57600
#define BR38400 38400
#define BR19200 19200
#define BR9600  9600
#define BR4800  4800

#define N_MOSTRES 5000
#define TEMPS     0 // en ms
#define CONSIGNAM1 1 // en %
#define VELOCITATM1 2 // en RPM
#define INTM1     3 // en A
#define CONSIGNAM2 4
#define VELOCITATM2 5
#define INTM2     6
#define CONSIGNAM3 7
#define VELOCITATM3 8
#define INTM3     9
#define CONSIGNAM4 10
#define VELOCITATM4 11
#define INTM4    12
#define CONSIGNAM5 13
#define VELOCITATM5 14
#define INTM5    15

#define CONSIGNA_MAX 90
#define CONSIGNA_MIN -90

#define M1 49
#define M2 50
#define M3 51
#define M4 52
#define M5 53
```

```

static int panelHandle;
static int Tab[4]; // conté els panelhandle de les 4 pestanyes

int ADRESSA;
int N_PORT;
char STR_PORT[5];

double Angle;
double Modul;
double MVert, Rotacio;
double TempsIni, TempsAct;

int Lectures [16][N_MOSTRES]; // temps + 5 motors x 3 valors = 16
int NumLectures;

//-----DEFINICIONS FUNCIONS NO CALLBACK-----

int GestComanda (int Comanda, int Parametre, int Node);
void GestGrafic();
int C2F(double Forca);

//-----MAIN I EXIT-----

int main (int argc, char *argv[])
{
    int c;

    ADRESSA=1;
    N_PORT=1;
    NumLectures=0;

    if (InitCVIRTE (0, argv, 0) == 0)
        return -1; /* out of memory */
    if ((panelHandle = LoadPanel (0, "Pantalla.uir", PANTALLA)) < 0)
        return -1;

    for (c=0;c<4;c++){
        GetPanelHandleFromTabPage (PANTALLA, PANTALLA_TAB, c, &Tab[c]);
    }

    DisplayPanel (panelHandle);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    return 0;
}

int CVICALLBACK Sortir (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            CloseCom (N_PORT);
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```

```

//-----FUNCIONS COMUNS-----

int CVICALLBACK STOP (int panel, int control, int event,
                      void *callbackData, int eventData1, int eventData2)
{ // Serveix per a fer una autrada ràpida de tots els motors
  switch (event)
  {
    case EVENT_COMMIT:
      // En primer lloc s'envia senyal de parar a tots els nodes per a evitat retard
      GestComanda(SETS, 0,M1);
      GestComanda(SETS, 0,M2);
      GestComanda(SETS, 0,M3);
      GestComanda(SETS, 0,M4);
      GestComanda(SETS, 0,M5);

      // A continuació es posen tots els controls a 0
      SetCtrlVal(Tab[0],TABPANEL_S_POT_INDIVIDUAL,0);

      SetCtrlVal(Tab[1],TABPANEL_2_S_MOT1,0);
      SetCtrlVal(Tab[1],TABPANEL_2_S_MOT2,0);
      SetCtrlVal(Tab[1],TABPANEL_2_S_MOT3,0);
      SetCtrlVal(Tab[1],TABPANEL_2_S_MOT4,0);
      SetCtrlVal(Tab[1],TABPANEL_2_S_MOT5,0);

      Modul=0;
      break;
  }
  return 0;
}

int CVICALLBACK OBRIR_PORT (int panel, int control, int event,
                             void *callbackData, int eventData1, int eventData2)
{ // Obre o tanca el port en funció de l'estat del switch.
  // A mes, activa o desactiva els controls qua no han d'esta activats amb el port obert
  int estat;
  switch (event)
  {
    case EVENT_COMMIT:
      estat=0;
      GetCtrlVal(PANTALLA,PANTALLA_B_PORT,&estat);
      switch(estat)
      {
        case 0: // TANCAR PORT
          if (CloseCom (N_PORT)<0)
          {
            MessagePopup("Error","No s'ha pogut tancar el port");
          }
          else
          {
            SetCtrlAttribute(PANTALLA,PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            //DESHABILITEM ELS CONTROLS NECESSARIS PANTALLA INDIVIDUAL

            SetCtrlVal(PANTALLA, PANTALLA_L_PORT,0);
            SetCtrlVal(Tab[0], TABPANEL_B_ENABLE_MOT_IND,0);
            SetCtrlAttribute(Tab[0], TABPANEL_B_ENABLE_MOT_IND,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_STOP_INDIVIDUAL,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_S_POT_INDIVIDUAL,ATTR_DIMMED,1);
            SetCtrlVal(Tab[0], TABPANEL_S_POT_INDIVIDUAL,0);

            SetCtrlAttribute(Tab[0], TABPANEL_B_BRATE,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_TEMPS,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_NOM,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_VERSIO,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_RESET,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_B_EEPROM,ATTR_DIMMED,1);

            SetCtrlVal(Tab[0], TABPANEL_B_EEPROM,0);
            SetCtrlVal(Tab[0], TABPANEL_L_EEPROM,0);

            SetCtrlAttribute(Tab[0], TABPANEL_C_BRATE,ATTR_DIMMED,1);
            SetCtrlAttribute(Tab[0], TABPANEL_C_NODE,ATTR_DIMMED,1);
          }
        }
      }
}

```

```

SetCtrlAttribute(Tab[0], TABPANEL_L_EEPROM,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[0], TABPANEL_B_RESET,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[0], TABPANEL_B_EEPROM,ATTR_DIMMED,1);

SetCtrlAttribute(PANTALLA, PANTALLA_C_PORT,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_GRABAR_EEPROM,ATTR_DIMMED,1);
SetCtrlAttribute(PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);

SetCtrlVal(Tab[0], TABPANEL_L_ENABLE_MOTOR_IND,0);
SetCtrlAttribute(Tab[0], TABPANEL_L_ENABLE_MOTOR_IND,ATTR_DIMMED,1);

//DESHABILITEM ELS CONTROLS NECESSARIS PANTALLA CONJUNT

SetCtrlVal(Tab[1], TABPANEL_2_B_ENABLE_MOT_CONJ,0);
SetCtrlAttribute(Tab[1], TABPANEL_2_B_ENABLE_MOT_CONJ,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT1,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT2,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT3,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT4,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT5,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_B_STOP_CONJUNT,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_L_ENABLE_MOTOR_CONJ,ATTR_DIMMED,1);
SetCtrlVal(Tab[1],TABPANEL_2_L_ENABLE_MOTOR_CONJ,0);
SetCtrlAttribute(Tab[1], TABPANEL_2_C_RECT ,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_B_ENABLE_FB ,ATTR_DIMMED,1);
SetCtrlAttribute(Tab[1], TABPANEL_2_L_ENABLE_FB ,ATTR_DIMMED,1);

//DESHABILITEM ELS CONTROLS NECESSARIS PANTALLA CONTROLAT
SetCtrlAttribute(Tab[2], TABPANEL_3_B_STOP_CONTROLAT,ATTR_DIMMED,1);
}
break;

case 1: //OBRIR PORT
if (OpenComConfig (N_PORT, STR_PORT, 57600, 0,8,1,30,30)<0)
{
MessagePopup("Error","No s'ha pogut obrir el port");
SetCtrlVal(PANTALLA, PANTALLA_B_PORT,0);
}
else
{
SetCtrlAttribute(PANTALLA,PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
//HABILITEM ELS CONTROLS NECESSARIS A PANTALLA INDIVIDUAL

SetCtrlVal(PANTALLA, PANTALLA_L_PORT,1);

SetCtrlAttribute(Tab[0], TABPANEL_B_ENABLE_MOT_IND,ATTR_DIMMED,0);

SetCtrlAttribute(Tab[0], TABPANEL_B_BRATE,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_TEMPS,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_NOM,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_VERSIO,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_RESET,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_B_EEPROM,ATTR_DIMMED,0);

SetCtrlAttribute(PANTALLA, PANTALLA_C_PORT,ATTR_DIMMED,1);

SetCtrlAttribute(PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);

SetCtrlAttribute(Tab[0], TABPANEL_L_EEPROM,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[0], TABPANEL_L_ENABLE_MOTOR_IND,ATTR_DIMMED,0);

//HABILITEM ELS CONTROLS NECESSARIS A PANTALLA CONJUNT

SetCtrlAttribute(Tab[1], TABPANEL_2_B_ENABLE_MOT_CONJ,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[1], TABPANEL_2_L_ENABLE_MOTOR_CONJ,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[1], TABPANEL_2_B_ENABLE_FB ,ATTR_DIMMED,0);
SetCtrlAttribute(Tab[1], TABPANEL_2_L_ENABLE_FB ,ATTR_DIMMED,0);

//HABILITEM ELS CONTROLS NECESSARIS A PANTALLA CONTROLAT
SetCtrlAttribute(Tab[2], TABPANEL_3_B_STOP_CONTROLAT,ATTR_DIMMED,0);
}
break;
}
break;

```

```

    }
    return 0;
}

int CVICALLBACK CANVI_PORT (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    // Actualitza la variable N_PORT que conté el prot actiu
    int tmp;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(PANTALLA,PANTALLA_C_PORT, &tmp);
            N_PORT=tmp;
            STR_PORT[3]=tmp+48;
            break;
    }
    return 0;
}

int GestComanda (int Comanda, int Parametre, int Node)
{
    //Gestiona una comanda. Envia el paquet i espera resposta
    //Comanda: Numero de la comanda
    //Parametre: PParametre en cas que la comanda en demani
    //Node: Node al que va adreçat la comanda (valor ASCII)

    char trama[15],resposta[15],bufin[4];
    int potencia,milers,centenes,desenes,unitats;
    int longtrama, tmp,c;

    trama[0]='u';
    trama[1]=Node;

    switch (Comanda){ // PRIMER ES PREPAREN LES TRAMES A ENVIAR

        case 1: //SETS
            potencia=Parametre;

            if (potencia==0){
                trama[4]='0';
                trama[2]='6';
                trama[6]=0;
            }
            if (potencia>=0){
                trama[3]='+';
            }
            else{
                trama[3]='-';
            }
            potencia=abs(potencia);

            if (potencia<10){
                trama[4]=potencia+48;
                trama[2]='6';
                trama[5]='\r';
                trama[6]=0;
                longtrama=6;
            }
            else if (potencia <100){
                desenes=potencia/10;
                unitats=potencia-desenes*10;
                trama[2]='7';
                trama[4]=desenes+48;
                trama[5]=unitats+48;
                trama[6]='\r';
                trama[7]=0;
                longtrama=7;
            }
            else {
                trama[2]='8';
                trama[4]='1';
                trama[5]='0';
                trama[6]='0';
                trama[7]='\r';
                trama[8]=0;
                longtrama=8;
            }
        }
    }
}

```

```
    }  
break;  
case 2://REQS  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='R';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 3://REQC  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='C';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 4://REQB  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='b';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 5://REQT  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='h';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 6://REQN  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='n';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 7://REQVER  
    trama[2]='6';  
    trama[3]='?';  
    trama[4]='v';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
break;  
case 8: //SETEEPROM  
    trama[2]='6';  
    trama[3]='x';  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;  
    if (Parametre==0){  
        trama[4]='0';  
    }  
    else{  
        trama[4]='1';  
    }  
break;  
case 9://RESET  
    trama[2]='5';  
    trama[3]='z';  
    trama[4]='\x';  
    trama[5]=0;  
    longtrama=5;  
break;  
case 10:  
    trama[2]='6';  
    trama[3]='n';  
    trama[4]=Parametre;  
    trama[5]='\x';  
    trama[6]=0;  
    longtrama=6;
```

```

break;
case 11://SETBR
    //0 = 57600
    //4 = 38400
    //3 = 19200
    //4 = 9600
    //1 = 4800
    trama[2]='6';
    trama[3]='b';
    trama[4]=Parametre+48;
    trama[5]='\x';
    trama[6]=0;
    longtrama=6;
break;
}

ComWrt (N_PORT, trama, longtrama); //S'escriu en el port

// CAPTURA DE RESPOSTA

for (c=0;c<=14;c++){
    resposta[c]=0; //inicialitzem la variable
}

if (Comanda==9){
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "EXECUTANT RESET");
}
else{
    tmp=GetInQLen (N_PORT);
    c=0;
    while (tmp==0 && c<5000){
        tmp=GetInQLen (N_PORT);
        c++;
    }
    bufin[0]=0;
    if (c<5000){
        c=0;
        while (tmp || bufin[0]!=13){
            bufin[0]=0;
            ComRd (N_PORT, bufin, 1);
            tmp=GetInQLen (N_PORT);
            resposta[c]=bufin[0];
            c++;
        }
    }
}

//GESTIÓ DE RESPOSTES

// verificacio de la trama
if (resposta[0]!='U'){
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}

if (resposta[1]!=trama[1]){
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació. Resposta
d'un altre node." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}

if (resposta[2]<=57 && resposta[2]>=52){ // chars de 4 a 9
    longtrama=resposta[2]-48;
}
else if (resposta[2]<=90 && resposta[2]>=65){ // chars de A a Z
    longtrama=resposta[2]-55;
}
else {
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació. Numero de
node incorrecte." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
}

```



```

    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}

if (resposta[longtrama-1]!=13){
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació. Longitud de
trama incorrecte." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}

//obtenció del parametre si n'hi ha i gestió de la resposta

switch (Comanda){
    case 1: //SETS
        if (resposta[2]=='4')
        {
            //InsertTextBoxLine (PRINCIPAL, PRINCIPAL_TEXTRESPOTES, -1, "SETS:ack");
        }
        else {
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació SETS:
Resposta no ACK." );
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
            return -1;
        }
        return 0;
    break;
    case 2: //REQS
        unitats=0;
        desenes=0;
        centenes=0;
        milers=0;
        c=0;

        if (resposta[4] != 45){ //si no es '-'
            if (longtrama==6){
                unitats=resposta[4]-48;
            }
            else if (longtrama==7){
                unitats=resposta[5]-48;
                desenes=resposta[4]-48;
            }
            else if (longtrama==8){
                unitats=resposta[6]-48;
                desenes=resposta[5]-48;
                centenes=resposta[4]-48;
            }
            else if (longtrama==9){
                unitats=resposta[7]-48;
                desenes=resposta[6]-48;
                centenes=resposta[5]-48;
                milers=resposta[4]-48;
            }
            else {
                InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació
REQS: Valor incoherent." );
                InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
                InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
                return -1;
            }
            c=unitats+desenes*10+centenes*100+milers*1000;
            c=c/5;
        }
        else { //si es '-'
            if (longtrama==7){
                unitats=resposta[5]-48;
            }
            else if (longtrama==8){
                unitats=resposta[6]-48;
                desenes=resposta[5]-48;
            }
            else if (longtrama==9){
                unitats=resposta[7]-48;
            }
        }
    }
}

```

```

        desenyes=resposta[6]-48;
        centenes=resposta[5]-48;
    }
    else if (longtrama==10){
        unitats=resposta[8]-48;
        desenyes=resposta[7]-48;
        centenes=resposta[6]-48;
        milers=resposta[5]-48;
    }
    else {
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació
REQS: Valor incoherent." );
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
        return -1;
    }
    c=(unitats+desenyes*10+centenes*100+milers*1000)*-1;
    c=c/5;
}
return c;

break;

case 3: //REQC
    //c=(resposta[4]-48);
    unitats=0;
    desenyes=0;
    centenes=0;
    milers=0;
    c=0;
    if (longtrama==5){
        unitats=resposta[3]-48;
    }
    else if (longtrama==6){
        unitats=resposta[4]-48;
        desenyes=resposta[3]-48;
    }
    else if (longtrama==7){
        unitats=resposta[5]-48;
        desenyes=resposta[4]-48;
        centenes=resposta[3]-48;
    }
    else if (longtrama==8){
        unitats=resposta[6]-48;
        desenyes=resposta[5]-48;
        centenes=resposta[4]-48;
        milers=resposta[3]-48;
    }
    else {
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació
REC: Valor incoherent." );
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
        //return -1;
    }
    c=(unitats+desenyes*10+centenes*100+milers*1000)*-1;
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Corrent es:"); //un cop
comprovat, eliminar això
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);

    return c;

break;

case 4: //REQB
    for (c=0;c<7;c++){
        trama[c]=resposta[c+3];
    }
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Baud rate actual:");
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    return 0;
break;

case 5: //REQT
    for (c=0;c<11;c++){
        trama[c]=resposta[c+3];

```

```

    }
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Temps de funcionament:");
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    return 0;
break;

case 6://REQN
for (c=0;c<11;c++){
    trama[c]=resposta[c+3];
}
InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Nom del node:");
InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
return 0;
break;

case 7: //REQVER
for (c=0;c<11;c++){
    trama[c]=resposta[c+3];
}
InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Versió:");
InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
return 0;
break;

case 8: //SETEEPROM
if (resposta[2]=='4')
{
    if (Parametre==0){
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Esriptura EEPROM
desactivada");
    }
    else{
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Esriptura EEPROM
activada");
    }
}
else {
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació
SETEEPROM: Valor incoherent." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}
return 0;
break;

case 9: //RESET
if (resposta[2]=='4')
{
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "RESET OK");
}
else {
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació RESET:
Valor incoherent." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}
return 0;
break;

case 10://SETADR
if (resposta[2]=='4')
{
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "GRABAR ADR: OK");
}
else {
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació GRABAR
EEPROM ADR: Valor incoherent." );
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
    InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
    return -1;
}
return 0;
break;

```

```

    case 11://SETBR
        if (resposta[2]=='4' && resposta[0]=='U')
        {
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "GRABAR EEPROM: OK");
        }
        else {
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "Error comunicació GRABAR
EEPROM BR: Valor incoherent." );
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, trama);
            InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, resposta);
            return -1;
        }
        return 0;
    }
    break;
}
return -1;
}

int CVICALLBACK TAB_CHANGE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //En cas de canviar de TAB s'aturen els motors
    int TabActiu;
    switch (event)
    {
        case EVENT_ACTIVE_TAB_CHANGE:
            SetCtrlVal(Tab[0],TABPANEL_S_POT_INDIVIDUAL,0);

            SetCtrlVal(Tab[1],TABPANEL_2_S_MOT1,0);
            SetCtrlVal(Tab[1],TABPANEL_2_S_MOT2,0);
            SetCtrlVal(Tab[1],TABPANEL_2_S_MOT3,0);
            SetCtrlVal(Tab[1],TABPANEL_2_S_MOT4,0);
            SetCtrlVal(Tab[1],TABPANEL_2_S_MOT5,0);

            Modul=0;
            break;
    }
    return 0;
}

int C2F(double Forca)
{
    //Calcula la consigna necessaria per a obrenir una força concreta (en llaç obert)
    int consigna,signe;
    double temp;

    if (Forca<-1.5969){
        temp=(-0.156-sqrt(-0.0108*Forca))/0.0054392;
        signe=-1;
    }
    else if (Forca>=-1.5969 && Forca<0){
        temp=sqrt(Forca/-0.000575);
        signe=-1;
    }
    else if (Forca>=0 && Forca<1.8326){
        temp=sqrt(Forca/0.0006374);
        signe=1;
    }
    else {
        temp=(0.177+sqrt(0.01218*Forca))/0.006088;
        signe=1;
    }
    consigna=abs(temp)*signe;
    return consigna;
}

int CVICALLBACK GRAPH (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Activa i desactiva el gràfic.
    //En cas d'activar-lo es guarda el temps inicial
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(PANTALLA,PANTALLA_B_GRAPH, &estat);
            switch(estat){

```

```

        case 1:
            SetCtrlAttribute (PANTALLA,PANTALLA_STRIPCHART, ATTR_DIMMED, 0);
            GetCurrentDateTime (&TempsIni);
            break;
        case 0:
            SetCtrlAttribute (PANTALLA,PANTALLA_STRIPCHART, ATTR_DIMMED, 1);
            break;
    }
    break;
}
return 0;
}

int CVICALLBACK NetejaTXT (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Neteja el textbox de missatges
    switch (event)
    {
        case EVENT_COMMIT:
            ResetTextBox (PANTALLA, PANTALLA_TEXTRESPOTES, "");

            break;
    }
    return 0;
}

int CVICALLBACK NetejaGR (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Neteja el gràfic i reseteja el temps
    switch (event)
    {
        case EVENT_COMMIT:

            NumLectures=0;
            ClearStripChart (PANTALLA,PANTALLA_STRIPCHART );
            GetCurrentDateTime (&TempsIni);
            break;
    }
    return 0;
}

int CVICALLBACK Grabar (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Graba en un arxiu les dades recollides
    int c,d,FileHandle;
    char NomArxiu[14],dada[12];
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(PANTALLA,PANTALLA_S_NARXIU,&NomArxiu[0]);
            FileHandle=OpenFile (NomArxiu, VAL_WRITE_ONLY, VAL_TRUNCATE, VAL_ASCII);

            for (c=0; c<9;c++){
                dada[c]=' ';
            }

            for (c=1; c<NumLectures;c++){
                for (d=0;d<16;d++){
                    Fmt (dada, "%d ",Lectures[d][c]);

                    WriteFile (FileHandle, dada, 8);
                    dada[0]=' ';
                    WriteFile (FileHandle, dada, 1);
                }
                dada[0]='\n';
                WriteFile (FileHandle, dada, 1);
            }
            CloseFile (FileHandle);

            break;
    }
    return 0;
}
}

```

```

//-----FUNCIONS CONTROL INDIVIDUAL-----
int CVICALLBACK ENNABLE_MOTORS (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Activa els controls del motor
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            estat=0;
            GetCtrlVal(Tab[0],TABPANEL_B_ENABLE_MOT_IND,&estat);

            switch(estat)
            {
                case 0:
                    SetCtrlAttribute (Tab[0],TABPANEL_B_STOP_INDIVIDUAL,ATTR_DIMMED,1);
                    SetCtrlAttribute (Tab[0],TABPANEL_S_POT_INDIVIDUAL,ATTR_DIMMED,1);
                    SetCtrlVal(Tab[0],TABPANEL_S_POT_INDIVIDUAL,0);
                    SetCtrlVal(Tab[0],TABPANEL_L_ENABLE_MOTOR_IND,0);
                    break;
                case 1:
                    SetCtrlAttribute (Tab[0],TABPANEL_B_STOP_INDIVIDUAL,ATTR_DIMMED,0);
                    SetCtrlAttribute (Tab[0],TABPANEL_S_POT_INDIVIDUAL,ATTR_DIMMED,0);
                    SetCtrlVal(Tab[0],TABPANEL_S_POT_INDIVIDUAL,0);
                    SetCtrlVal(Tab[0],TABPANEL_L_ENABLE_MOTOR_IND,1);
                    break;
            }
            break;
    }
    return 0;
}

int CVICALLBACK REQ_GRABAR_EEPROM (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    //Graba
    char tmp;
    int BRate;
    int NodeAdr;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (Tab[0],TABPANEL_C_BRATE,&tmp);
            BRate=tmp;
            GestComanda (SETBR, BRate, ADRESSA);
            GetCtrlVal (Tab[0],TABPANEL_C_NODE,&tmp);
            NodeAdr=tmp;
            GestComanda (SETADR, NodeAdr, ADRESSA);

            break;
    }
    return 0;
}

int CVICALLBACK REQ_RESET (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
            GestComanda(RESET,0,ADRESSA);
            Delay (2.50);
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            break;
    }
    return 0;
}

```

```
}

int CVICALLBACK REQ_VERSIO (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
            GetCtrlVal(Tab[0],TABPANEL_B_NODE_INDIVIDUAL,&ADRESSA);
            GestComanda(REQVER,0,ADRESSA);

            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            break;
    }
    return 0;
}

int CVICALLBACK REQ_NOM (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
            GestComanda(REQN,0,ADRESSA);

            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            break;
    }
    return 0;
}

int CVICALLBACK REQ_TEMPS (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
            GestComanda(REQT,0,ADRESSA);

            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            break;
    }
    return 0;
}

int CVICALLBACK REQ_BRATE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,0);
            GestComanda(REQB,0,ADRESSA);

            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
            break;
    }
    return 0;
}

int CVICALLBACK CANVI_ADR (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(Tab[0],TABPANEL_B_NODE_INDIVIDUAL, &tmp);
            ADRESSA=tmp;
            break;
    }
}
```

```

    return 0;
}

int CVICALLBACK ENABLE_EEPROMW (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(Tab[0], TABPANEL_B_EEPROM,&estat);
            switch (estat)
            {
                case 0:
                    SetCtrlVal(Tab[0], TABPANEL_L_EEPROM,0);
                    SetCtrlAttribute(Tab[0], TABPANEL_C_BRATE,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[0], TABPANEL_C_NODE,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[0], TABPANEL_B_GRABAR_EEPROM,ATTR_DIMMED,1);

                    break;
                case 1:
                    SetCtrlVal(Tab[0], TABPANEL_L_EEPROM,1);
                    SetCtrlAttribute(Tab[0], TABPANEL_C_BRATE,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[0], TABPANEL_C_NODE,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[0], TABPANEL_B_GRABAR_EEPROM,ATTR_DIMMED,0);

                    break;
            }
        break;
    }
    return 0;
}

void GestGrafic(){
    int Lectura[16];
    int c,TabAct;
    char Node;

    if (NumLectures<N_MOSTRES-1){

        GetCurrentDateTime (&TempsAct);
        Lectures[TEMPS][NumLectures]=abs((TempsAct-TempsIni)*1000);

        GetCtrlAttribute (PANTALLA, PANTALLA_TAB, ATTR_CTRL_INDEX, &TabAct);
        switch (TabAct)
        {
            case 0:
                GetCtrlVal (Tab[0], TABPANEL_S_POT_INDIVIDUAL , &Lectures[1][NumLectures]);
                GetCtrlVal (Tab[0], TABPANEL_N_VELOCITAT , &Lectures[2][NumLectures]);
                GetCtrlVal (Tab[0], TABPANEL_N_AMPERIMETRE , &Lectures[3][NumLectures]);
                GetCtrlVal (Tab[0], TABPANEL_B_NODE_INDIVIDUAL , &Node);
                Lectures[4][NumLectures]=Node;
            break;
            case 1:
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT1 , &Lectures[CONSIGNAM1][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT2 , &Lectures[CONSIGNAM2][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT3 , &Lectures[CONSIGNAM3][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT4 , &Lectures[CONSIGNAM4][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT5 , &Lectures[CONSIGNAM5][NumLectures]);

                GetCtrlVal (Tab[1], TABPANEL_2_T_RPM1 , &Lectures[VELOCITATM1][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_T_RPM2 , &Lectures[VELOCITATM2][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_T_RPM3 , &Lectures[VELOCITATM3][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_T_RPM4 , &Lectures[VELOCITATM4][NumLectures]);
                GetCtrlVal (Tab[1], TABPANEL_2_T_RPM5 , &Lectures[VELOCITATM5][NumLectures]);
            break;
        }

        NumLectures++;

        for (c=0;c<16;c++){
            Lectura[c]=Lectures[c][NumLectures];
        }

        PlotStripChart (PANTALLA, PANTALLA_STRIPCHART , Lectura, 16, 1, 1, VAL_INTEGER);
    }
}

```



```

    }
    else {
        InsertTextBoxLine (PANTALLA, PANTALLA_TEXTRESPOTES, -1, "BUFFER LECTURES DESBORDAT. No
es guardaran mes dades");
    }
}

int CVICALLBACK Refresca (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int Consigna[5], TabAct, Velocitat, Intensitat, EstatGrafic, EstatFB, ExcesConsigna;
    char Node;
    double x, y, OffsetV;

    switch (event)
    {
        case EVENT_TIMER_TICK:
            SetCtrlAttribute (PANTALLA, PANTALLA_T_REFRESCA, ATTR_ENABLED, 0);
            GetCtrlAttribute (PANTALLA, PANTALLA_TAB, ATTR_CTRL_INDEX, &TabAct);
            switch (TabAct)
            {
                case 0: //si pantalla individual activada
                    GetCtrlVal (Tab[0], TABPANEL_S_POT_INDIVIDUAL, &Consigna[0]);
                    GetCtrlVal (Tab[0], TABPANEL_B_NODE_INDIVIDUAL, &Node);
                    //Num_Node=Node-48;
                    GestComanda(SETS, Consigna[0], Node); //Num_Node);
                    Velocitat=GestComanda(REQS, 0, Node); //Num_Node);
                    SetCtrlVal (Tab[0], TABPANEL_N_VELOCITAT, Velocitat);
                    Intensitat=GestComanda(REQC, 0, Node); //Num_Node);
                    SetCtrlVal (Tab[0], TABPANEL_N_AMPERIMETRE, Intensitat);

                break;

                case 1: //si pantalla conjunt activada
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT1 , &Consigna[0]);
                    Consigna[0]=Consigna[0]*-1; //els motors estan muntats en sentit contrari
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT2 , &Consigna[1]);
                    Consigna[1]=Consigna[1]*-1; //els motors estan muntats en sentit contrari
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT3 , &Consigna[2]);
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT4 , &Consigna[3]);
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT5 , &Consigna[4]);

                    GestComanda(SETS, Consigna[0], M1);
                    GestComanda(SETS, Consigna[1], M2);
                    GestComanda(SETS, Consigna[2], M3);
                    GestComanda(SETS, Consigna[3], M4);
                    GestComanda(SETS, Consigna[4], M5);

                    GetCtrlVal (Tab[1], TABPANEL_2_B_ENABLE_FB , &EstatFB);
                    if (EstatFB) {
                        Velocitat=GestComanda(REQS, 0, M1);
                        SetCtrlVal(Tab[1], TABPANEL_2_T_RPM1, Velocitat);
                        Velocitat=GestComanda(REQS, 0, M2);
                        SetCtrlVal(Tab[1], TABPANEL_2_T_RPM2, Velocitat);
                        Velocitat=GestComanda(REQS, 0, M3);
                        SetCtrlVal(Tab[1], TABPANEL_2_T_RPM3, Velocitat);
                        Velocitat=GestComanda(REQS, 0, M4);
                        SetCtrlVal(Tab[1], TABPANEL_2_T_RPM4, Velocitat);
                        Velocitat=GestComanda(REQS, 0, M5);
                        SetCtrlVal(Tab[1], TABPANEL_2_T_RPM5, Velocitat); //invalid int expected
                    }

                break;

                case 2: //si pantalla controlat activada

                    // moviment horitzontal.

                    //OBTENIM components x i y
                    x=sin (Angle/360*TwoPi ())*Modul;
                    y=cos (Angle/360*TwoPi ())*Modul;

                    Consigna[0]=C2F(y/-2);
                    Consigna[1]=C2F(y/-2);
            }
    }
}

```

```

Consigna[2]=C2F(x/-2); //si va al revés multiplicar per -1
Consigna[3]=C2F(x/-2);

// rotació
Consigna[2]=Consigna[2]+C2F(Rotacio/2);
Consigna[3]=Consigna[3]+C2F(Rotacio/-2);

// verificar que consigna no passa de CONSIGNA_MAX
if (Consigna[2]>CONSIGNA_MAX ){
  ExcesConsigna=Consigna[2]-CONSIGNA_MAX;
  Consigna[2]=Consigna[2]-ExcesConsigna;
  Consigna[3]=Consigna[3]-ExcesConsigna;
}
if (Consigna[2]<CONSIGNA_MAX*-1 ){
  ExcesConsigna=Consigna[2]+CONSIGNA_MAX;
  Consigna[2]=Consigna[2]-ExcesConsigna;
  Consigna[3]=Consigna[3]-ExcesConsigna;
}
if (Consigna[3]>CONSIGNA_MAX ){
  ExcesConsigna=Consigna[3]-CONSIGNA_MAX;
  Consigna[2]=Consigna[2]-ExcesConsigna;
  Consigna[3]=Consigna[3]-ExcesConsigna;
}
if (Consigna[3]<CONSIGNA_MAX*-1 ){
  ExcesConsigna=Consigna[3]+CONSIGNA_MAX;
  Consigna[2]=Consigna[2]-ExcesConsigna;
  Consigna[3]=Consigna[3]-ExcesConsigna;
}

// moviment vertical.
GetCtrlVal (Tab[2], TABPANEL_3_N_OFFSETV , &OffsetV);

Consigna[4]=C2F(MVert+OffsetV);
if (Consigna[4]>CONSIGNA_MAX) Consigna[4]=CONSIGNA_MAX;
if (Consigna[4]<CONSIGNA_MIN) Consigna[4]=CONSIGNA_MIN;

GestComanda(SETS, Consigna[0],M1);
GestComanda(SETS, Consigna[1],M2);
GestComanda(SETS, Consigna[2],M3);
GestComanda(SETS, Consigna[3],M4);
GestComanda(SETS, Consigna[4],M5);

break;
case 3: //si pantalla info
  // no fem res
  break;
}

GetCtrlVal(PANTALLA,PANTALLA_B_GRAPH,&EstatGrafic);
if (EstatGrafic) {
  GestGrafic();
}

SetCtrlAttribute (PANTALLA,PANTALLA_T_REFRESCA,ATTR_ENABLED,1);
break;
}
return 0;
}

```

```

//-----FUNCIIONS CONJUNT-----
int CVICALLBACK ENNABLE_MOT_CONJ (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(Tab[1], TABPANEL_2_B_ENABLE_MOT_CONJ,&estat);
            switch (estat)
            {
                case 0:
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT1,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT2,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT3,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT4,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT5,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_B_STOP_CONJUNT,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_C_RECT ,ATTR_DIMMED,1);

                    SetCtrlVal(Tab[1],TABPANEL_2_S_MOT1,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_S_MOT2,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_S_MOT3,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_S_MOT4,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_S_MOT5,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_L_ENABLE_MOTOR_CONJ,0);

                    break;
                case 1:
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT1,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT2,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT3,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT4,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT5,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_B_STOP_CONJUNT,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_C_RECT ,ATTR_DIMMED,0);
                    SetCtrlVal(Tab[1],TABPANEL_2_L_ENABLE_MOTOR_CONJ,1);

                    break;
            }
        }
    }
    return 0;
}

int CVICALLBACK EnableFB (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(Tab[1],TABPANEL_2_B_ENABLE_FB,&estat);
            SetCtrlVal(Tab[1],TABPANEL_2_L_ENABLE_FB,estat);
            break;
    }
    return 0;
}

```

```
int CVICALLBACK CH_RECT (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int estat,tmp;

    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (Tab[1], TABPANEL_2_C_RECT, &estat);
            switch (estat)
            {
                case 0:
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT2,ATTR_DIMMED,0);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT4,ATTR_DIMMED,0);
                    break;
                case 1:
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT2,ATTR_DIMMED,1);
                    SetCtrlAttribute(Tab[1], TABPANEL_2_S_MOT4,ATTR_DIMMED,1);
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT1, &tmp);
                    SetCtrlVal (Tab[1], TABPANEL_2_S_MOT2, tmp);
                    GetCtrlVal (Tab[1], TABPANEL_2_S_MOT3, &tmp);
                    SetCtrlVal (Tab[1], TABPANEL_2_S_MOT4, tmp);

                    break;
            }
            break;
    }
    return 0;
}

int CVICALLBACK CH_M (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int tmp;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal (Tab[1], TABPANEL_2_C_RECT, &tmp);

            if (tmp) {
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT1, &tmp);
                SetCtrlVal (Tab[1], TABPANEL_2_S_MOT2, tmp);
                GetCtrlVal (Tab[1], TABPANEL_2_S_MOT3, &tmp);
                SetCtrlVal (Tab[1], TABPANEL_2_S_MOT4, tmp);
            }
            break;
    }
    return 0;
}
```

```
//-----FUNCIONS FUNCIONAMENT CONTROLAT-----
```

```
int CVICALLBACK AV (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=0;
            break;
    }
    return 0;
}

int CVICALLBACK RE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=180;
            break;
    }
    return 0;
}

int CVICALLBACK DR (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=90;
            break;
    }
    return 0;
}

int CVICALLBACK ES (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=270;
            break;
    }
    return 0;
}
```

```
int CVICALLBACK NOP (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
    }
    return 0;
}

int CVICALLBACK AVE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=315;
            break;
    }
    return 0;
}

int CVICALLBACK AVD (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=45;
            break;
    }
    return 0;
}

int CVICALLBACK REE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=225;
            break;
    }
    return 0;
}

int CVICALLBACK RED (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=135;
            break;
    }
    return 0;
}
```

```
int CVICALLBACK ASC (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            MVert=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FV,&MVert);
            break;
    }
    return 0;
}

int CVICALLBACK DES (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            MVert=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FV,&MVert);
            MVert=MVert*-1;
            break;
    }
    return 0;
}

int CVICALLBACK ROTE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Rotacio=0;

            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_DG,&Rotacio);

            break;
    }
    return 0;
}

int CVICALLBACK ROTD (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Rotacio=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_DG,&Rotacio);
            Rotacio=Rotacio*-1;
            break;
    }
    return 0;
}
```

```
int CVICALLBACK GE (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            Rotacio=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=0;
            GetCtrlVal(Tab[2],TABPANEL_3_N_DG,&Rotacio);
            break;
    }
    return 0;
}

int CVICALLBACK GD (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            Rotacio=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            Angle=0;
            GetCtrlVal(Tab[2],TABPANEL_3_N_DG,&Rotacio);
            Rotacio=Rotacio*-1;
            break;
    }
    return 0;
}

int CVICALLBACK MouControlat (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
        case EVENT_COMMIT:
            Modul=0;
            break;
        case EVENT_LEFT_CLICK:
            GetCtrlVal(Tab[2],TABPANEL_3_D_DIRECCIO,&Angle);
            GetCtrlVal(Tab[2],TABPANEL_3_N_FH,&Modul);
            break;
    }
    return 0;
}
```



```
int CVICALLBACK ConfigF (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    int estat;
    switch (event)
    {
        case EVENT_COMMIT:
            GetCtrlVal(Tab[2], TABPANEL_3_B_CONFIGF, &estat);
            switch (estat)
            {
                case 0:
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_DG, ATTR_DIMMED, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_FH, ATTR_DIMMED, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_FV, ATTR_DIMMED, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_OFFSETV, ATTR_DIMMED, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D1, ATTR_VISIBLE, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D2, ATTR_VISIBLE, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D3, ATTR_VISIBLE, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D4, ATTR_VISIBLE, 0);

                    break;
                case 1:
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_DG, ATTR_DIMMED, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_FH, ATTR_DIMMED, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_FV, ATTR_DIMMED, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_N_OFFSETV, ATTR_DIMMED, 0);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D1, ATTR_VISIBLE, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D2, ATTR_VISIBLE, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D3, ATTR_VISIBLE, 1);
                    SetCtrlAttribute (Tab[2], TABPANEL_3_D4, ATTR_VISIBLE, 1);

                    break;
            }
        break;
    }
    return 0;
}
```