

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials (GETI)

Programa de càlcul de molles helicoidals

MEMÒRIA

Autor: Andreu Punsola Soler

Director: Lluís Roger Casals

Convocatòria: Juny 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



ETSEIB

Resum

En aquest projecte s'ha desenvolupat un programa d'ordinador amb interfície gràfica per a calcular diferents paràmetres útils sobre molles helicoidals¹, així com la seva corba característica. Aquest programa està pensat per a l'ús en disseny de màquines, ajudant en el procés de dimensionament de molles helicoidals que formin part del sistema.

Per començar es descriu el funcionament de la mecànica de medis continus que regeix les molles helicoidals. A continuació s'expliquen què són les interfícies gràfiques d'usuari, com es desenvolupen i quina és l'estructura general del codi que les regeix. Finalment s'explica com, mesclant aquests dos camps, es crea un programa d'escriptori utilitzant *MATLAB® App Designer* per a assistir en el dimensionament de molles.

¹Des de l'Ortografia de l'IEC del 2016 helicoidal s'escriu sense dièries. Font: <https://www.encyclopedia.cat/EC-GDLC-e00072625.xml>

Índex

1	Prefaci	8
1.1	Origen del projecte	8
1.2	Requeriments previs	8
2	Introducció	9
2.1	Objectius del projecte	9
2.2	Estat de l'art	9
3	Molles helicoidals	11
3.1	Paràmetres geomètrics	12
3.2	Fabricació	12
3.3	Materials més utilitzats	14
3.4	Acabaments	15
3.5	Càlcul de molles	16
3.5.1	Constant de la molla	17
3.5.2	Deformació màxima	18
3.5.3	Càlcul de tensions	18
3.5.4	Altres càlculs	20
4	Interfície gràfica d'usuari (GUI)	22
4.1	Definició	22
4.2	Gestors d'esdeveniments	23
4.3	Entorn de desenvolupament	23
4.4	<i>MATLAB® App Designer</i>	24
5	Programa	27
5.1	Escenari gràfic	27
5.2	Funcionament	29
5.2.1	Inicialització	30
5.2.2	Bucle principal	31
5.3	Classe <code>Material</code>	33
5.4	Funcions auxiliars	33
5.4.1	Validació de l'entrada de dades de l'usuari	33
5.4.2	Obtenció dels paràmetres de la molla	35
5.4.3	Nombre d'espines inactives	36
5.5	Actualització de l'escenari gràfic	36
5.5.1	Funció <code>mostramolla</code>	36
5.5.2	Renderitzat de la molla	37
5.5.3	Funció <code>mostracorba</code>	38
5.5.4	Generació del gràfic de la corba de la molla	39
5.5.5	Emplenar les dades	41
5.6	Llançament del producte	43
6	Pressupost	47
6.1	Cost de personal	47
6.2	Cost de llicències	47
6.3	Cost total associat	47

7 Impacte ambiental	48
Conclusions	49
Agraïments	50
Annex	51
A Materials comuns per a molles	51
B Codi font	53
Bibliografia	61

Índex de figures

2.1 Interfície de l' <i>Spring Creator</i> . Font: <i>Access Springs</i> [2].	9
2.2 Resultats de l' <i>Spring Creator</i> . Font: <i>Access Springs</i> [2].	10
3.1 Diverses molles helicoidals. Font: assignatura Vehicles [18].	11
3.2 Diverses molles. Font: Google Images	11
3.3 Molla amb $n = 6$ espires. Font: pròpia.	12
3.4 Esquema del control numèric en una enrotlladora per a molles petites. Font: <i>Access Springs</i> [1].	13
3.5 Enrotlladora per a molles petites. Font: <i>Whitelegg</i> [22].	13
3.6 Enrotlladora de molles grosses. Font: <i>King Springs</i> [8].	14
3.7 Diferents acabaments de molles. Font: <i>Access Spring</i> [1].	15
3.8 Esquema de forces en una molla. Font: pròpia.	17
3.9 Estat de tensions. Font: pròpia	19
4.1 La interfície gràfica d'usuari del Xerox Star, el primer ordinador comercial que en va tenir. Font: Karl Voit [21].	22
4.2 Xcode, la plataforma de desenvolupament de programari per a iOS, macOS, watchOS i tvOS d'Apple. Font: Apple Developer [3].	24
4.3 Vista de disseny d'interfície del <i>MATLAB® App Designer</i> . Font: <i>MATLAB®</i> [11] .	25
4.4 Vista de codi del <i>MATLAB® App Designer</i> . Font: <i>MATLAB®</i> [11]	26
5.1 Elements de la interfície gràfica. Font: pròpia.	27
5.2 Aspecte del programa mostrant el renderitzat de la molla. Font: pròpia.	28
5.3 Aspecte del programa mostrant la corba força-desplaçament. Font: pròpia. . . .	29
5.4 Esquema del funcionament del programa. Font: pròpia	32
5.5 Exemple d'entrada errònia de l'usuari. Font: pròpia.	35
5.6 Renderitzats d'una molla amb $n = 14$, $r = 3$ mm, $R = 20$ mm i $p = 10$ mm. Font: pròpia.	38
5.7 Corba característica d'una molla helicoidal d'Inconel 600 amb $n = 13$, $d = 4$ mm, $R = 19$ mm i $p = 6$ mm. Font: pròpia.	41
5.8 Dades d'una molla helicoidal d'Inconel 600 amb $n = 13$, $d = 4$ mm, $R = 19$ mm i $p = 6$ mm. Font: pròpia.	43
5.9 Menú del <i>MATLAB® Application Compiler</i> . Font: pròpia.	44
5.10 Pantalla de presentació del programa. Font: pròpia.	45
5.11 Procés d'instal·lació del programa. Font: pròpia.	45

Índex de taules

6.1	Cost de personal	47
A.1	Propietats dels metalls més comuns per a molles	52

Glossari

Símbols

γ	El doble de la variació angular diferencial entre direccions perpendiculars canviada de signe. [rad]
δ	Desplaçament vertical de la molla helicoidal sota càrrega. [mm]
θ	Angle que gira el fil d'una barra a torsió en aplicar-hi un moment torsor. [rad]
ρ	Densitat. [kg/m ³]
σ	Tensió normal uniaxial. [MPa]
τ	Tensió tallant. [MPa]
ω	Pulsació pròpia d'un sistema. [rad/s]
A	Àrea que treballa a cisalladura. [m ²]
C	Relació d'enrotllament, quocient entre el radi de l'hèlix directriu d'una molla helicoidal i el radi del fil. No té unitats.
d	Gruix o diàmetre del fil d'una molla helicoidal. [mm]
E	Mòdul de Young, mesura de la rigidesa d'un material en la seva zona elàstica. Relació entre la tensió uniaxial i l'allargament unitari uniaxial. [GPa]
E_{pe}	Energia potencial elàstica, l'energia associada a deformar un sòlid elàstic. [J]
F	Força a la que es sotmet una molla helicoidal. [N]
f	Freqüència pròpia d'un sistema. [Hz]
G	Mòdul de cisallament o d'elasticitat transversal. Relació entre la tensió tallant i la variació angular. [GPa]
I_t	Inèrcia a torsió d'una secció d'un perfil. [m ⁴]
K_s	Factor corrector de l'esforç tallant. Calculat a partir de la relació d'enrotllament, permet trobar la tensió tallant causada per una força aplicada a una molla helicoidal a partir de la tensió tallant deguda al moment multiplicada per aquest factor. No té unitats.
k	Constant elàstica d'una molla helicoidal. Relació entre la força aplicada i el desplaçament vertical. [N/mm]
L	Longitud de la molla helicoidal. [mm]
L_{bloc}	Longitud de la molla helicoidal totalment comprimida, amb totes les espirals en contacte. [mm]
L_0	Longitud de la molla helicoidal sense cap càrrega aplicada. [mm]
M_t	Moment torsor aplicat a un perfil prismàtic. [N·m]

m	Massa [kg]
n	Nombre d'espines d'una molla helicoidal. No té unitats.
n_e	Nombre d'espines que treballen elàsticament d'una molla helicoidal. No té unitats.
p	Pas de l'hèlix. Distància entre dues espines consecutives. [mm]
R	Radi de l'hèlix directriu de la molla helicoidal. [mm]
r	Radi del fil d'una molla helicoidal. [mm]
T	Força de cisallament aplicada a un perfil prismàtic [N]

Termes

<i>auto coiler</i>	Auto enrotlladora; màquina que enrotlla automàticament filferro acotant uns paràmetres geomètrics desitjats.
<i>callback</i>	Crida de retorn; codi que passa com a argument en una funció.
<i>event handler</i>	Gestor d'esdeveniment; funció que gestiona els inputs rebuts per un programa.
<i>GUI</i>	Interfície gràfica d'usuari; mitjà de comunicació gràfic entre l'usuari i l'ordinador.
<i>IDE</i>	Entorn integrat de desenvolupament; programa per programar.
<i>slider</i>	Control lliscant que permet triar un nombre d'un rang.
<i>splash screen</i>	Pantalla que apareix quan un programa està carregant.
<i>string</i>	Seqüència de caràcters alfanumèrics amb un ordre i una llargada determinada.

1 Prefaci

1.1 Origen del projecte

Aquest projecte sorgeix de la proposta del professor Lluís Roger, director d'aquest treball.

1.2 Requeriments previs

Els requeriments previs per realitzar aquest treball són coneixements d'elasticitat i resistència de materials, i programació orientada a esdeveniments i habilitats de disseny d'interfícies gràfiques d'usuari.

2 Introducció

2.1 Objectius del projecte

L'objectiu principal és crear un programa d'escriptori amb interfície gràfica amb el qual es pugui obtenir la corba característica d'una molla dins del seu rang de funcionament. També s'obtin- dran paràmetres útils que es puguin calcular a partir de les dimensions de la molla i el material, que l'usuari introdueix.

Aquest programa està pensat per al seu ús en el disseny de màquines en general, per ajudar a l'hora de dimensionar les molles helicoidals (les més comunes), facilitant la feina en evitar haver de prototipar diferents combinacions de dimensions de la molla i de materials.

2.2 Estat de l'art

L'únic servei o programari similar a l'objecte d'aquest treball que s'ha trobat és a la pàgina web de l'empresa del sector de les molles *Access Spring* [2], que inclou un servei per ajudar el seu client potencial a l'hora d'escollir una molla helicoidal.

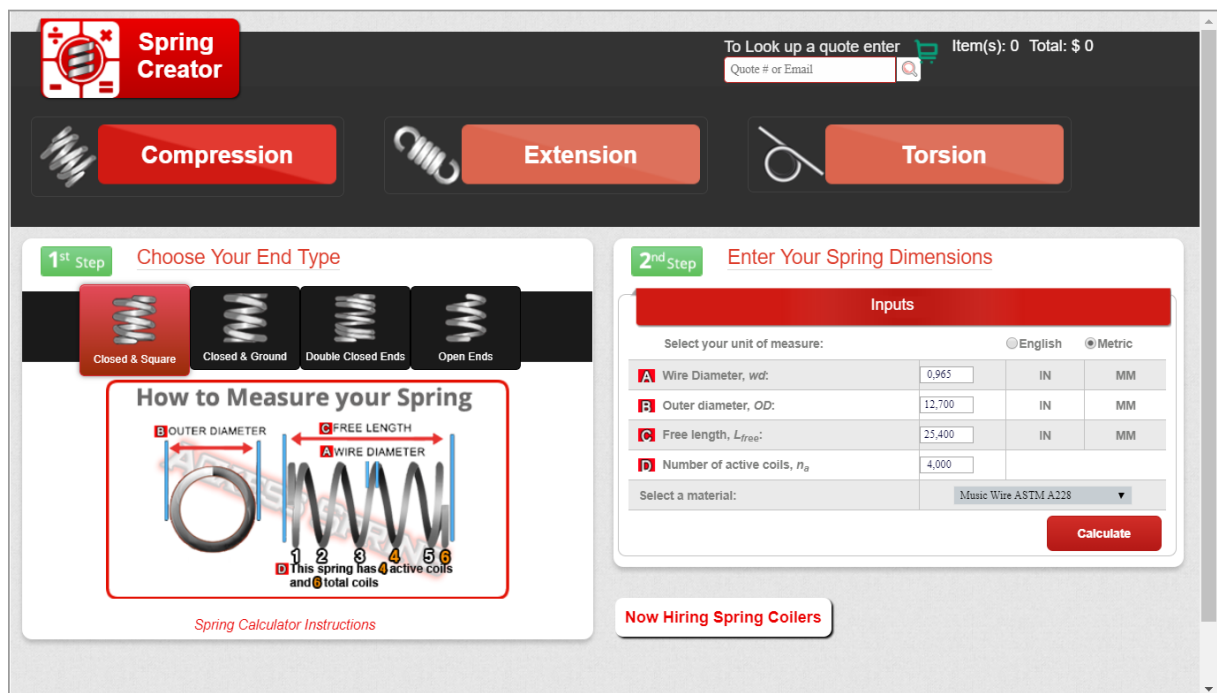


Figura 2.1: Interfície de l'*Spring Creator*. Font: *Access Springs* [2].

A la part esquerra (figura 2.1) es descriuen els acabaments de la molla (veure apartat 3.4) i a la part dreta apareixen el gruix del fil, el diàmetre exterior, la longitud sense càrrega i el nombre de voltes. Hi ha 11 metalls diferents per triar. En prémer el botó "Calculate", apareix just a sota el resultat que mostra la figura 2.2.

3rd Step Review Your Answers

Answers

Custom Part Number
 Custom Part Number : AC965-12700-6.000-MW-25.400-C-N-MM

Your Design's Closest Stock Spring

Price	Stock Part Number	Outer Diameter (in.)	Free Length (in.)	Rate (lbs./in.)
\$2.13	PC038-500-6000-MW-1000-C-N-IN	0.500	1.000	7.594

View Results in Stock

Rates & Loads

Spring Rate (or Spring constant), k : **1.329 N/mm**

True Maximum Load, F_{max} : **27.985 N**

Maximum Load Considering Solid Height, F_{max} : **24.776 N**

Safe Travel

Potential True Maximum Travel w/ Longer Free Length, $True\ Travel_{max}$: **21.060 mm**

Maximum Travel Considering Solid Height, $Solid\ Height\ Travel_{max}$: **18.645 mm**

Minimum Loaded Height : **6.755 mm**

Physical Dimensions

Diameter of spring wire, d : **0.965 mm**

Outer diameter of spring, D_o : **18.765 mm**

THE SPRING STORE

YOUR DESIGN'S SIMILAR STOCK SPRINGS

Price	Stock Part Number	Outer Diameter (in.)	Free Length (in.)	Rate (lbs./in.)
\$5.64	PC035-480-5380-MW-1000-CG-N-IN	0.480	1.000	7.238
\$2.13	PC038-500-6000-MW-1000-C-N-IN	0.500	1.000	7.594
\$4.54	PC047-546-9500-MW-0940-C-N-IN	0.546	0.940	7.523

View Results in Stock

Instant Spring Quote

Below is an Instant Quote of the custom spring you designed. If desired Qty is between tiers it's more economical to jump up a tier. Minimum quote quantity 25 pieces (no less). 34.00

Would you like to order now?
 Just Click the "Buy Now Button" Or Call (951) 276 - 2777

[Email Instant Quote and Blueprint](#)

Small Volume - Lowest Overall Cost

25 Pieces	50 Pieces	100 Pieces	250 Pieces
\$ 12.200 each	\$ 8.121 each	\$ 4.315 each	\$ 1.841 each

Figura 2.2: Resultats de l'Spring Creator. Font: Axxess Springs [2].

Aquest aplicatiu web suggereix el tipus de molla més semblant que l'empresa té en estoc, en mostra la constant elàstica, la força i el desplaçament màxims, i en presenta varies mesures més. El simulador també exposa els preus per diferents lots de molles.

3 Molles helicoidals

Les molles helicoidals són elements elàstics que normalment estan fabricades amb un fil metàl·lic enrotllat en forma d'hèlix. El més freqüent és que el fil tingui secció circular, tot i que algunes molles tenen el fil de secció rectangular.

Les molles helicoidals s'utilitzen en diversos camps de l'enginyeria per emmagatzemar energia, per absorbir impactes o per mantenir una força entre dos sòlids. Els materials més utilitzats per a la seva fabricació són l'acer i el titani (veure taula A.1 de l'annex A).



Figura 3.1: Diverses molles helicoidals. Font: assignatura Vehicles [18].

S'anomenen molles helicoidals perquè la directriu del fil segueix la forma d'una hèlix. La forma geomètrica que adquireix es diu helicoide.

Aquest tipus de molles són els més utilitzats en la indústria perquè tenen un gran rang de constants de rigidesa k [N/m] que es poden obtenir jugant amb els diferents paràmetres geomètrics que les defineixen. A més, a igualtat de prestacions, aquestes molles tenen uns costos de fabricació menors a d'altres tipologies (discoidals, ballestes, torsionals, entre d'altres, mostrades a la figura 3.2).

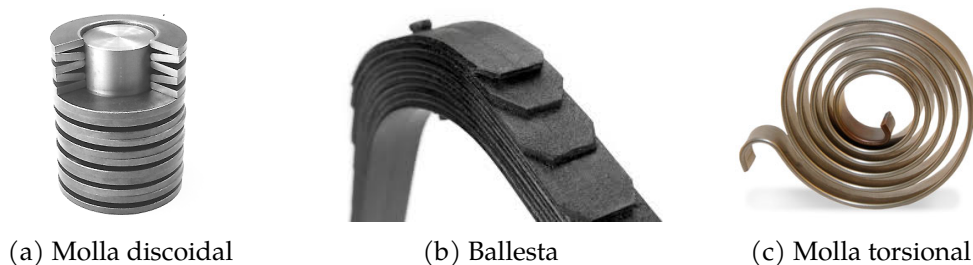


Figura 3.2: Diverses molles. Font: Google Images

3.1 Paràmetres geomètrics

Per definir la forma d'una molla helicoidal, anomenada helicoide, calen principalment quatre valors:

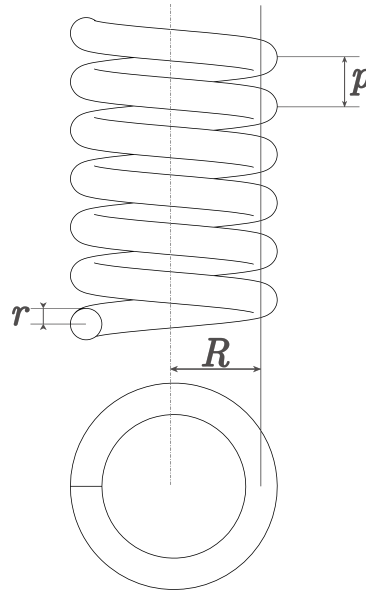


Figura 3.3: Molla amb $n = 6$ espirals. Font: pròpia.

- El nombre d'espires n : és el nombre de voltes que el fil fa a la directriu de l'hèlix. No té unitats.
- El pas p : la distància entre dues espirals consecutives. Es mesura normalment en mil·límetres.
- El radi del fil r , que defineix el gruix del fil. Es mesura en mil·límetres.
- El radi de l'hèlix R , que marca la distància entre la generatriu de l'hèlix i la del fil. Es mesura en mil·límetres.

La molla més senzilla i habitual, els 3 últims paràmetres són constants al llarg de tota la directriu de l'hèlix, com a la figura 3.3. Tanmateix, s'utilitzen molles de pas variable, de radi d'hèlix variable (com a la figura 3.1), i menys sovint, de radi de fil variable. Aquestes geometries permeten aconseguir, entre d'altres, molles més compactes (que poden ser útils si la geometria del disseny mecànic limita l'espai), o molles amb diferent rigidesa al llarg de la seva carrera.

3.2 Fabricació

Hi ha dues maneres habituals de fabricar molles helicoidals, basades en enrotllar filferro. S'utilitzen màquines de control numèric anomenades enrotlladores (*auto coiler* en anglès) que defineixen la geometria de la directriu de l'hèlix amb precisió, com es pot veure a la figura 3.4. Aquestes màquines s'alimenten automàticament amb filferro.

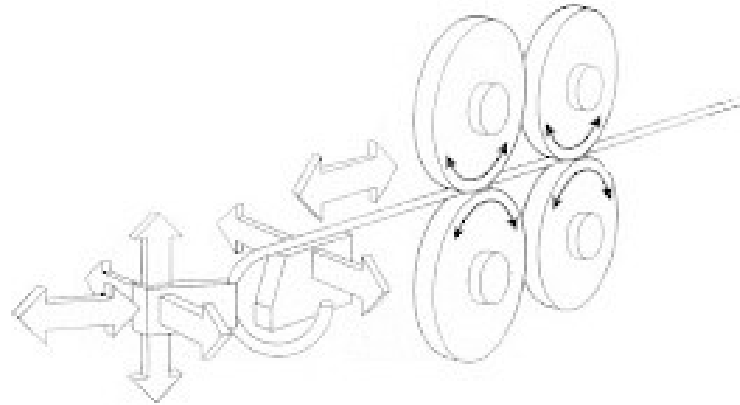


Figura 3.4: Esquema del control numèric en una enrotlladora per a molles petites. Font: *Axcess Springs* [1].

Per a molles petites, com ara la d'un bolígraf, se sol fer servir una enrotlladora que només té capçal i alimentació, com la que es veu a la figura 3.5. Al sortir el filferro pel capçal, aquest l'orienta mitjançant dos angles per a establir el pas i el diàmetre de l'espiral. El filferro surt enrotllat i quan la molla és completa, la màquina la talla i peça cau. Ja que les molles petites requereixen poc treball de fabricació, aquest procés se sol dur a terme en fred, segons l'empresa del sector de les molles *Diamond Wire Spring Company* [6].

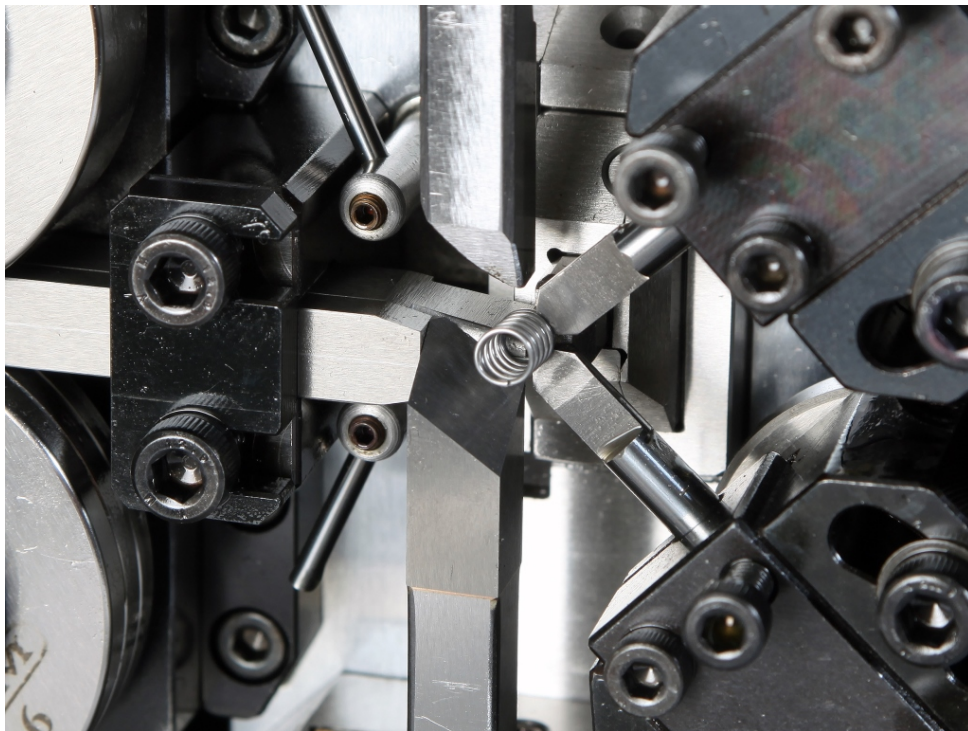


Figura 3.5: Enrotlladora per a molles petites. Font: *Whitelegg* [22].

Quan les molles són més grans, com les d'una suspensió McPherson [18], es fabriquen amb unes

enrotlladores similars a la que apareix a la imatge de la figura 3.6, que condueixen el filferro (normalment en calent) al voltant d'un cilindre giratori, després d'haver-se fixat primer un extrem al cilindre.

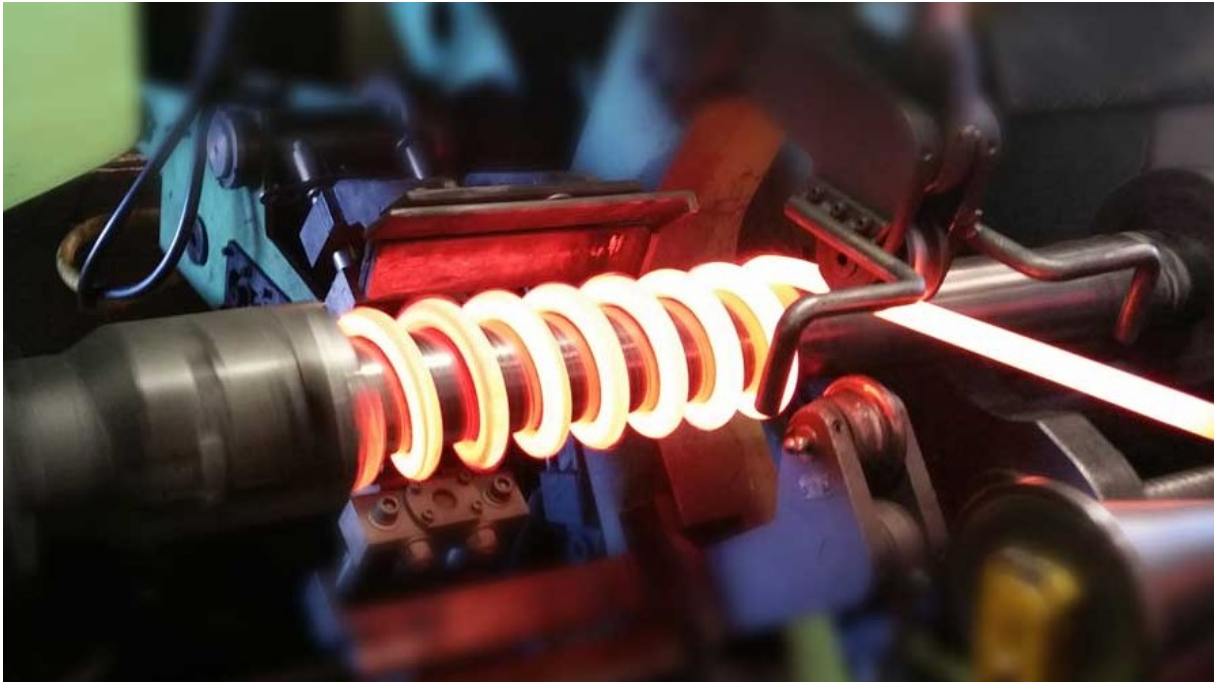


Figura 3.6: Enrotlladora de molles grosses. Font: King Springs [8].

Si s'està fabricant una tirada curta o un prototip, o no es disposa d'una màquina de control numèric, es pot fabricar la molla enrotllant filferro en un torn. En aquest cas, cal tenir en compte els riscos per a la persona que operi el torn, ja que si el filferro no està ben subjectat es pot desenrotllar ràpidament, cosa que pot ser extremadament perillosa per a les persones properes, especialment si es tracta d'un filferro gruixut, perquè emmagatzema molta energia de deformació durant l'operació.

Es poden dur a terme altres operacions en la fabricació de la molla, com ara llimar els extrems superior i inferior per tenir l'acabament desitjat, tractaments superficials per a millorar l'aspecte de la molla o tractaments tèrmics per millorar-ne les propietats mecàniques.

3.3 Materials més utilitzats

Els materials més utilitzats en la fabricació de molles són sobretot metalls. El que realment aporta joc per obtenir les propietats mecàniques desitjades és l'aliatge dels metalls. Així es pot assolir el mòdul elàstic de cisellament G , la densitat ρ i la duresa idònies. També es fan servir composts de fibres de vidre, carboni o aràmida i matrius de resina, tal com exposen Petrich, M. et al. [16], però el seu càlcul resistent resta més enllà de l'abast d'aquest treball.

El metall d'ús més extensiu és l'acer, que és el metall més estudiat quan es tracta d'aliar-lo, i té un baix cost. Dins dels acers, es fan servir acers durs o semidurs amb alt contingut de carboni, acers aliats amb crom i vanadi o amb crom i silici, i quan les condicions de funcionament ho requereixen, s'utilitza l'acer inoxidable.

Pel que fa els altres metalls, també s'utilitzen el bronze al fòsfor i el coure al beril·li, per la seva resistència a la corrosió i la seva conductivitat elèctrica. A part, existeixen aliatges comercials com els Monel, els Inconel, l'Elgiloy, el NiSpan o el Hastelloy, segons la web *Spring-I-Pedia* [10].

A la taula A.1 de l'annex A es poden trobar els materials mencionats en aquest apartat i les seves propietats químiques i elàstiques, i els seus codis estandaritzats.

3.4 Acabaments

El nombre d'espires n de la molla es compon per dos tipus d'espires: les actives i les inactives.

- El nombre d'espires actives n_e en una molla és el nombre d'espires que poden doblegar-se sota la càrrega, sent les úniques que treballen elàsticament.
- S'anomenen espires inactives a les voltes de fil que no treballen elàsticament. El nombre d'espires inactives depèn de com s'hagin fabricat els extrems de la molla.

Segons l'empresa de molles *Access Spring* [1], els tipus d'acabament possible són els quatre següents:

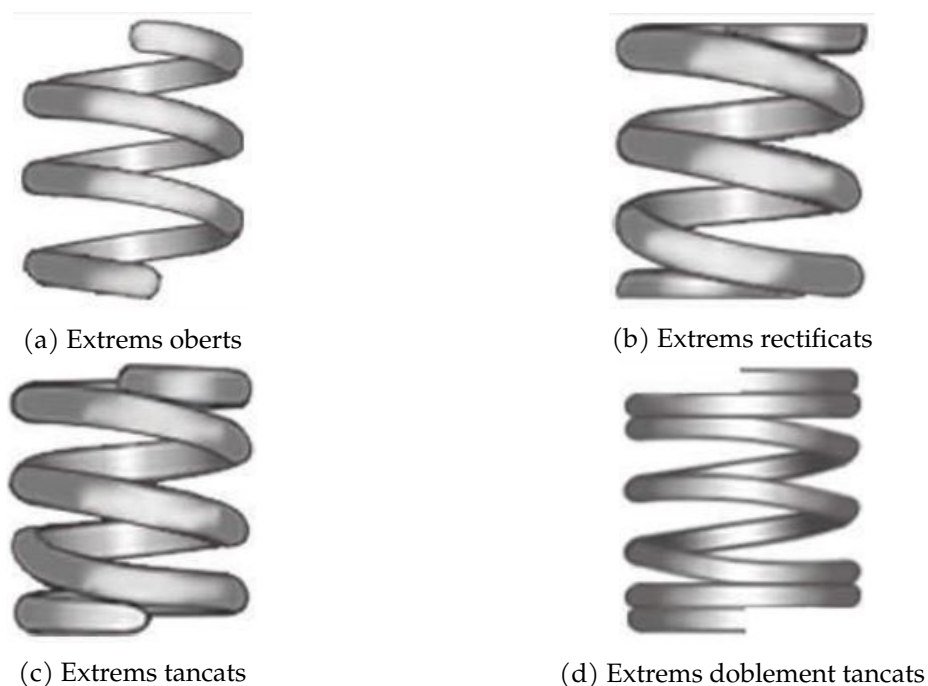


Figura 3.7: Diferents acabaments de molles. Font: *Access Spring* [1].

La longitud en buit de la molla L_0 , que canvia segons el tancament escollit, es pot calcular a partir del pas p , el nombre de voltes de l'hèlix n_e , i el radi del fil r .

- Extrems oberts (figura 3.7a):
Es recomana aquest tipus d'acabament quan es necessita un recorregut de molla suficient perquè es comprimeixi fins a una longitud de bloc (totes les espires en contacte) petita en comparació a L_0 . Amb aquest tipus de tancaments, totes les espires de la molla són

actives. Cal tenir en compte que la molla s'hauria d'instal·lar en un eix o en un forat per mantenir-la fixa al lloc i evitar el vinclament.

$$n_e = n \quad (1)$$

$$L_0 = n_e p + 2r = np + 2r \quad (2)$$

- Extrems rectificats (figura 3.7b):

En aquest acabament, les últimes voltes de l'espira estan en contacte amb les immediatament anteriors i es mecanitzen els finals de la molla per obtenir dues superfícies planes. Tot i el cost afegit de fabricació, aquest acabament és força comú. Aquests acabaments aporten més estabilitat a la molla, la protegeixen del vinclament, i és útil per exemple si la longitud de la molla en buit L_0 és gran en comparació amb el diàmetre de l'hèlix $2R$.

$$n_e = n - 2 \quad (3)$$

$$L_0 = n_e p = (n - 2) p \quad (4)$$

- Extrems tancats (figura 3.7c):

Les últimes voltes de l'espira també estan en contacte amb les immediatament anteriors, però sense mecanitzar-se. En la majoria de casos, les molles amb finals tancats s'aguanten rectes i tenen bastanta estabilitat. És el tancament més comú.

$$n_e = n - 2 \quad (5)$$

$$L_0 = n_e p + 2r = (n - 2) p + 2r \quad (6)$$

- Extrems doblement tancats (figura 3.7d):

Són com els extrems tancats però a cada extrem hi ha dues espires en contacte. Aporta estabilitat en molles de gran diàmetre i fil prim. Aquest acabament és més econòmic que el rectificat.

$$n_e = n - 4 \quad (7)$$

$$L_0 = n_e p + 4r = (n - 4) p + 4r \quad (8)$$

Les equacions (1), (3), (5) i (7) que calculen el nombre d'espires actives a partir del tancament s'han extret de la pàgina web *Spring-I-Pedia* [9]. Les equacions (2), (4), (6) i (8) per calcular la longitud en buit s'han deduït a partir de la geometria.

3.5 Càlcul de molles

Aquest treball es centra en el càlcul resistent de molles helicoidals en situacions de càrrega estàtica. S'ha considerat que l'estudi de la fatiga del material queda més enllà del seu abast.

Una molla helicoidal, simplificadament, es pot estudiar com una barra a torsió que té per directriu una hèlix. La càrrega F es considera aplicada als extrems de la molla i produeix en el fil un moment torsor M_t i una força a cisallament T , com mostra la figura 3.8.

$$M_t = F \cdot R \quad (9)$$

$$T = F \quad (10)$$

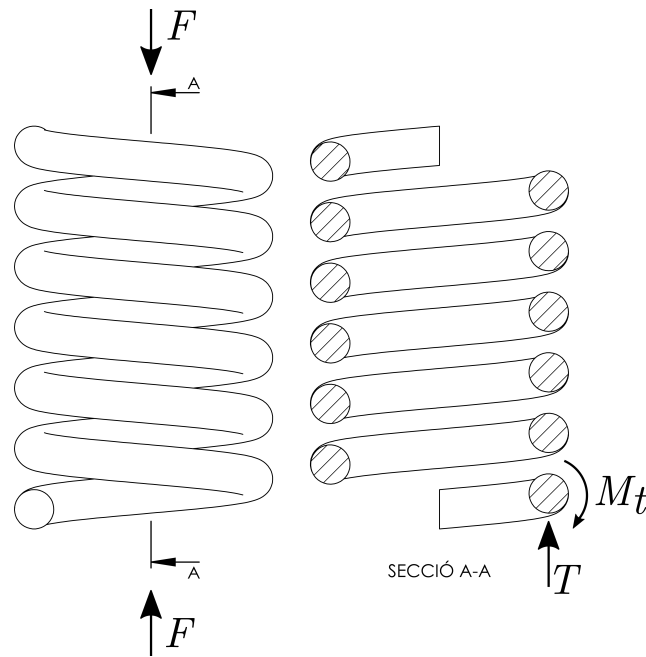


Figura 3.8: Esquema de forces en una molla. Font: pròpia.

3.5.1 Constant de la molla

A partir de la llei de Hooke (11):

$$\tau = G\gamma \quad (11)$$

On G és el mòdul elàstic a torsió del material i γ és el doble de la variació angular diferencial entre direccions perpendiculars canviada de signe. L'efecte de la força de cisallament s'ignora perquè aporta molt poc a la deformació en comparació al moment torsor. L'helicoides és la barra a torsió enrotllada, per tant la tensió tallant del fil de la molla en aplicar-se un diferencial de força dF en un anell circular de gruix dr que gira θ (quocient entre la deflectació de la molla δ i el radi de l'hèlix R) és la descrita per l'equació (12).

$$\tau = \frac{dF}{2\pi dr} \quad (12)$$

Alhora, la deformació a cisalla s'especifica a l'equació (13), que descriu el quocient entre el que es mou l'exterior del fil al girar un angle θ entre la longitud útil del fil, $2\pi Rn_e$.

$$\gamma = \frac{r\theta}{2\pi Rn_e} \quad (13)$$

Aleshores es poden combinar les equacions (12) i (13) a l'equació (11)

$$\frac{dF}{2\pi r dr} = G \frac{r\theta}{2\pi R n_e} \quad (14)$$

$$\frac{dF}{dr} = G \frac{r^2\theta}{R n_e} \quad (15)$$

$$r \frac{dF}{dr} = G \frac{r^3\theta}{R n_e} \quad (16)$$

$$\frac{d\tau}{dr} = G \frac{r^3\theta}{R n_e} \quad (17)$$

$$\int d\tau = \int G \frac{r^3\theta}{R n_e} dr \quad (18)$$

$$\tau = \frac{Gr^4}{4R n_e} \theta \quad (19)$$

L'energia de deformació de la peça en qüestió es pot trobar des del punt de vista de la molla encongint-se una distància δ i des del punt de vista de la barra a torsió girant un angle θ , com mostra l'equació (20).

$$E_{pe} = \frac{k\delta^2}{2} = \frac{1}{2} \frac{Gr^4}{4R n_e} \theta^2 \quad (20)$$

Així doncs, tenint en compte que $\theta = \delta/R$, se'n pot aïllar la constant elàstica de la molla, com es pot veure a l'equació (22):

$$\frac{k\delta^2}{2} = \frac{1}{2} \frac{Gr^4}{4R n_e} \left(\frac{\delta}{R}\right)^2 \quad (21)$$

$$k = \frac{Gr^4}{4R^3 n_e} \quad (22)$$

3.5.2 Deformació màxima

També és interessant conèixer la deformació màxima que pot tenir una molla. Aquesta situació (sempre i quan el material no falli abans, com es discuteix a l'apartat 3.5.3) succeeix quan totes les espirals de la molla helicoidal estan en contacte. L'alçada de l'hèlix sense càrrega és L_0 (veure apartat 3.4), i el gruix de fil en contacte és $L_{bloc} = 2rn$. Per tant, com es mostra a l'equació (23), la compressió màxima és l'espai entre espirals multiplicat pel nombre d'espirals actives.

$$\delta_{max} = L_0 - L_{bloc} = L_0 - 2rn = n_e(p - 2r) \quad (23)$$

A partir d'aquesta deformació màxima de l'equació (23) i la constant de la molla (22) es pot trobar la força que causa la compressió màxima (equació (24)), anomenada F_{max} .

$$F_{max} = k\delta_{max} = \frac{Gr^4}{4R^3 n_e} n_e(p - 2r) = \frac{Gr^4}{4R^3} (p - 2r) \quad (24)$$

3.5.3 Càlcul de tensions

Un cop determinats els esforços als quals està sotmesa la secció del fil (equacions (9) i (10)), es pot estudiar l'estat de tensions.

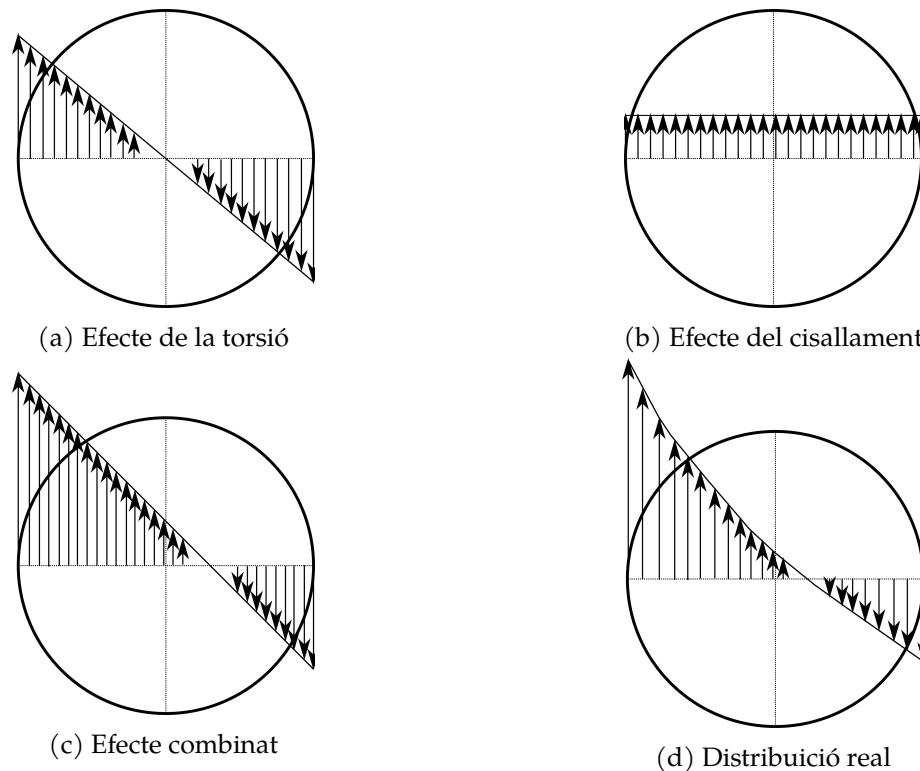


Figura 3.9: Estat de tensions. Font: pròpia

El moment torsor causa unes tensions tallant lineals i proporcionals al radi, segons la teoria de Coulomb (figura 3.9a), i l'esforç de cisallament en crea una distribució constant de tensions tallant segons Collignon (figura 3.9b). Per tant, teòricament, l'estat de tensions tallants és la superposició de les dues (figura 3.9c). Aquesta distribució de tensions tallants té un màxim, descrit per l'equació (25).

$$\tau_{max} = \frac{M_f r}{I_t} + \frac{F}{A} = \frac{2FR}{\pi r^3} + \frac{F}{\pi r^2} = \left(1 + \frac{1}{2C}\right) \frac{2FR}{\pi r^3} = K_s \frac{2FR}{\pi r^3}, \quad C = \frac{R}{r} \quad (25)$$

S'anomena *relació d'enrotllament* al paràmetre C , i permet calcular K_s (equació (26)), conegut com a *factor corrector de l'esforç tallant*.

$$K_s = 1 + \frac{1}{2C} \quad (26)$$

En realitat, s'ha observat que hi ha una concentració de tensions a la part interior del fil de la molla deguda a la curvatura de l'espira, com mostra la figura 3.9d: la distribució de tensions canvia respecte la prevista i s'accentua en el punt màxim de tensió. Tot i així, només és necessari tenir en compte aquest factor en el càlcul dinàmic de les molles helicoidals, ja que en casos de càrregues estàtiques, el propi material de la molla deforma plàsticament (a escala reduïda) fent que s'eliminin les concentracions de tensions com s'explica a Disseny i càlcul de molles [4].

Tenint el valor de la tensió màxima, es pot comparar amb un criteri de fallida, com per exemple, von Mises (ja que els metalls que s'utilitzen per fer molles són materials dúctils), per a determinar la fallida o no del material dins d'una tensió admissible. En el cas de les tensions

normals pures no hi ha tensions en direcció axial (que s'anomenarà 1) ni en les perpendiculars al fil (que s'anomenaran 2 i 3), tal i com mostra l'equació (27), així com no existeixen tensions tallants en els plans formats per les direccions 1 i 2 i les direccions 1 i 3, com mostra l'equació (28). L'única tensió tallant existent és la descrita a l'equació (25), que correspon al pla format per les direccions 2 i 3, com indica l'equació (29)

$$\sigma_{11} = \sigma_{22} = \sigma_{33} = 0 \quad (27)$$

$$\tau_{12} = \tau_{31} = 0 \quad (28)$$

$$\tau_{23} = \tau_{max} \quad (29)$$

La tensió equivalent de von Mises a comparar amb la de fallida és per tant la que mostra l'equació (30), tenint en compte les equacions (27), (28) i (29) explicades anteriorment.

$$\sigma_{vM} = \sqrt{\frac{(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2}{2} + 3(\tau_{12}^2 + \tau_{23}^2 + \tau_{31}^2)} = \sqrt{3} \cdot \tau_{max} \quad (30)$$

La tensió tallant màxima admissible es troba a partir de l'equació (31), sabent la tensió normal admissible del material, que en el camp de disseny de molles es troba a partir de la seva resistència última a la tracció, com s'explica a *Comprehensive Spring Design* [20]:

$$\tau_{adm} = \frac{\sigma_{adm}}{\sqrt{3}} \quad (31)$$

Per tant, utilitzant les equacions (25) i (31), es pot trobar l'expressió de la força màxima admissible que es pot aplicar a la molla helicoidal, com mostra l'equació (32).

$$F_{adm} = \frac{\pi r^3 \sigma_{adm}}{2\sqrt{3}K_s R} \quad (32)$$

Es pot trobar la màxima compressió admissible de la molla a partir de la força admissible de l'equació (32) i la constant de la molla de l'equació (22), tal com mostra l'equació (33).

$$\delta_{adm} = \frac{F_{adm}}{k} = \frac{\pi r^3 \sigma_{adm}}{2\sqrt{3}K_s R} \cdot \frac{4R^3 n_e}{Gr^4} = \frac{2\pi n_e \sigma_{adm} R^2}{\sqrt{3}K_s Gr} \quad (33)$$

Cal tenir en compte que aquesta deformació admissible de la molla va lligada a un coeficient de seguretat. A més, és possible que no es doni mai en una determinada molla helicoidal si abans d'aquesta deformació les espirals fan contacte, és a dir, $\delta_{max} \leq \delta_{adm}$.

3.5.4 Altres càlculs

- Energia potencial elàstica màxima

La màxima energia potencial elàstica E_{max} que es pot emmagatzemar és l'àrea sota la corba força-desplaçament. Després de trobar el desplaçament i la força màxims admissibles d'una molla helicoidal amb les equacions (23) i (24) o (33) i (32), es tria la parella δ, F de menor valor, i es pot esbrinar la màxima energia que pot emmagatzemar la molla com a potencial elàstica amb l'equació (34):

$$E_{max} = \frac{F_{max} \delta_{max}}{2} \quad (34)$$

- Massa

La massa m es calcula a partir del volum de la molla i de la densitat del material. El volum de la molla helicoidal es pot interpretar com una superfície πr^2 al llarg d'una directriu de longitud $2\pi nR$. Per tant, multiplicant aquests dos termes per la densitat s'obté la massa a l'equació (35).

$$m = \rho 2\pi^2 n R r^2 \quad (35)$$

- Freqüència pròpia

Considerant que l'única massa suspesa en una molla és la mateixa massa de la molla, es pot esbrinar quina és la primera freqüència natural del sistema massa-molla ideal. Els efectes de dissipació energètica de l'aire on es troba el sistema o les microdeformacions plàstiques del material s'ignoren. La pulsació pròpia del sistema ω és la descrita per l'equació (36).

$$\omega = \sqrt{\frac{k}{m}} \quad (36)$$

A partir de la pulsació es pot esbrinar fàcilment la freqüència f en Hertz, com es veu a l'equació (37).

$$f = \frac{\omega}{2\pi} = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (37)$$

4 Interfície gràfica d'usuari (GUI)

4.1 Definició

Una interfície gràfica d'usuari (*graphic user interface* en anglès, d'aquí l'acrònim *GUI*) és un mitjà de comunicació entre l'usuari i un ordinador a través d'elements gràfics, com diu l'equip del web *Microsoft.com* [15]. El principal objectiu és estandaritzar i simplificar l'ús de programes d'ordinador, principalment utilitzant el ratolí per seleccionar opcions en menús o prendre decisions amb botons en comptes de comandes escrites. Aquests mètodes permeten augmentar notablement la interactivitat del programari, tot i que realitzi les mateixes funcions que les comandes escrites.

L'espai gràfic, com el de la figura 4.1, se sol basar en una finestra amb elements, com per exemple botons, entrades de text, caselles de selecció i zones per mostrar informació, ja sigui en forma de text, d'imatge o d'altres com vídeos, àudios...

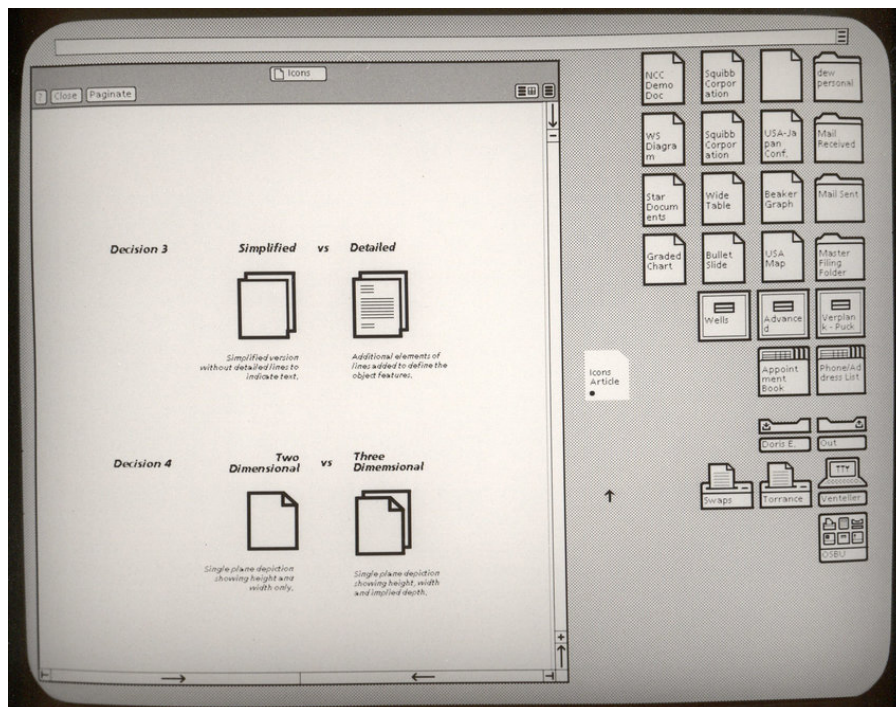


Figura 4.1: La interfície gràfica d'usuari del Xerox Star, el primer ordinador comercial que en va tenir. Font: Karl Voit [21].

Les accions que l'usuari fa al programa fan disparar els anomenats esdeveniments, que són l'equivalent informàtic de les accions en un objecte de l'espai gràfic que el programa reconeix i gestiona posteriorment. L'estructura bàsica del programa és un bucle *while* que té com a condició de funcionament que no s'hagi aturat l'execució del programa. En aquest bucle, s'espera al següent esdeveniment i un cop succeeix, l'estat del programa s'actualitza segons escaigui per cada esdeveniments possible.

Els programes orientats a esdeveniments, com les interfícies gràfiques d'usuari, es poden escriure en qualsevol llenguatge de programació, però el més usual és fer-ho en llenguatges d'alt

nivell, és a dir, amb una alta abstracció respecte dels detalls de funcionament de la maquinària física de l'ordinador.

4.2 Gestors d'esdeveniments

S'anomenen gestors d'esdeveniments (en anglès, *event handlers*) a les funcions que gestionen els inputs rebuts en el programa. Solen ser crides de retorn (*callbacks* en anglès), és a dir, codi que passa com a argument en una funció. En aquests gestors, qui programa lliga les ordres adequades amb l'acció que l'usuari realitza, amb la finalitat d'actualitzar l'estat del programa al nou estat que escaigui.

L'acció de clicar, per posar un exemple, sol estar composta de dos esdeveniments amb els seus gestors corresponents: un quan es clica i un altre quan s'aixeca el dit del botó i es deixa de clicar. De la mateixa manera, una llista de selecció pot tenir gestors per als esdeveniments de desplegar-la, marcar un ítem o escollir-lo.

En conclusió, els gestors d'esdeveniments són els elements programàtics que permeten al codi obtenir informació de les accions de l'usuari i executar instruccions en resposta.

4.3 Entorn de desenvolupament

La gran majoria de vegades, les interfícies d'usuari s'escriuen en entorns de desenvolupament integrats (*IDEs*, per les seves sigles en anglès), que són normalment programes d'escriptori amb interfície gràfica que incorporen principalment, segons l'escriptora tècnica Margaret Rouse [17]:

- Un editor de codi, preparat per revisar la sintaxi d'un o més llenguatges.
- Un compilador (o interpretador, en el seu defecte) per convertir el codi en llenguatge màquina, que permeti executar el programari que s'estigui desenvolupant.
- Un depurador per netejar errors en el que s'estigui desenvolupant.
- Un navegador d'arxius, de classes i/o d'objectes, per visualitzar l'estructura del programari que s'està elaborant i navegar-lo.

Alguns *IDEs* són específics per desenvolupar programari per a ús humà a través d'interfícies gràfiques en escriptori, web o mòbil. Difereixen dels altres *IDEs* en què incorporen eines de disseny d'interfícies gràfiques, les quals permeten crear l'espai gràfic desitjat pel programa que s'està desenvolupant a partir d'una interfície gràfica, afegint-hi i editant els elements d'una manera intuïtiva.

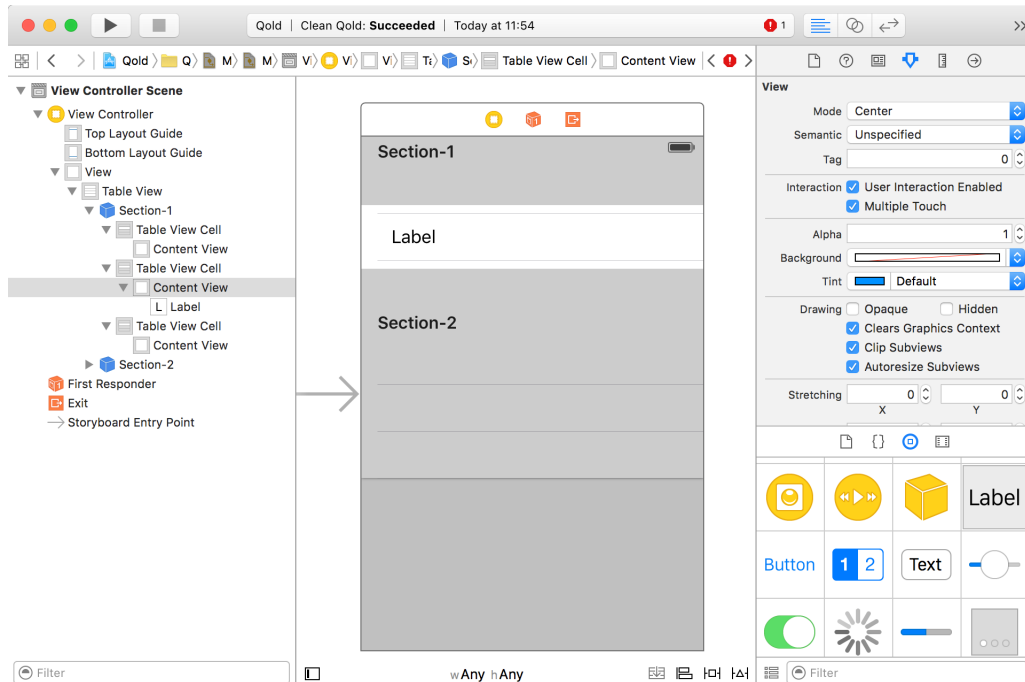


Figura 4.2: Xcode, la plataforma de desenvolupament de programari per a iOS, macOS, watchOS i tvOS d'Apple. Font: Apple Developer [3].

Normalment, el mateix *IDE* s'encarrega d'interpretar una alteració en el disseny de la interfície gràfica (posició dels elements, colors, mides, etcètera) i canviar el codi corresponent en consegüent, i viceversa, que un canvi en el codi actualitzi l'estat del disseny d'interfície gràfica.

4.4 *MATLAB® App Designer*

A l'entorn de computació numèrica *MATLAB®*, de la companyia americana *MathWorks*, s'inclou des de la seva versió 9.0 (llançament R2016a) l'*App Designer* [11], un entorn de desenvolupament orientat a crear aplicacions per al propi *MATLAB®*, o aplicacions independents d'escriptori i/o navegador.

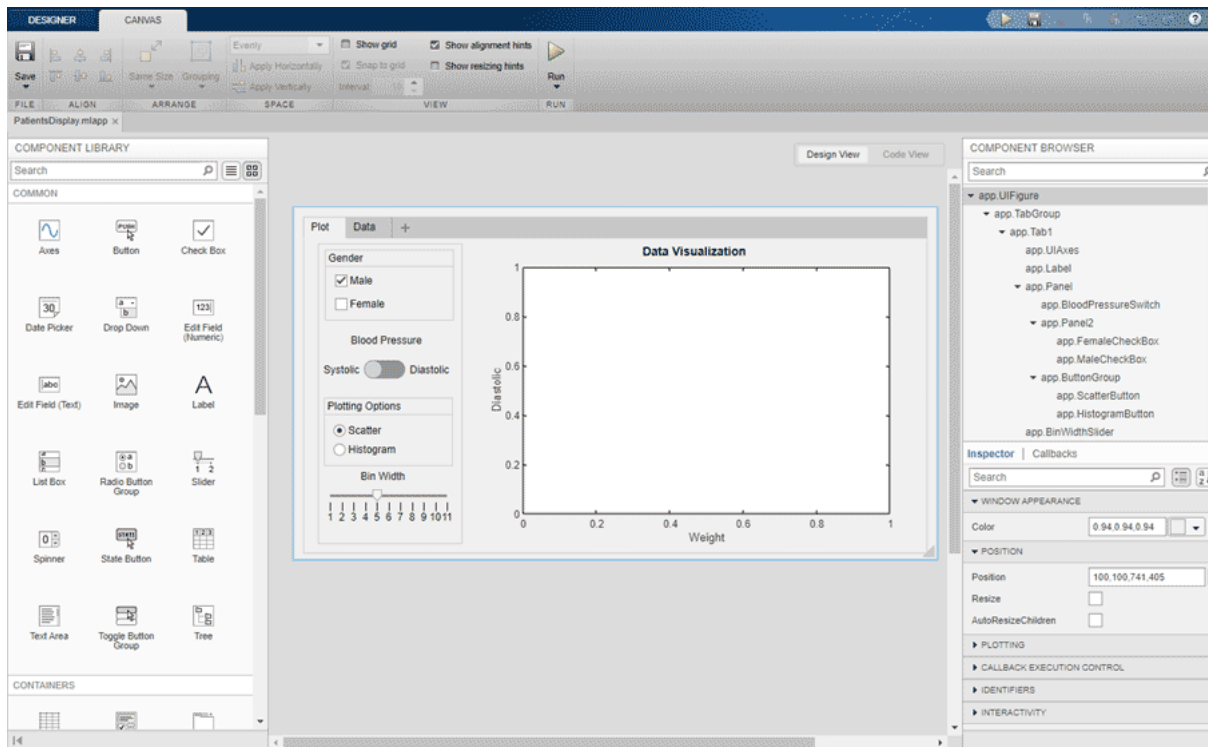


Figura 4.3: Vista de disseny d'interfície del MATLAB®
App Designer. Font: MATLAB® [11]

Aquest entorn té dos modes de funcionament: *Design View* (vista de disseny) i *Code View* (vista de codi). El primer (figura 4.3) permet arrossegar els elements que es vulguin introduir des de la part esquerra (la llibreria de components) i deixar-los anar sobre l'espai gràfic; també dona consells d'alineació dels elements. L'*App Designer* genera automàticament el codi que especifica el disseny i la disposició dels elements gràfics.

A la part dreta hi ha un navegador dels elements que s'estan utilitzant en el disseny de l'espai gràfic, que permet veure'n la jerarquia clarament. Aquesta jerarquia indica quins elements estan continguts per un altre. També hi ha l'editor de propietats dels elements i unes opcions per afegir-hi gestors d'esdeveniments, a baix a la dreta.

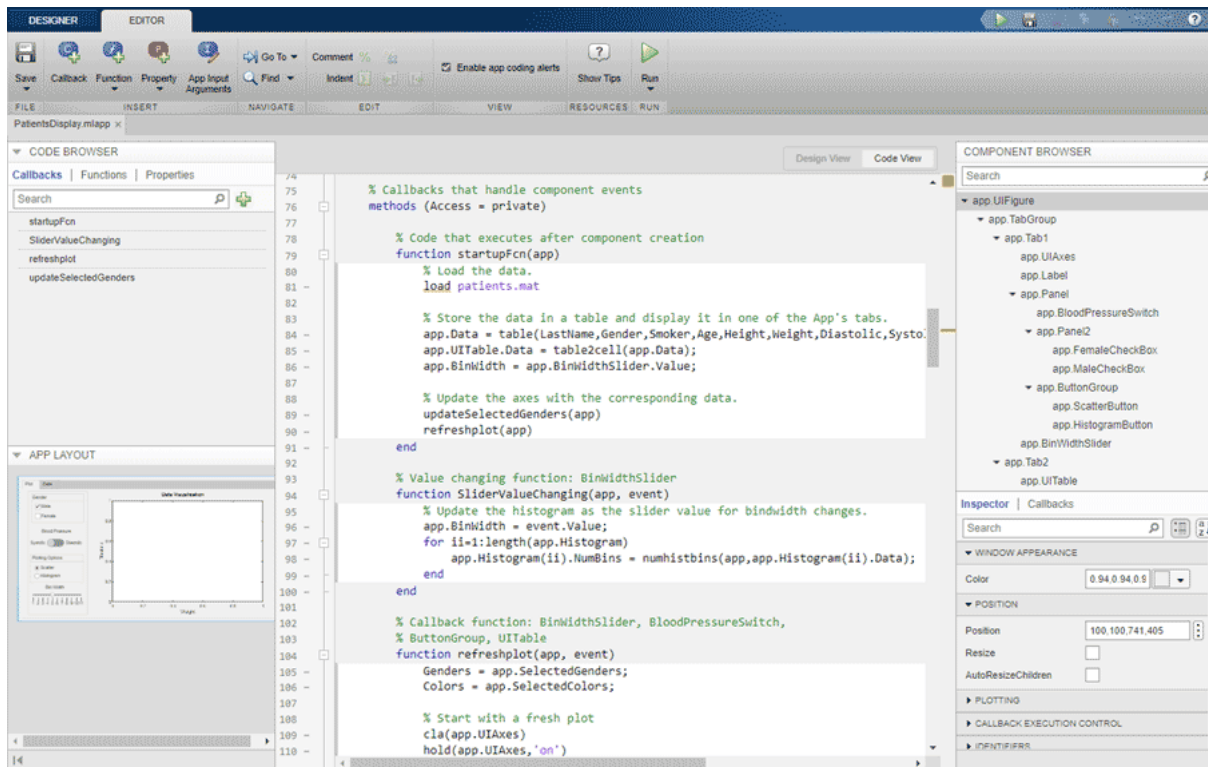


Figura 4.4: Vista de codi del MATLAB® *App Designer*. Font: MATLAB® [11]

Pel que fa la vista de codi (figura 4.4) a la part esquerra, s'hi troba un navegador de codi que permet accedir ràpidament a gestors d'esdeveniments, funcions i propietats. Just a sota es troba una miniatura de la disposició gràfica. A la part dreta apareix el mateix navegador i l'editor de propietats que hi ha a la vista de disseny.

A la part central es pot escriure a les zones blanques. El color gris indica el codi generat pel mateix *App Designer*, que no es pot editar directament, sinó a través de l'editor de propietats dels elements.

5 Programa

5.1 Escenari gràfic

L'escenari gràfic dissenyat amb l'ajuda de la vista de disseny d'interfície del *MATLAB® App Designer* es mostra a la figura 5.1.

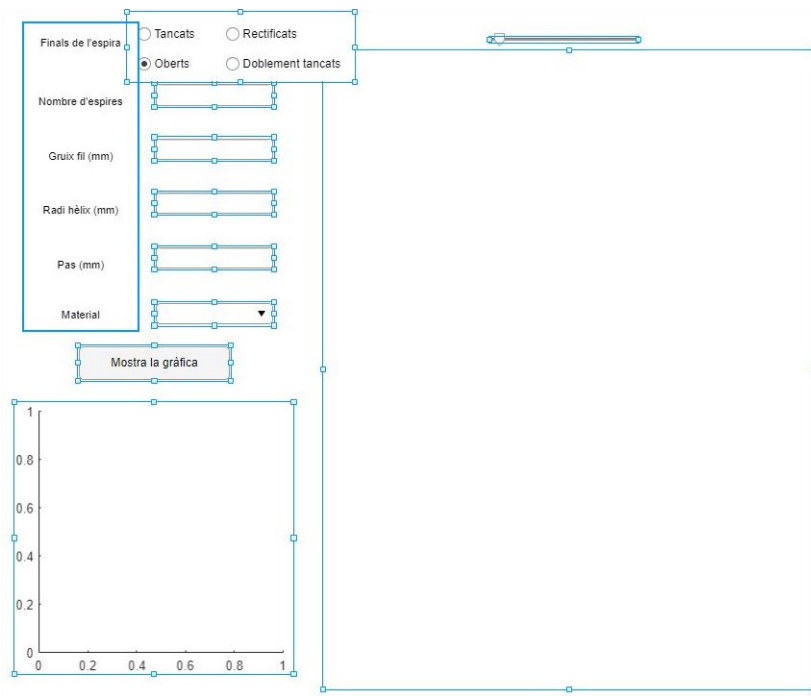


Figura 5.1: Elements de la interfície gràfica. Font: pròpia.

Els objectes que el componen són:

- `app.grafic`, la part dreta, que és l'espai gràfic on es mostra un renderitzat de la molla descrita (apartat 5.5.2)) o la seva corba força-desplaçament (apartat 5.5.4).
- `app.slider_resolucio`, a sobre de l'espai gràfic, que és un control lliscant o *slider* que permet canviar la resolució del renderitzat de la molla (veure apartat 5.5.2).
- `app.dades`, a baix a l'esquerra, que és un espai gràfic on s'hi mostren els valors de la longitud en buit de la molla, la deformació màxima, la força màxima, l'energia potencial elàstica que emmagatzema, la seva massa i la seva freqüència pròpia (veure apartat 5.5.5).
- `app.text_opcionsfinal`, `app.text_n`, `app.text_d`, `app.text_R`, `app.text_pas` i `app.text_material`, a dalt a l'esquerra. Són 6 quadres de text un sota l'altre. Cal destacar que s'utilitza el gruix del fil $d = 2r$ en comptes del radi perquè és molt més habitual parlar del gruix d'un fil que del seu radi.
- `app.Tancaments`, a dalt al costat de l'`app.slider_resolucio`. Es tracta de quatre botons de selecció, agrupats en un sol objecte. Aquests botons serveixen per descriure l'acabament de la molla helicoidal (veure apartat 3.4).

- `app.TancatsButton` a dalt a l'esquerra
 - `app.RectificatsButton` a dalt a la dreta
 - `app.ObertsButton` a baix a l'esquerra
 - `app.DoblementtancatsButton` a baix a la dreta
- `app.input_n`, `app.input_d`, `app.input_R` i `app.input_pas`, just a sota dels botons de selecció. Es tracta de 4 entrades de text on l'usuari escriu mitjançant el teclat del seu equip els valors dels paràmetres de la molla, explicats a l'apartat 3.1.
 - `app.tria_material` ubicat sota d'`app.input_pas` i a la dreta d'`app.text_material`. És un desplegable on es selecciona el metall del qual està feta la molla que l'usuari descriu a partir dels 18 metalls de la taula A.1 de l'annex A.
 - `app.boto_tria` just a sota, que és el botó amb el qual es canvia què s'enseny a l'espai gràfic `app.grafic` de la dreta: Si hi ha el renderitzat de la molla, a l'apretar el botó es passa a mostrar la corba força-deformació, i viceversa.

El programa, en funcionament, té l'aspecte de la figura 5.2 mostrada a continuació.

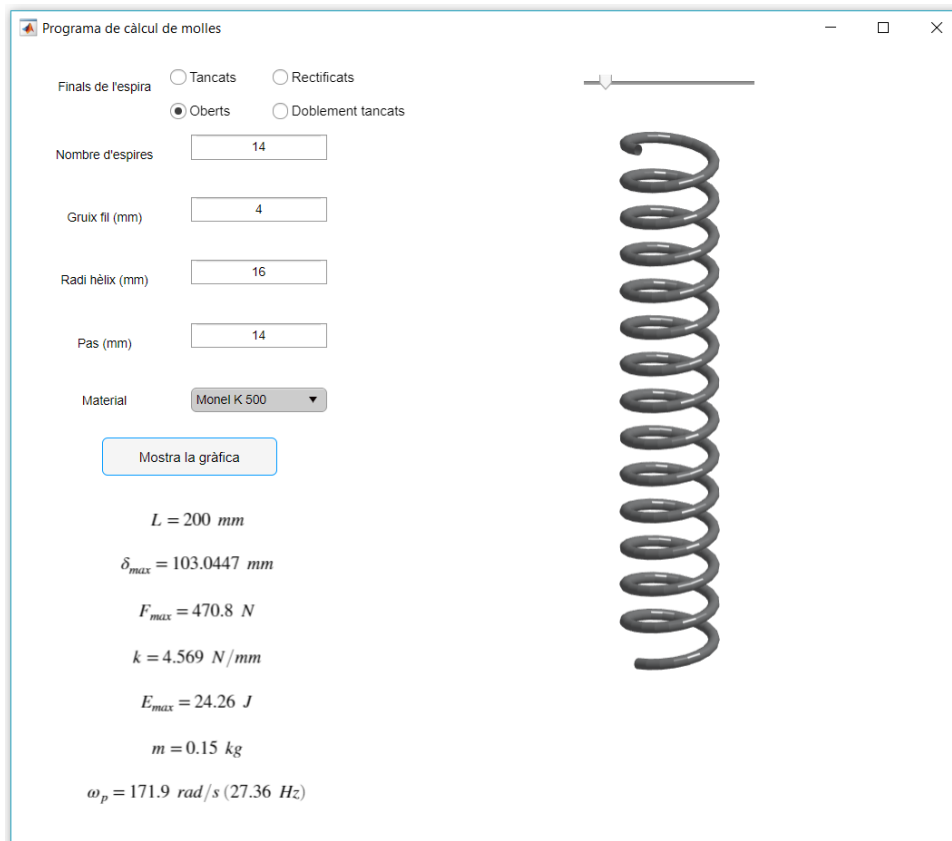


Figura 5.2: Aspecte del programa mostrant el renderitzat de la molla. Font: pròpia.

El renderitzat és interactiu; si es clica sobre la molla es pot girar en tres dimensions.

En clicar al botó `app.boto_tria`, el renderitzat i el *slider* desapareixen i apareix la corba força-desplaçament, com es veu a la figura 5.3.

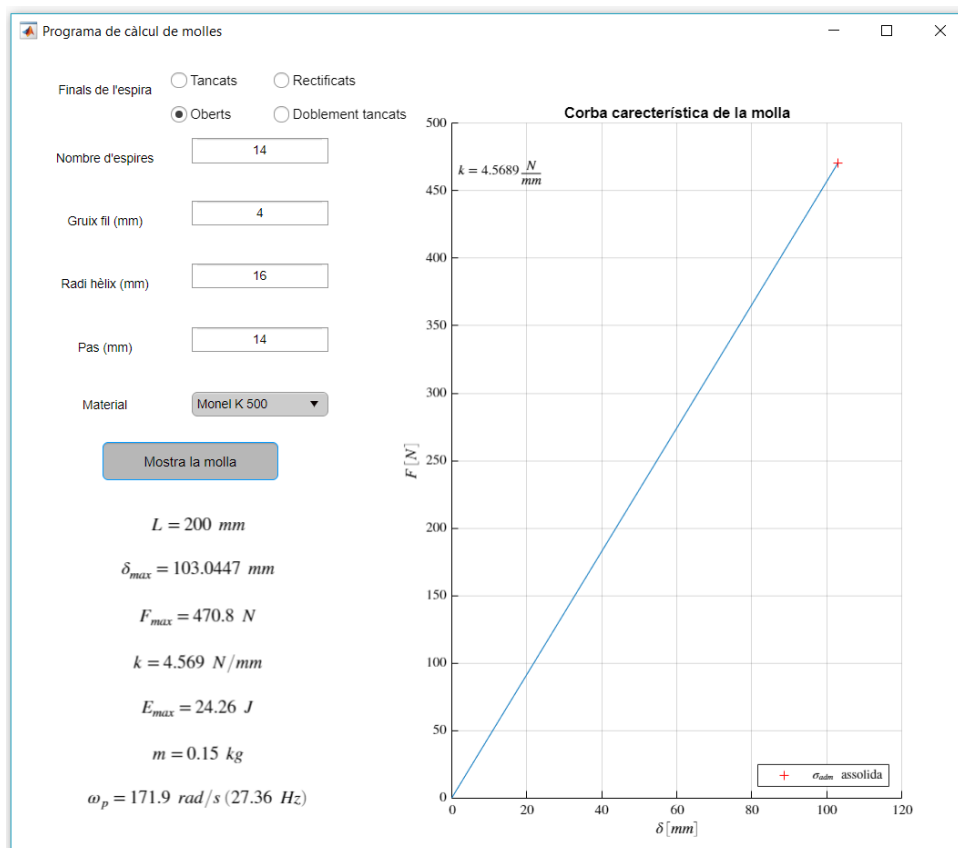


Figura 5.3: Aspecte del programa mostrant la corba força-desplaçament. Font: pròpia.

5.2 Funcionament

El funcionament intern del programa es pot separar en dues etapes. La primera etapa s'executa quan el programa s'engega, i s'explica en detall a l'apartat 5.2.1. L'altra etapa regeix quan el programa ja està funcionant, i es pot entendre com a un sistema orientat a actualitzar l'estat gràfic del programa segons les accions de l'usuari. A l'apartat 5.2.2 es desgrana aquest sistema.

El programa s'estructura en gran part en un fitxer principal, `GUI_App.mlapp` (annex B) que inclou les funcions següents:

- `extreuparametres`: De la línia 29 a la 38. Detallada a l'apartat 5.4.2.
- `entradavalida`: De la línia 40 a la 81. Detallada a l'apartat 5.4.1.
- `mostramolla`: De la línia 83 a la 95. Detallada a l'apartat 5.5.1.
- `mostracorba`: De la línia 97 a la 103. Detallada a l'apartat 5.5.3.
- `actualitza`: De la línia 105 a la 111. Detallada a l'apartat 5.5.

Fora del fitxer principal, en cinc fitxers `.m` a part amb el mateix nom que la funció que alberguen, s'hi emmagatzemen:

- La funció `graficacorba`: Detallada a l'apartat 5.5.4.
- La funció `renderitzamolla`: Detallada a l'apartat 5.5.2.
- La funció `ompleredades`: Detallada a l'apartat 5.5.5.
- La funció `espiresinactives`: Detallada a l'apartat 5.4.3.
- La classe `Material`: Detallada a l'apartat 5.3.

5.2.1 Inicialització

En engegar-se l'aplicació, primer de tot s'amaguen l'espai on es dibuixen la molla o la corba i l'espai on es posen les dades de la molla. Això es fa imposant propietats `app.grafic.Visible` i `app.dades.Visible` a `'off'`. El fet de mantenir aquests dos elements ocults fins al seu final redueix el cost computacional de l'operació d'inicialització.

Tot seguits'inicialitzen tots els botons d'opcions dels tancaments (anomenats *radio buttons* en anglès) a no marcats menys el botó d'"Oberts". La propietat que es canvia s'anomena `Value`, pren el valor 0 quan el botó no està marcat i, 1 en cas contrari.

Després es generen valors aleatoris per als paràmetres de la molla que es dibuixarà en iniciar-se el programa.

- El nombre de voltes n_e es troba a partir d'un nombre enter pseudoaleatori uniformement distribuït entre 2 i 16.
- El radi de fil r és un nombre aleatori entre 0 i 3, arrodonit i al qual se li afegeix 1.
- El radi de l'hèlix R es troba a partir d'afegir al radi de fil r un nombre aleatori entre 0 i 30 arrodonit.
- El pas de la molla p és 1 més l'arrodoniment de 2,5 cops el radi del fil més un nombre aleatori entre 0 i 3.

Aquests valors es converteixen a *string* (cadena de caràcters) per a assignar-los a les propietats `Value` dels objectes `app.input_n`, `app.input_d` (es dobla r), `app.input_R` i `app.input_pas` respectivament.

S'utilitza el constructor de la classe `Material` (apartat 5.3), s'introdueixen en una llista els materials de la taula A.1 de l'annex A. Per exemple, el bronze al fòsfor s'introdueix de la següent manera:

```
|154 Material('Bronze al fòsfor', 43.1e3, 723.95, 8829.9, [0.804 0.489 0.196])
```

Aquests materials s'afegeixen als elements del desplegable per escollir el material. En l'atribut `Items` de l'objecte `app.tria_material` s'hi guarden els noms dels materials que es mostraran en el desplegable. D'altra banda, a l'atribut `ItemData` s'hi guarden els objectes de tipus

Material que contenen tota la informació necessària de la selecció que faci l'usuari. Això evita haver de buscar el material a partir del nom cada cop que se seleccioni una opció.

A continuació, es crea la superfície de la molla utilitzant la funció `renderitza_molla`, explicada en detall a l'apartat 5.5.2.

Després es crida a la funció `ompleredades` (apartat 5.5.5) que crea la taula amb la informació calculada de la molla.

Finalment, es torna a fer visibles tant `app.grafic` com `app.dades` assignant els seus atributs `Visible` a `'on'`.

5.2.2 Bucle principal

El funcionament nominal del programa es basa, com s'ha explicat a l'apartat 4.1, en un bucle que espera el proper esdeveniment i actua en conseqüència, actualitzant l'estat. Aquest comportament es pot veure esquematitzat a la figura 5.4.

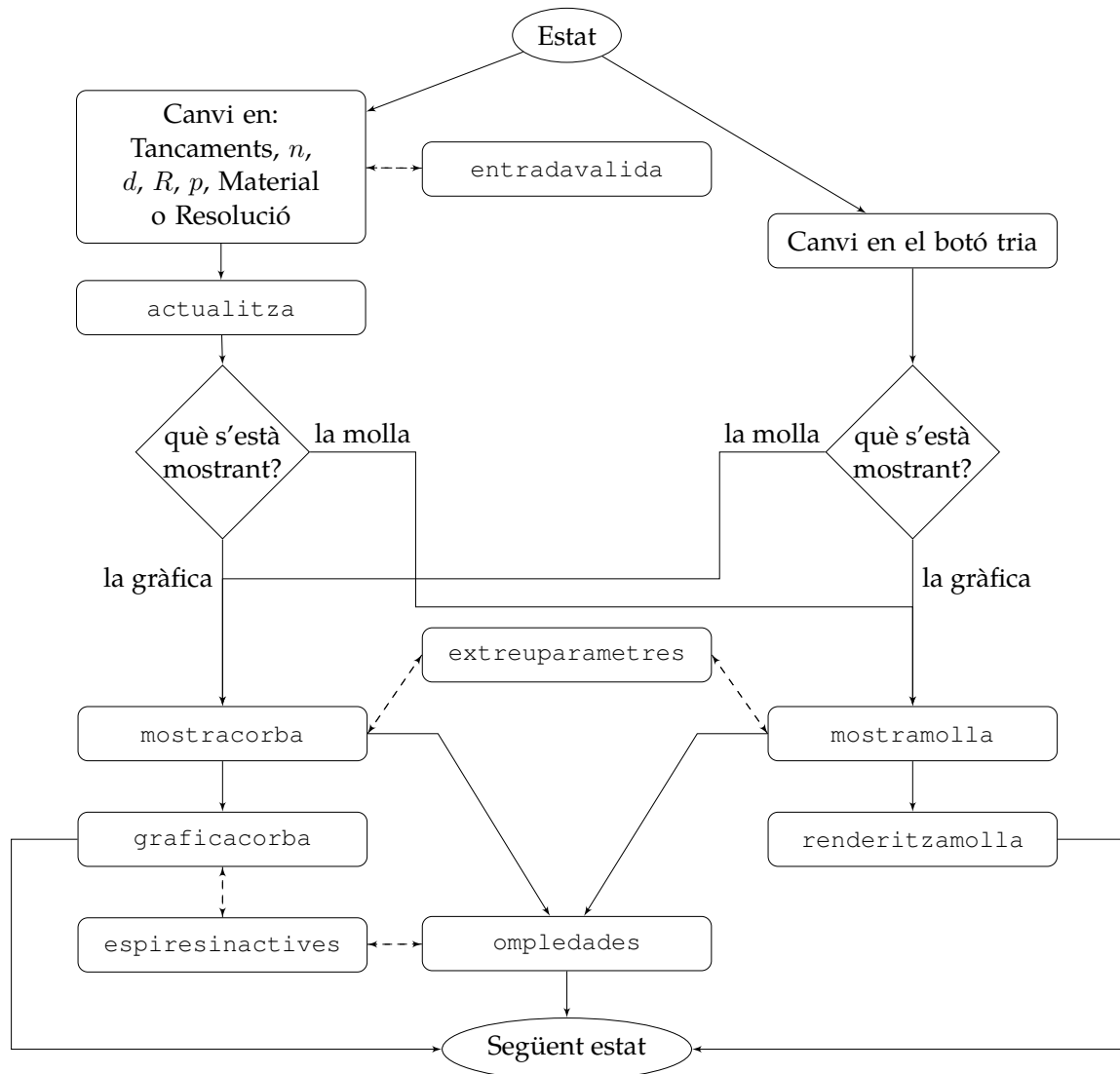


Figura 5.4: Esquema del funcionament del programa. Font: pròpia

Pel que fa als esdeveniments relacionats amb la interacció amb els quadres d'entrada de text dels paràmetres de la molla, el desplegable de materials, el tancament escollit i la resolució del renderitzat, en tots els seus gestors d'esdeveniment el primer que es fa és comprovar si hi ha hagut un canvi en el valor, comparant el valor actual amb l'anterior. Per exemple, el gestor d'esdeveniment del camp de text `app.input_n` és el següent (extret del codi font del fitxer `GUI_App.mlapp`, de l'annex B):

```

185 function input_n_Callback(app, event)
186     if entradavalida(app) & ~strcmp(event.Value, event.PreviousValue)
187         actualitza(app);
188     end
189 end
  
```

Si aquest condicional és assolit, es fa una crida a la funció `entradavalida` (apartat 5.4.1), que comprova que les dades introduïdes serveixin per a fer els càlculs. Si els gestors d'esdeveni-

ments tenen el vistiplau de la validació, la funció `actualitza` (apartat 5.5) s'encarrega de cridar les funcions `mostracorba` (apartat 5.5.3) o `mostramolla` (apartat 5.5.1) segons què hi hagi a la pantalla en aquell moment, per actualitzar-ho, com el seu nom indica.

Al contrari, si l'usuari prem el botó per escollir entre la gràfica o el renderitzat, passa al revés: es crida a `mostracorba` o `mostramolla` per canviar el que s'està ensenyant.

La funció `mostracorba` canvia les dades calculades a partir de la molla i genera un nou gràfic, i la funció `mostramolla` canvia les dades calculades a partir de la molla i genera un nou renderitzat.

5.3 Classe `Material`

Per a emmagatzemar els materials i les seves propietats s'utilitza la següent classe:

```
1 classdef Material
2     properties
3         Nom
4         G
5         sigma_adm
6         densitat
7         color
8     end
9     methods
10        function obj = Material(Nom, G, sigma_adm, densitat, color)
11            obj.Nom = Nom;
12            obj.G = G;
13            obj.sigma_adm = sigma_adm;
14            obj.densitat = densitat;
15            obj.color = color;
16        end
17    end
18 end
```

Tots els objectes d'aquesta classe tenen associats un nom, un mòdul elàstic a torsió, una tensió admissible, una densitat i un color per mostrar en el renderitzat de la molla d'aquell material.

L'únic mètode d'aquesta classe és el seu constructor, que pren com a arguments els valors de les propietats descrites anteriorment i les assigna adequadament a l'objecte.

5.4 Funcions auxiliars

5.4.1 Validació de l'entrada de dades de l'usuari

Abans de fer càlculs a partir de les dades que introdueix l'usuari, es comprova que les dades que aquest ha entrat són vàlides i permeten calcular una molla helicoidal real. El codi que fa això és el següent:

```
40 function [valids] = entradavalida(app)
41     in = app.input_n.Value;
42     id = app.input_d.Value;
43     iR = app.input_R.Value;
44     ipas = app.input_pas.Value;
```

```

45  if all(ismember(in, '0123456789+-.eEdD')) & floor(str2double(in)) == ...
        ceil(str2double(in)) & str2double(in) > espiresinactives(app)
46      app.input_n.BackgroundColor = [1 1 1];
47      valid_n = true;
48  else
49      app.input_n.BackgroundColor = [0.9 0 0];
50      valid_n = false;
51  end
52  if all(ismember(id, '0123456789+-.eEdD')) && str2double(id) >= 0
53      app.input_d.BackgroundColor = [1 1 1];
54      valid_d = true;
55  else
56      app.input_d.BackgroundColor = [0.9 0 0];
57      valid_d = false;
58  end
59  if all(ismember(iR, '0123456789+-.eEdD')) && str2double(iR) >= 0
60      app.input_R.BackgroundColor = [1 1 1];
61      valid_R = true;
62  else
63      app.input_R.BackgroundColor = [0.9 0 0];
64      valid_R = false;
65  end
66  if all(ismember(ipas, '0123456789+-.eEdD')) && str2double(ipas) >= 0
67      app.input_pas.BackgroundColor = [1 1 1];
68      valid_pas = true;
69  else
70      app.input_pas.BackgroundColor = [0.9 0 0];
71      valid_pas = false;
72  end
73  if str2double(ipas) > str2double(id) && str2double(iR) > str2double(id)/2
74      app.input_d.BackgroundColor = [1 1 1];
75      valid_colisio = true;
76  else
77      app.input_d.BackgroundColor = [0.9 0 0];
78      valid_colisio = false;
79  end
80  valids = valid_n & valid_d & valid_R & valid_pas & valid_colisio;
81  end

```

Es fan 5 validacions diferents en total, que s'han de complir alhora perquè es consideri vàlida l'entrada.

Si alguna d'aquestes 5 condicions no es compleix, l'espai on s'entra el text associat a aquella validació es pinta de color vermell per a marcar l'error i fer veure a l'usuari que s'ha de canviar aquella entrada. Les cinc coses que s'han de validar són les següents:

- El nombre d'espines n
Per considerar acceptable el text entrat per l'usuari al camp `input_n` han de coincidir tres situacions alhora: el text ha d'equivaldre a un nombre, ha de ser un nombre enter, i hi ha d'haver més espines que les espines inactives a causa del tancament escollit, com s'explica a l'apartat 3.4 (línia 6).
- El gruix del fil d
Ha de ser una cadena de caràcters numèrica (com no passa a l'exemple de la figura 5.5) i el gruix del fil ha de ser major que 0 (línia 13).

- El radi de l'hèlix R
Ha de ser una cadena de caràcters numèrica i el radi ha de ser major que 0 (línia 20).
- El pas de la molla p
Ha de ser una cadena de caràcters numèrica i el pas ha de ser major que 0 (línia 27).
- Col·lisió
Es comprova que el gruix del fil sigui més petit que el pas de la molla i que el radi de l'hèlix sigui més gran que el radi del fil (línia 34).

Nombre d'espines	<input type="text" value="12"/>
Gruix fil (mm)	<input style="background-color: red; color: white;" type="text" value="6-"/>
Radi hèlix (mm)	<input type="text" value="11"/>
Pas (mm)	<input type="text" value="11"/>
Material	<input type="text" value="Bronze al fòsfor"/>

Figura 5.5: Exemple d'entrada errònia de l'usuari. Font: pròpia.

Es considera una entrada vàlida si, i només si, el producte booleà d'aquests cinc elements és vertader (línia 80).

5.4.2 Obtenció dels paràmetres de la molla

Els valors dels quadres de text on l'usuari introdueix n , d , R i p són cadenes de caràcters. Per passar-ho a un nombre, a la funció `extreuparametres` es fa la conversió a nombre real amb `str2double`.

```

29 function [n, r, R, pas] = extreuparametres(app)
30     in = app.input_n.Value;
31     id = app.input_d.Value;
32     iR = app.input_R.Value;
33     ipas = app.input_pas.Value;
34     n = str2double(in);
35     r = str2double(id)/2;
36     R = str2double(iR);
37     pas = str2double(ipas);
38 end

```

5.4.3 Nombre d'espines inactives

El propòsit d'aquesta funció és determinar quantes espines inactives té la molla descrita per l'usuari, segons l'opció d'acabament que hi hagi marcada. S'utilitzen les equacions (1), (3), (5) i (7) de l'apartat 3.4.

```

1 function [noactives] = espinesinactives(app)
2   noactives = 0;
3   if app.ObertsButton.Value
4     noactives = 0;
5   elseif app.TancatsButton.Value
6     noactives = 2;
7   elseif app.DoblementtancatsButton.Value
8     noactives = 4;
9   elseif app.RectificatsButton.Value
10    noactives = 2;
11  end
12 end

```

5.5 Actualització de l'escenari gràfic

Un cop s'ha determinat que el canvi que l'usuari ha fet en els paràmetres que descriuen la molla és vàlid, amb la funció `actualitza` es decideix si el que cal actualitzar és el renderitzat de la molla (que es fa amb la funció `mostramolla` detallada a l'apartat 5.5.1) o la gràfica força-desplaçament (funció `mostracorba` detallada a l'apartat 5.5.3).

```

105 function actualitza(app)
106   if app.boto_tria.Value == 0
107     mostramolla(app);
108   else
109     mostracorba(app);
110   end
111 end

```

5.5.1 Funció `mostramolla`

Aquesta funció, mostrada a continuació, s'encarrega de què s'emplenin les dades de nou i que es creï la superfície de la molla.

```

83 function mostramolla(app)
84   cla(app.grafic, 'reset');
85   app.dades.Visible = 'off';
86   app.grafic.Visible = 'off';
87   [n, r, R, pas] = extreuparametres(app);
88   resolucio = app.slider_resolucio.Value;
89   renderitza_molla(app, n, r, R, pas, resolucio);
90   title(app.grafic, '');
91   app.dades.Visible = 'on';
92   app.grafic.Visible = 'on';
93   app.slider_resolucio.Visible = 'on';
94   ompladades(app, n, r, R, pas);
95 end

```

Primer de tot, en aquesta funció es reseteja l'espai gràfic on anirà la molla. Tot seguit s'oculten tant l'espai del gràfic com l'espai on es mostren les dades. Amb la funció `extreuparametres` (explicada a l'apartat 5.4.2) s'obtenen els valors numèrics de n , r , R i p . S'obté també el valor de la resolució de renderitzat desitjada consultant el valor `d'app.slider_resolucio.Value` (línia 88). Aquesta informació es passa a la funció `renderitza_molla` (apartat 5.5.2).

5.5.2 Renderitzat de la molla

Per obtenir la superfície tridimensional d'una molla helicoidal, s'ha de començar amb l'equació (38), parametritzada en t , que descriu una hèlix orientada en l'eix Z :

$$\vec{r}(t) = (R \cos t, R \sin t, ht), \quad t \in [0, 2\pi n] \quad (38)$$

on n és el nombre de voltes que té l'hèlix i h és el pas per angle girat, tal com mostra l'equació (39).

$$h = \frac{p}{2\pi} \quad (39)$$

La superfície que es desitja està formada pels punts equidistants a l'hèlix en el pla perpendicular a aquesta en cadascun dels seus punts. És a dir, s'han de descriure cercles centrats i perpendiculars a l'hèlix. Per trobar el pla on situar el cercle, s'utilitzen els vectors normal (equació (41)) i binormal (equació (42)), que és alhora perpendicular al vector tangent (equació (40)) i al vector normal.

$$\vec{t}(t) = \frac{d\vec{r}}{dt} = (-R \sin t, R \cos t, h) \quad (40)$$

$$\vec{n}(t) = \frac{\frac{d\vec{t}}{dt}}{\left\| \frac{d\vec{t}}{dt} \right\|} = (-\cos t, -\sin t, 0) \quad (41)$$

$$\vec{b}(t) = \frac{\vec{t} \times \vec{n}}{\left\| \vec{t} \right\|} = \frac{(h \sin t, -h \cos t, R)}{\sqrt{h^2 + R^2}} \quad (42)$$

Aleshores, en el pla definit per $\vec{n}(t)$ i $\vec{b}(t)$, s'hi pot descriure un cercle de radi r , parametritzat per u com es mostra a l'equació (43).

$$S(t, u) = \vec{r}(t) + r\vec{n}(t) \cos u + r\vec{b}(t) \sin u, \quad t \in [0, 2\pi n], \quad u \in [0, 2\pi] \quad (43)$$

La implementació en MATLAB® és la següent:

```

1 function [superficie] = renderitza_molla(app, n, r, R, pas, res)
2     h = pas/(2*pi);
3     [t, u] = meshgrid(linspace(0, 2*n*pi, res), linspace(0, 2*pi, res));
4     x = (R*cos(t)) - r*cos(t).*cos(u) + r*h*sin(t).*sin(u)/sqrt(R^2+h^2);
5     y = (R*sin(t)) - r*sin(t).*cos(u) - r*h*cos(t).*sin(u)/sqrt(R^2+h^2);
6     z = h*t + r*R*sin(u)/sqrt(R^2+h^2);

8     app.axes_gui.Position = [331,13,500,650];
9     hold(app.axes_gui);
10    superficie = surf(app.axes_gui, x, y, z, 'FaceColor',...
11    app.tria_material.Value.color, 'EdgeColor', 'none');

13    daspect(app.axes_gui, [1 1 1]); % Mantenir l'escala dels eixos a 1:1:1
14    view(app.axes_gui, [1 1 0.5]);
15    l = light(app.axes_gui, 'Position',[-0.4 0.2 0.9], 'Style','infinite');
```

```

16 l.Position = [-0.1 0.6 0.8];
17 material(app.axes_gui, 'metal'); % Acabat metàl·lic
18 rotate3d(app.axes_gui, 'on'); % Activa per defecte la interacció 3D

20 app.axes_gui.XColor = 'none'; % No ensenyar els eixos
21 app.axes_gui.YColor = 'none';
22 app.axes_gui.ZColor = 'none';
23 app.axes_gui.Color = 'none';
24 app.axes_gui.BackgroundColor = [1 1 1];
25 hold(app.axes_gui, 'off');
26 end

```

Depenent del paràmetre `res`, es generen més punts o menys en la superfície de la molla, permetent triar el nombre adient per la capacitat de computació de cada ordinador, jugant entre la resolució i la rapidesa del renderitzat. A la figura 5.6 es pot veure l'influència d'aquest paràmetre en el resultat.

A la línia 11 s'observa que el color amb que la funció `surf` crea la superfície de la molla és la propietat `color` del material escollit en el desplegable.

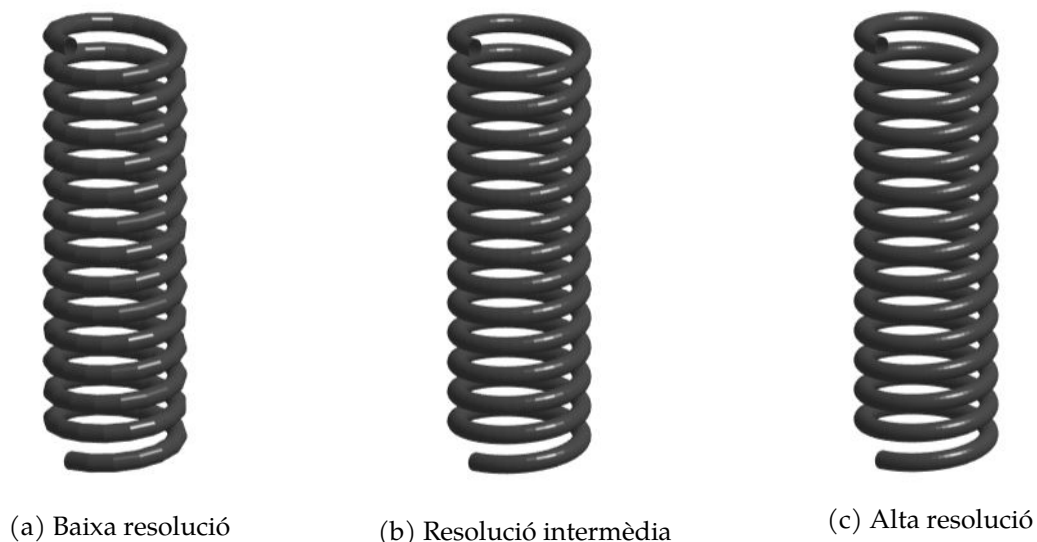


Figura 5.6: Renderitzats d'una molla amb $n = 14$, $r = 3$ mm, $R = 20$ mm i $p = 10$ mm. Font: pròpia.

S'utilitza també una il·luminació determinada (línies 15 i 16) que aporten realisme al renderitzat, així com la textura metàl·lica (línia 17).

5.5.3 Funció `mostracorba`

Aquesta funció, de manera anàloga a `mostramolla`, s'encarrega de que s'emplenin les dades de baix a l'esquerra de nou i que es creï la nova gràfica força-desplaçament de la molla.

```

97 function mostracorba(app)
98     cla(app.grafic, 'reset');
99     app.slider_resolucio.Visible = 'off';

```



```

100 [n, r, R, pas] = extreuparametres(app);
101 ompleredades(app, n, r, R, pas);
102 grafica_corba(app, n, r, R, pas);
103 end

```

Primer de tot, en aquesta funció es reseteja l'espai gràfic on anirà la gràfica. En aquest cas no és necessari deixar de fer visibles els elements gràfics, ja que la seva actualització és ràpida en comparació al renderitzat; el selector de resolució però, ja no té sentit que sigui visible. Utilitzant la funció `extreuparametres` (apartat 5.4.2) s'obtenen els valors numèrics de n , r , R i p . S'obté també el valor de la resolució de renderitzat desitjada. Aquesta informació es passa a la funció `grafica_corba`, explicada en detall a l'apartat 5.5.4.

5.5.4 Generació del gràfic de la corba de la molla

La funció que crea el gràfic força-desplaçament de la molla descrita per l'usuari és la següent:

```

1 function [grafic] = grafica_corba(app, n, r, R, pas)
2   app.grafic.Position = [331,13,500,650];
3   app.grafic.BackgroundColor = [1 1 1];
4   rotate3d(app.grafic, 'off');
5   set(app.grafic, 'defaulttextinterpreter', 'latex');

7   G = app.tria_material.Value.G;
8   sigma_adm = app.tria_material.Value.sigma_adm;
9   noact = espiresinactives(app);

11  k = G*r^4/(4*(n-noact)*R^3); % N/mm
12  def_max = (n-noact)*(pas - 2*r); % mm
13  def = linspace(0, def_max, 500); % mm
14  F = k.*def; % N

16  C = R/r;
17  Ks = 1+1/(2*C);

19  F_adm = pi*r^3*sigma_adm/(2*sqrt(3)*Ks*R);
20  fallida = 0;
21  if F_adm <= F(end)
22    def_adm = F_adm/k;
23    fallida = [def_adm F_adm];
24    F = F(F <= F_adm);
25    def = def(def <= def_adm);
26  end

28  grafic = plot(app.grafic, def, F, 'HandleVisibility','off');
29  hold(app.grafic, 'on');
30  if fallida
31    % posa el punt limit al plot
32    plot(app.grafic, fallida(1), fallida(2), '+r');
33    textadm = strcat('$\sigma_{adm} = ', num2str(sigma_adm), '\; \mathrm{MPa} ...
34    \; \; \; \mathrm{assolida}$');
35    legend(app.grafic, {textadm}, 'interpreter', 'latex', 'Location', ...
36    'southeast');
37  end

37  view(app.grafic, 2);
38  ylim(app.grafic, [0 max(F)*1.1]);
39  xlim(app.grafic, [0 max(def)*1.1]);

```

```

40 title(app.grafic, 'Corba carecterística de la molla');
41 ylabel(app.grafic, '$F$, \left [ N \right ]$', 'interpreter', 'latex');
42 xlabel(app.grafic, '$\delta$, \left [ mm \right ]$', 'interpreter', 'latex');
43 set(app.grafic, 'XGrid', 'on');
44 set(app.grafic, 'YGrid', 'on');
45 set(app.grafic, 'TickLabelInterpreter', 'latex');
46 textk = strcat('$k = ', num2str(k), ' \frac{N}{mm}$');
47 text(app.grafic, def(6), F(end-6), textk, 'Interpreter', 'latex');
48 pbaspect(app.grafic, [1 1.5 1]);

50 app.grafic.XColor = 'k';
51 app.grafic.YColor = 'k';
52 app.grafic.ZColor = 'k';
53 app.grafic.Color = 'white';
54 axis(app.grafic, 'auto');
55 hold(app.grafic, 'off');
56 end

```

Preliminarment, s'apaga la funció de la rotació 3D interactiva. A continuació s'extreuen el mòdul elàstic a torsió G i la tensió admissible σ_{adm} a partir de la selecció de material de l'usuari (`app.tria_material.Value`, línies 7 i 8), i el nombre d'espines inactives utilitzant la funció `espinesinactives` (línia 9).

Amb l'equació (22) es calcula la constant elàstica a la línia 11, i la deformació màxima a la línia 12 amb l'equació (23). Aleshores es generen 500 valors de desplaçament entre 0 i δ_{max} , i es troben els 500 valors de força que hi corresponen.

A la línia 19 es determina la força que fa que la tensió màxima en la molla equivalgui la tensió admissible amb l'equació (32). Si aquesta força és menor a l'últim valor de força calculat abans (que correspon a la deflactació màxima), s'arriba a la tensió admissible abans que totes les espines col·lisionin. Quan es dona aquest cas, es determina el valor del punt força-deformació on succeeix (per marcar-ho al gràfic després) i es retallen els 500 valors de força i de desplaçament per no incloure els que sobreixen.

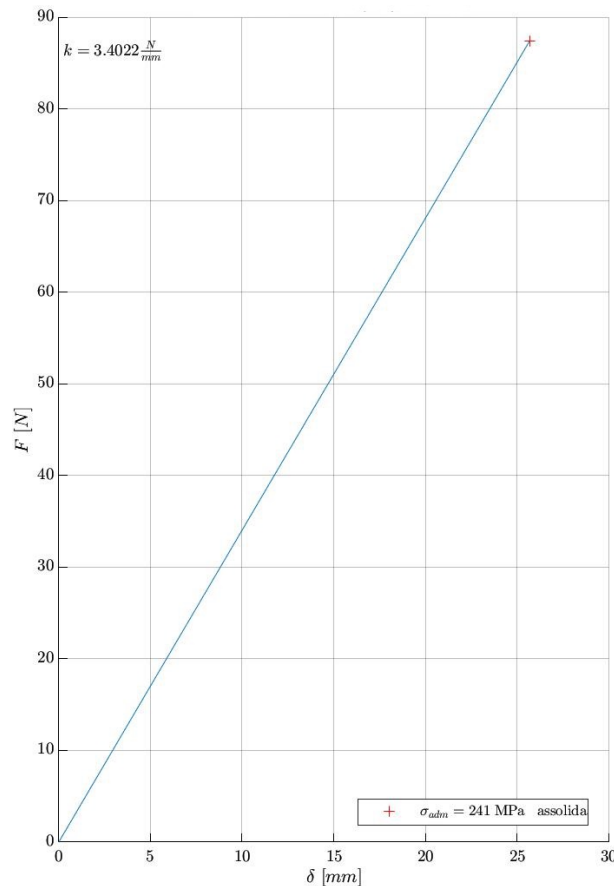


Figura 5.7: Corba característica d'una molla helicoidal d'Inconel 600 amb $n = 13$, $d = 4$ mm, $R = 19$ mm i $p = 6$ mm. Font: pròpia.

Finalment, es fa el gràfic amb la funció `plot`. Si hi ha el punt (δ_{adm}, F_{adm}) , es marca amb una creu vermella (línia 32) i s'hi afegeix una llegenda que ho indica (línies 33 i 34), com s'ensenya a la figura 5.7. A la línia 47 s'afegeix un text flotant amb el valor de la constant de la molla sobre el gràfic.

Les altres instruccions són per polir l'aspecte del gràfic d'una manera dinàmica, que no dependi dels valors exactes d'una sola corba. Els valors màxims dels dos eixos són un 10% més del valor màxim de la corba, el gràfic té un títol que diu "Corba característica de la molla", els eixos indiquen la magnitud i les unitats del que mesuren, entre d'altres coses.

5.5.5 Emplenar les dades

Per emplenar el quadre de dades calculades a partir de la molla descrita, s'utilitza la funció següent:

```

1 function omplendades(app, n, r, R, pas)
2     cla(app.dades);
3     set(app.dades, 'defaulttextinterpreter', 'latex');
4
5     G = app.tria_material.Value.G;
```

```

6  dens = app.tria_material.Value.densitat;
7  sigma_adm = app.tria_material.Value.sigma_adm;

9  noact = espiresinactives(app);

11 k = G*r^4/(4*(n-noact)*R^3); % N/mm
12 def_max = (n-noact)*(pas - 2*r); % mm
13 F_max = k*def_max; % N

15 C = R/r;
16 Ks = 1+1/(2*C);

18 F_adm = pi*r^3*sigma_adm/(2*sqrt(3)*Ks*R);
19 if F_adm <= F_max
20     def_adm = F_adm/k;
21     F_max = F_adm;
22     def_max = def_adm;
23 end

25 if app.ObertsButton.Value
26     L = (n-noact)*pas+2*r;
27 elseif app.TancatsButton.Value
28     L = (n-noact)*pas+2*r;
29 elseif app.DoblementtancatsButton.Value
30     L = (n-noact)*pas+4*r;
31 elseif app.RectificatsButton.Value
32     L = (n-noact)*pas;
33 end

35 Emax = F_max*def_max*1e-03/2;
36 m = dens*2*pi^2*n*R*r^2*1e-9;
37 omega = sqrt(k*1e3/m);
38 f = omega/(2*pi);
39 txt = {'$L= \: ' num2str(L) '\:\: mm$'},'', ...
40     ['$\delta_{max} = \: ' num2str(def_max) '\:\: mm$'],'', ...
41     ['$F_{max} = \: ' num2str(F_max,4) '\:\: N$'],'', ...
42     ['$k = \: ' num2str(k,4) '\:\: N/mm$'],'', ...
43     ['$E_{max} = \: ' num2str(Emax,4) '\:\: J$'],'',...
44     ['$m = \: ' num2str(m,2) '\:\: kg$'],'',...
45     ['$\omega_p = \: ' num2str(omega,4) '\:\: rad/s \: (' num2str(f,4) ...
46     '\:\: Hz$']});
46 hold(app.dades, 'on');
47 pbaspect(app.dades, [1 1 1]);
48 text(app.dades, 0.5, 0.48, txt, 'Interpreter', 'latex', 'FontSize', 16,...
49     'HorizontalAlignment','center');

51 app.dades.XColor = 'none';
52 app.dades.YColor = 'none';
53 app.dades.ZColor = 'none';
54 app.dades.Color = 'none';
55 hold(app.dades, 'off');
56 end

```

Com a primer pas, s'esborra tot el que hi ha en l'espai gràfic `app.dades`. A continuació s'extreuen el mòdul elàstic a torsió G i la tensió admissible σ_{adm} a partir de la selecció de material de l'usuari (`app.tria_material.Value`, línies 5 i 6), i el nombre d'espores inactives utilitzant la funció `espiresinactives`.

Amb l'equació (22) es calcula la constant elàstica a la línia 11, i la deformació màxima de la línia 12 amb l'equació (23), i multiplicant-lo per la constant elàstica k a la línia 13 es troba el valor de la força màxima.

A la línia 18 es determina la força amb que la tensió màxima en la molla equival la tensió admissible amb l'equació (32). Si aquesta força és menor a la força màxima (que correspon a la deflexió màxima), s'arriba a la tensió admissible abans que totes les espirals col·lisionin. Quan es dona aquest cas, la força màxima passa a ser la força admissible, i anàlogament amb la deformació (línies 21 i 22).

Les equacions (2), (4), (6) i (8) s'utilitzen entre la línia 25 i 33 per a determinar la longitud de la molla en buit. L'energia potencial elàstica màxima que es pot emmagatzemar en aquesta molla es busca a partir de l'equació (34) a la línia 35, la massa de la molla a la línia 36 amb l'equació (35) i la freqüència pròpia (en pulsacions per segon i en Hertz) amb les equacions (36) i (37) a les línies 37 i 38.

$$L = 82 \text{ mm}$$

$$\delta_{max} = 25.4747 \text{ mm}$$

$$F_{max} = 86.67 \text{ N}$$

$$k = 3.402 \text{ N/mm}$$

$$E_{max} = 1.104 \text{ J}$$

$$m = 0.16 \text{ kg}$$

$$\omega_p = 144 \text{ rad/s (22.92 Hz)}$$

Figura 5.8: Dades d'una molla helicoidal d'Inconel 600 amb $n = 13$, $d = 4 \text{ mm}$, $R = 19 \text{ mm}$ i $p = 6 \text{ mm}$. Font: pròpia.

Els valors de L_0 , δ_{max} , F_{max} , k , E_{max} , ω i f es formaten en una cadena de caràcters (línies 39-45) que MATLAB® processarà amb un interpretador de L^AT_EX (línia 48). L'ús d'aquest interpretador permet estilar el text amb subíndexs, com es veu a la figura 5.8.

5.6 Llançament del producte

Un cop finalitzada l'etapa de disseny, del programa se'n fa un instal·lador que inclou una compilació del programa i MATLAB® Runtime [13], que és un conjunt de llibreries independents que fan que un programari MATLAB® sigui executable sense necessitat de tenir MATLAB® instal·lat a l'ordinador.

Seguint les instruccions de l'*Application Compiler* [12] al qual s'accedeix a través d'un accés di-

recte del menú superior de l'*App Designer*, s'introdueixen nom, autoria i descripció de l'aplicació i les icones.

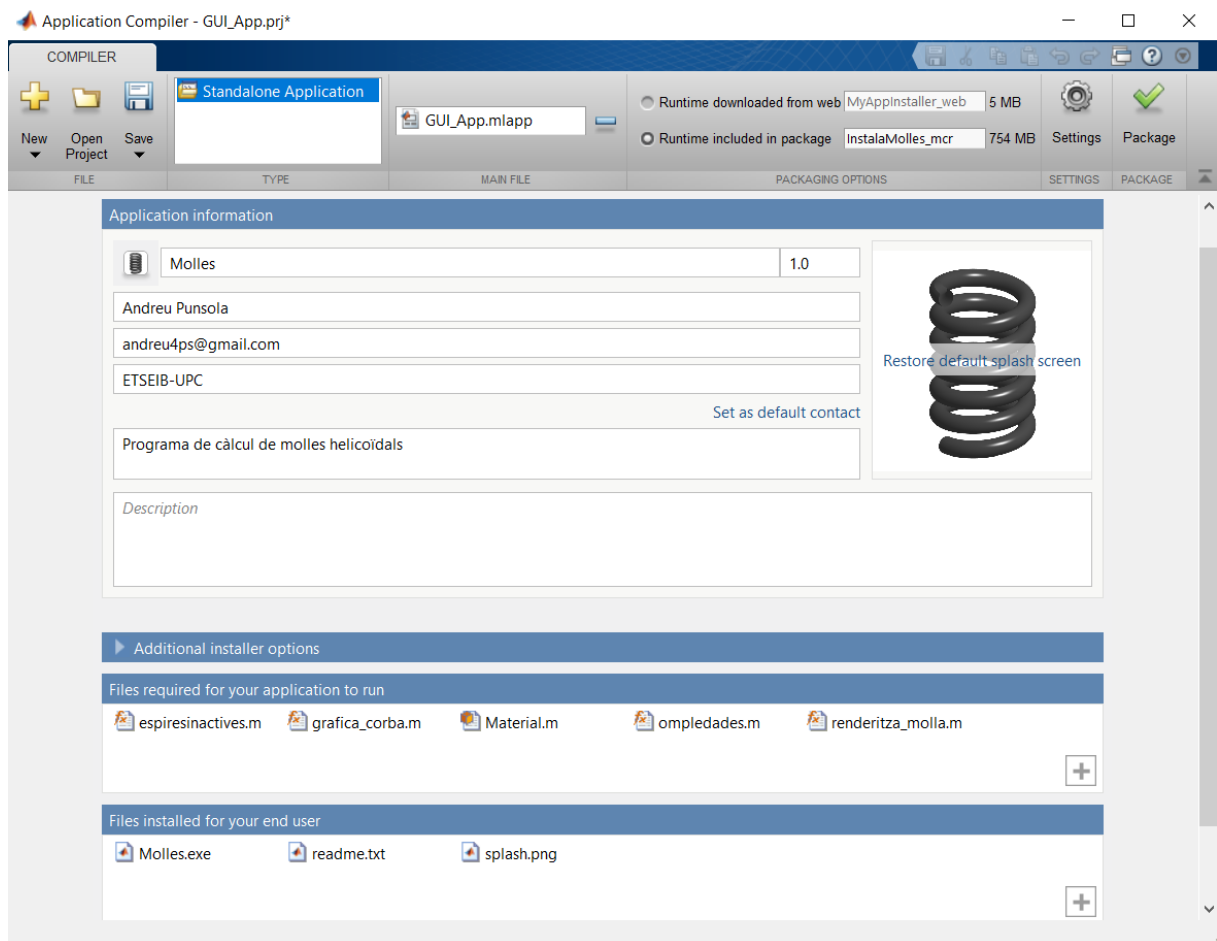


Figura 5.9: Menú del MATLAB® *Application Compiler*. Font: pròpia.

Si es vol, es pot escollir que el *MATLAB® Runtime* es baixi d'internet durant la instal·lació o que l'instal·lador el porti ja inclòs, cosa que n'augmenta la mida.

L'*Application Compiler* detecta que els fitxers `espinesinactives.m`, `grafica_corba.m`, `Material.m`, `ompladades.m` i `renderitza_molla.m` són necessaris per al funcionament del programa i els inclou automàticament.

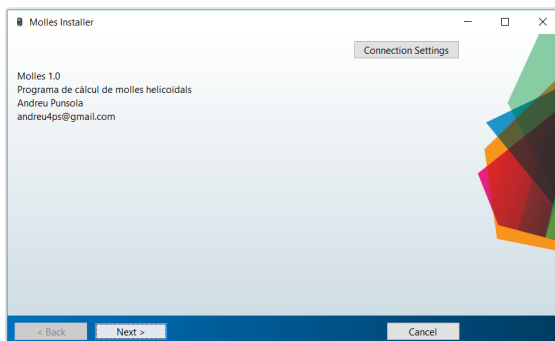
S'hi afegeix també una imatge (figura 5.10) que es mostra quan el programa s'engega de zero, que s'anomena pantalla de presentació (*splash screen* en anglès) [19]. Aquesta imatge s'ha generat utilitzant una variació de la funció `renderitz_molla` (apartat 5.5.2).



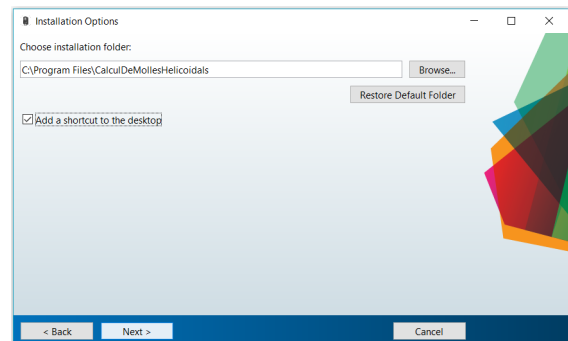
Andreu Punsola
Programa de càlcul de molles helicoidals

Figura 5.10: Pantalla de presentació del programa. Font: pròpia.

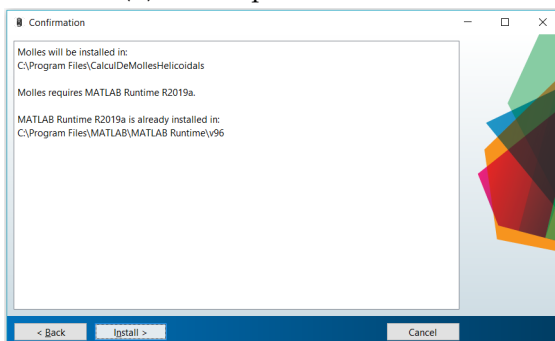
Quan tots els paràmetres són els desitjats, en clicar al botó *Package* (empaquetar) i esperar una estona, es genera l'arxiu *InstalaMolles_mcr.exe*, de 918 Mb. La mida d'aquest instal·lador es pot reduir si no s'inclou el *MATLAB® Runtime*; en aquest cas, la màquina on s'instal·li ha de tenir connexió a Internet. L'instal·lador conté tot el necessari per tenir i executar el programa, i el procés d'instal·lació és ben senzill com es mostra a la figura 5.11. En aquest cas, l'arxiu *InstalaMolles_web.exe* pesa tan sols 2,5 Mb.



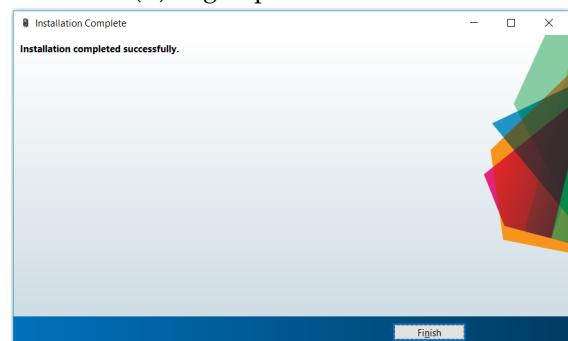
(a) Primer pas d'instal·lació



(b) Segon pas d'instal·lació



(c) Tercer pas d'instal·lació



(d) Quart pas d'instal·lació

Figura 5.11: Procés d'instal·lació del programa. Font: pròpia.

En l'executar l'instal·lador, s'obre la finestra de la figura 5.11a, que introdueix el procés d'instal·lació i mostra la informació sobre l'aplicació que s'ha introduït al compilador (figura 5.9). Tot seguit, a la figura 5.11b es pot escollir el directori on s'instal·larà el programa, i a la 5.11c es resumeix tot el que l'instal·lador farà. El següent pas és clicar al botó d'instal·lar i esperar a que aparegui la pantalla de la figura 5.11d. Aleshores el programa ja està instal·lat i es pot trobar en el navegador de programari del sistema operatiu.

6 Pressupost

6.1 Cost de personal

Les hores dedicades a aquest projecte es desgranen en dos tasques principals, programar i fer aquesta memòria. S'ha tingut en compte un preu/hora de 25€/h.

Tasca	Hores	Preu
Programa	250	6250€
Memòria	170	4250€
Total	420	10500€

Taula 6.1: Cost de personal

6.2 Cost de llicències

L'únic programari privat utilitzat és el MATLAB® de Mathworks, del qual l'estudiantat de la Universitat Politècnica de Catalunya disposa de llicència gratuïta.

6.3 Cost total associat

El cost total d'aquest projecte és de 10500 euros.

7 Impacte ambiental

Es redueix l'ús de material i hores de funcionament de maquinària per a crear prototips de molles helicoidals per assajar-los i determinar-ne els paràmetres que el programa ja ofereix.

Dit d'una altra manera, poder preveure la massa, la constant elàstica, i altres propietats d'una molla helicoidal sense haver de fabricar-la, evita haver de fabricar prototips, que solen fer-se a mà (que és més perillós que amb màquines de control numèric, com s'ha explicat a l'apartat 3.2). Tot i que els prototips fallits es poden reciclar i recuperar gairebé la totalitat del material utilitzat, és un procés energèticament demandant.

Conclusions

En el present Treball de Final de Grau s'ha assolit l'objectiu de crear un programa d'escriptori per calcular molles helicoidals, pensat per al seu ús en el disseny de màquines, per ajudar a l'hora de dimensionar-les, facilitant la feina en evitar haver de prototipar diferents combinacions de dimensions de la molla i de materials

En primer lloc, s'han deduït les equacions que en regeixen els paràmetres rellevants (la constant elàstica, la deformació màxima, l'energia potencial elàstica que pot emmagatzemar, etcètera) i la seva relació amb la geometria i el material de la molla, a partir de l'estudi de l'elasticitat i la resistència de materials de les molles helicoidals.

En segon lloc, s'ha creat una interfície gràfica interactiva, on es descriu la molla i s'obtenen diferents resultats útils per al disseny mecànic, com la corba força-deformació de la molla o un renderitzat tridimensional de la forma de la molla descrita.

Com a continuació d'aquest treball, es podria estendre el programa per calcular molles helicoidals de pas variable, radi d'hèlix variable o gruix de fil variable. Per a fer-ho, caldria deduir la relació entre força i desplaçament en funció de com evolucionen el radi de l'hèlix, el pas i el gruix del fil al llarg de la directriu de la molla.

Cal tenir en compte que caldria també pensar una manera de fer perquè l'usuari descrigui l'evolució del radi de l'hèlix, el pas i el gruix del fil al llarg de la directriu de la molla d'una manera intuïtiva de plantejar, fàcil d'introduir al programa i senzilla de convertir programàticament a l'objecte matemàtic que representa.

Agraïments

A l'Assemblea d'Estudiants, pel suport.

Al professor Francesc Roure, pel seu ajut a l'hora de trobar propietats d'acers.

Al professor Antonio Mateo, per referir a la base de dades *MatWeb* [14].

Annex

A Materials comuns per a molles

La taula següent s’ha fet a partir de la taula de materials per a fil de molles de la web d’*Spring-I-Pedia* [10]. Els valors de les tensions admissibles s’han calculat a partir dels coeficients recomanats a la taula 7.2 de *Comprehensive Spring Design* [20] multiplicats per la tensió mínima de ruptura mitjana de cada metall de la taula 7.3 de la mateixa publicació. En els aliatges on no hi havia la tensió mínima de ruptura, s’han utilitzat els valors proporcionats per la base de dades en línia *MatWeb* [14].

Material	Nom comercial	Composició química nominal	Densitat, ρ (g/cm ³)	Tensió admissible, σ_{adm} (MPa) [20]	Mòdul de Young, E (GPa)	Mòdul de cisallament, G (GPa)	Temperatura màxima de funcionament (°C)	Mètode de fabricació	Ús principal
<i>Music wire</i>	DIN 2076-84 DIN 17223/1-84 ISO 8458-1 ISO 8458-2 AISI 1086	C 0,7 - 1% Mn 0,2 - 0,6%	7,8611	975,81	206,84	82,73 (fins a 0.254 cm) 79,29 (més de 0.254 cm)	121	Estirat en fred en la qualitat més alta, fil d’alt carboni	Bon acabat superficial
<i>Hard Drawn MB</i>	DIN 177-88 DIN 2076-84 DIN 17223/1-84 ISO 2232-90 ISO 6984-90 AISI 1065	C 0,45 - 0,85% Mn 0,6 - 1,3%	7,8611	638,48	206,84	79,29	121	Estirat en fred	Usos d’esforç mitjà. Es pot galvanitzar fàcilment
<i>Oil Tempered MB</i>	DIN 17223/2-90 ISO 8458-1 ISO 8458-2 ASTM A229	C 0,55 - 0,85% Mn 0,6 - 1,2%	7,8611	752,42	206,84	79,29	121	Estirat en fred i tractat tèrmicament abans de la fabricació	Lleugeres escates superficials. Cable de molles de propòsit general
<i>Oil Tempered Chrome Silicon</i>	ISO 8458-1 ISO 8458-3	C 0,51 - 0,59% Cr 0,6 - 0,8% Si 1,2 - 1,6%	7,8611	837,74	206,84	79,29	246	Estirat en fred i tractat tèrmicament abans de la fabricació	Utilitzat en càrregues d’impacte i temperatures de funcionament moderadament elevades
<i>Oil Tempered Chrome Vanadium</i>	ASTM A231 UNE F1430 DIN 1.8159	C 0,48 - 0,53% Mn 0,7 - 0,9% Cr 0,8 - 1,1% V mín. 0,15%	7,8611	760,17	206,84	79,29	218	Estirat en fred i tractat tèrmicament abans de la fabricació	Utilitzat en càrregues d’impacte i temperatures de funcionament moderadament elevades
Acer inoxidable 302/304	ASTM A313	Cr 17 - 19% Ni 8-10%	7,9165	561,08	193,05	68,95	288	Estirat en fred	Propòsits generals. Resistència a la corrosió i la temperatura
Acer inoxidable 316	ASTM A313	Cr 16 - 18% Ni 10-14% M 2-3%	7,9165	641,24	193,05	68,95	288	Estirat en fred	Propòsits generals. Més resistència a la corrosió i la temperatura que el 302
Acer inoxidable 17-7 PH	AMS 5678	Cr 16 - 18% Ni 6,5-7,5% Mo 0,75-1,5%	7,8057	932,38	203,4	75,84	343	Estirat en fred i endurit per envelleiment tèrmic	Alta resistència i propòsits generals de resistència a la corrosió
Bronze al fòsfor	ASTM B159	Cu 94 - 96% Sn 4-6%	8,8576	379,23	103,42	43,1	93	Estirat en fred	Bona resistència a la corrosió i conductivitat elèctrica

Material	Nom comercial	Composició química nominal	Densitat, ρ (g/cm ³)	Tensió admissible, σ_{adm} (MPa) [20]	Mòdul de Young, E (GPa)	Mòdul de cisallament, G (GPa)	Temperatura màxima de funcionament (°C)	Mètode de fabricació	Ús principal
Coure al beril·li	ASTM B197	Cu 98% Sn 2%	8,2486	465,41	127,55	48,26	204	Estirat en fred	Bona resistència a la corrosió i conductivitat elèctrica. Bona resistència física
Monel 400	AMS 7233	Ni 66% Cu 31,5%	8,8299	200 [14]	179,26	65,5	232	Estirat en fred	Bona resistència a la corrosió a temperatures moderadament elevades
Monel K 500	QQ-N-286	Ni 65% Cu 29,5%	8,74	392 [14]	179,26	65,5	288	-	Excel·lent resistència a la corrosió a temperatures moderadament elevades
Inconel 600	AMS B166	Ni 76% Cr 15,8% Fe 7,2%	8,4147	241 [14]	213,74	75,84	371	Estirat en fred	Bona resistència a la corrosió a temperatures moderadament elevades
Inconel 718	AMS B637	Ni 52,5% Cr 18,6% Fe 18,5%	8,2486	275,6 [14]	199,95	77,22	593	Estirat en fred i endurit per envelleiment tèrmic	Bona resistència a la corrosió a temperatures moderadament elevades
Inconel X750	AMS 5524	Ni 73% Cr 15% Fe 6,75%	8,2486	474 [14]	213,74	82,74	399-593	Estirat en fred i endurit per envelleiment tèrmic	Bona resistència a la corrosió a temperatures moderadament elevades
Elgiloy	AMS 5833	Co 39-41% Cr 19-21% Ni 15-16% Fe romanent	8,3	320 [14]	220,63	82,73	121	Estirat en fred i endurit per envelleiment tèrmic	Excel·lent resistència a la corrosió
NiSpan C	AMS 5225	Ni 41-43,5% Cr 4,9-5,75% Ti 2,2-2,75% Al 0,3-0,8% Fe romanent	8,1379	506,67 [14]	165,47-199,95	62,05-68,95	66	Endurable per envelleiment tèrmic	Aliatge de mòdul elàstic constant
Hastelloy (C276)	ASTM B574	Cr 14,5-16,5% Mo 15-17% W 3-4,5% Fe 4-7% Ni romanent	8,89	206,67 [14]	211,67	81,36	371	-	Alta resistència a la corrosió

Taula A.1: Propietats dels metalls més comuns per a molles

B Codi font

El codi del fitxer principal, GUI_App.mlapp, és el següent:

```

1  classdef GUI_App < matlab.apps.AppBase
2
3      % Properties that correspond to app components
4      properties (Access = public)
5          programa                matlab.ui.Figure
6          grafic                  matlab.ui.control.UIAxes
7          dades                   matlab.ui.control.UIAxes
8          text_n                  matlab.ui.control.Label
9          input_n                 matlab.ui.control.EditField
10         text_d                  matlab.ui.control.Label
11         input_d                 matlab.ui.control.EditField
12         text_R                  matlab.ui.control.Label
13         input_R                 matlab.ui.control.EditField
14         text_pas                matlab.ui.control.Label
15         input_pas               matlab.ui.control.EditField
16         tria_material           matlab.ui.control.DropDown
17         text_material           matlab.ui.control.Label
18         slider_resolucio        matlab.ui.control.Slider
19         text_opcionsfinal       matlab.ui.control.Label
20         Tancaments              matlab.ui.container.ButtonGroup
21         TancatsButton           matlab.ui.control.RadioButton
22         ObertsButton            matlab.ui.control.RadioButton
23         RectificatsButton       matlab.ui.control.RadioButton
24         DoblementtancatsButton  matlab.ui.control.RadioButton
25         boto_tria               matlab.ui.control.StateButton
26     end
27
28     methods (Access = private)
29         function [n, r, R, pas] = extreuparametres(app)
30             in = app.input_n.Value;
31             id = app.input_d.Value;
32             iR = app.input_R.Value;
33             ipas = app.input_pas.Value;
34             n = str2double(in);
35             r = str2double(id)/2;
36             R = str2double(iR);
37             pas = str2double(ipas);
38         end
39
40         function [valids] = entradavalida(app)
41             in = app.input_n.Value;
42             id = app.input_d.Value;
43             iR = app.input_R.Value;
44             ipas = app.input_pas.Value;
45             if all(ismember(in, '0123456789+-.eEdD')) & floor(str2double(in)) == ...
46                 ceil(str2double(in)) & str2double(in) > espiresinactives(app)
47                 app.input_n.BackgroundColor = [1 1 1];
48                 valid_n = true;
49             else
50                 app.input_n.BackgroundColor = [0.9 0 0];
51                 valid_n = false;
52             end
53             if all(ismember(id, '0123456789+-.eEdD')) && str2double(id) >= 0
54                 app.input_d.BackgroundColor = [1 1 1];
55                 valid_d = true;

```

```

55     else
56         app.input_d.BackgroundColor = [0.9 0 0];
57         valid_d = false;
58     end
59     if all(ismember(iR, '0123456789+-.eEdD')) && str2double(iR) >= 0
60         app.input_R.BackgroundColor = [1 1 1];
61         valid_R = true;
62     else
63         app.input_R.BackgroundColor = [0.9 0 0];
64         valid_R = false;
65     end
66     if all(ismember(ipas, '0123456789+-.eEdD')) && str2double(ipas) >= 0
67         app.input_pas.BackgroundColor = [1 1 1];
68         valid_pas = true;
69     else
70         app.input_pas.BackgroundColor = [0.9 0 0];
71         valid_pas = false;
72     end
73     if str2double(ipas) > str2double(id) && str2double(iR) > str2double(id)/2
74         app.input_d.BackgroundColor = [1 1 1];
75         valid_colisio = true;
76     else
77         app.input_d.BackgroundColor = [0.9 0 0];
78         valid_colisio = false;
79     end
80     valids = valid_n & valid_d & valid_R & valid_pas & valid_colisio;
81 end

82
83 function mostramolla(app)
84     cla(app.grafic, 'reset');
85     app.dades.Visible = 'off';
86     app.grafic.Visible = 'off';
87     [n, r, R, pas] = extreuparametres(app);
88     resolucio = app.slider_resolucio.Value;
89     renderitza_molla(app, n, r, R, pas, resolucio);
90     title(app.grafic, '');
91     app.dades.Visible = 'on';
92     app.grafic.Visible = 'on';
93     app.slider_resolucio.Visible = 'on';
94     omplredades(app, n, r, R, pas);
95 end

96
97 function mostracorba(app)
98     cla(app.grafic, 'reset');
99     app.slider_resolucio.Visible = 'off';
100    [n, r, R, pas] = extreuparametres(app);
101    omplredades(app, n, r, R, pas);
102    grafica_corba(app, n, r, R, pas);
103 end

104
105 function actualitza(app)
106     if app.boto_tria.Value == 0
107         mostramolla(app);
108     else
109         mostracorba(app);
110     end
111 end
112 end

113
114 % Callbacks that handle component events

```



```

115 methods (Access = private)
117 % Code that executes after component creation
118 function GUI_OpeningFcn(app)
119     screenSize = get(groot, 'ScreenSize');
120     screenWidth = screenSize(3);
121     screenHeight = screenSize(4);
122     esquerra = screenWidth*0.2;
123     avall = screenHeight*0.1;
124     app.programa.Position = [esquerra avall 840 700];

126 % Omplir els camps
127 app.grafic.Visible = 'off';
128 app.dades.Visible = 'off';

130 app.ObertsButton.Value = 1;
131 app.TancatsButton.Value = 0;
132 app.RectificatsButton.Value = 0;
133 app.DoblementtancatsButton.Value = 0;

135 n = 1 + randi(15);
136 r = 1 + round(rand()*3);
137 R = r + round(rand()*30);
138 pas = 1 + round(2.5*r + rand()*3);

140 app.input_n.Value = num2str(n);
141 app.input_d.Value = num2str(2*r);
142 app.input_R.Value = num2str(R);
143 app.input_pas.Value = num2str(pas);

145 global materials;
146 materials = {Material('Music wire', 82.73e3, 975.81, 7861.1, [0.5 0.5 ...
147     0.5]),...
148     Material('Hard drawn MB', 79.29e3, 638.48, 7861.1, [0.5 0.5 0.5]),...
149     Material('Oil tempered MB', 79.29e3, 752.42, 7861.1, [0.5 0.5 0.5]),...
150     Material('Oil tempered Cr-Si', 79.29e3, 837.74, 7861.1, [0.5 0.5 ...
151     0.5]),...
152     Material('Oil tempered Cr-V', 79.29e3, 760.17, 7861.1, [0.5 0.5 0.5]),...
153     Material('Inox 302/304', 68.95e3, 561.08, 7916.5, [0.5 0.5 0.5]),...
154     Material('Inox 316', 68.95e3, 641.24, 7916.5, [0.5 0.5 0.5]),...
155     Material('Inox 17-7PH', 75.84e3, 932.38, 7805.7, [0.5 0.5 0.5]),...
156     Material('Bronze al fòsfor', 43.1e3, 723.95, 8829.9, [0.804 0.489 ...
157     0.196]),...
158     Material('Coure al beril·li', 48.26e3, 465.41, 8248.6, [0.533 0.177 ...
159     0.09]),...
160     Material('Monel 400', 65.5e3, 200, 8829.9, [0.7922 0.8 0.8078]),...
161     Material('Monel K 500', 65.5e3, 392, 8740, [0.7922 0.8 0.8078]),...
162     Material('Inconel 600', 75.84e3, 241, 8414.7, [0.557 0.573 0.58]),...
163     Material('Inconel 718', 77.22e3, 275.6, 8248.6, [0.557 0.573 0.58]),...
164     Material('Inconel X750', 82.93e3, 474, 8248.6, [0.557 0.573 0.58]),...
165     Material('Elgiloy', 82.73e3, 320, 8300, [0.902 0.745 0.541]),...
166     Material('NiSpan C', 65.5e3, 506.67, 8137.9, [0.7922 0.8 0.8078]),...
167     Material('Hastelloy', 81.36e3, 206.67, 8890, [0.7922 0.8 0.8078])};
168 for i = 1:length(materials)
169     app.tria_material.Items{i} = materials{i}.Nom;
170     app.tria_material.ItemsData{i} = materials{i};
171 end
172 app.tria_material.Value = app.tria_material.ItemsData{randi([1 ...
173     length(materials)])};

```

```
170     renderitza_molla(app, n, r, R, pas, 500);
171     omplredades(app, n, r, R, pas);

173     app.grafic.Visible = 'on';
174     app.dades.Visible = 'on';
175     end

177     % Value changed function: input_d
178     function input_d_Callback(app, event)
179         if entradavalida(app) & ~strcmp(event.Value, event.PreviousValue)
180             actualitza(app);
181         end
182     end

184     % Value changed function: input_n
185     function input_n_Callback(app, event)
186         if entradavalida(app) & ~strcmp(event.Value, event.PreviousValue)
187             actualitza(app);
188         end
189     end

191     % Value changed function: input_pas
192     function input_pas_Callback(app, event)
193         if entradavalida(app) & ~strcmp(event.Value, event.PreviousValue)
194             actualitza(app);
195         end
196     end

198     % Value changed function: input_R
199     function input_R_Callback(app, event)
200         if entradavalida(app) & ~strcmp(event.Value, event.PreviousValue)
201             actualitza(app);
202         end
203     end

205     % Value changed function: slider_resolucio
206     function slider_resolucio_Callback(app, event)
207         if event.Value ~= event.PreviousValue
208             actualitza(app);
209         end
210     end

212     % Value changed function: tria_material
213     function tria_material_Callback(app, event)
214         if ~strcmp(event.Value, event.PreviousValue)
215             actualitza(app);
216         end
217     end

219     % Value changed function: boto_tria
220     function boto_triaValueChanged(app, event)
221         value = app.boto_tria.Value;
222         if value == 0
223             app.boto_tria.Text = 'Mostra la gràfica';
224             app.slider_resolucio.Value = 300;
225             mostramolla(app);
226         else
227             app.boto_tria.Text = 'Mostra la molla';
228             mostracorba(app);
229         end
    end
```

```
230     end
231
232     % Selection changed function: Tancaments
233     function TancamentsSelectionChanged(app, event)
234         if entradavalida(app)
235             actualitza(app);
236         end
237     end
238 end
239
240 % Component initialization
241 methods (Access = private)
242
243 % Create UIFigure and components
244 function createComponents(app)
245
246     % Create programa and hide until all components are created
247     app.programa = uifigure('Visible', 'off');
248     app.programa.Color = [1 1 1];
249     app.programa.Position = [400 90 840 706];
250     app.programa.Name = 'Programa de càlcul de molles - Andreu Punsola';
251
252     % Create grafic
253     app.grafic = uiaxes(app.programa);
254     app.grafic.PlotBoxAspectRatio = [1 1.39852398523985 1];
255     app.grafic.FontSize = 13;
256     app.grafic.GridColor = 'none';
257     app.grafic.XColor = [0 0 0];
258     app.grafic.YColor = [0 0 0];
259     app.grafic.ZColor = [0 0 0];
260     app.grafic.Color = 'none';
261     app.grafic.NextPlot = 'replaceall';
262     app.grafic.Projection = 'perspective';
263     app.grafic.Visible = 'off';
264     app.grafic.BackgroundColor = [1 1 1];
265     app.grafic.Position = [331 7 500 676];
266
267     % Create dades
268     app.dades = uiaxes(app.programa);
269     app.dades.PlotBoxAspectRatio = [1.01298701298701 1 1];
270     app.dades.FontSize = 13;
271     app.dades.NextPlot = 'replace';
272     app.dades.BackgroundColor = [1 1 1];
273     app.dades.Position = [16 29 283 275];
274
275     % Create text_n
276     app.text_n = uilabel(app.programa);
277     app.text_n.HorizontalAlignment = 'center';
278     app.text_n.VerticalAlignment = 'top';
279     app.text_n.FontSize = 11;
280     app.text_n.Position = [27 598 112 31];
281     app.text_n.Text = {''; 'Nombre d''espires'};
282
283     % Create input_n
284     app.input_n = uieditfield(app.programa, 'text');
285     app.input_n.ValueChangedFcn = createCallbackFcn(app, @input_n_Callback, ...
286         true);
287     app.input_n.HorizontalAlignment = 'center';
288     app.input_n.FontSize = 11;
289     app.input_n.Position = [159 607 120 22];
```

```
290     % Create text_d
291     app.text_d = uilabel(app.programa);
292     app.text_d.HorizontalAlignment = 'center';
293     app.text_d.VerticalAlignment = 'top';
294     app.text_d.FontSize = 11;
295     app.text_d.Position = [27 543 112 31];
296     app.text_d.Text = {''; 'Gruix fil (mm)'};

298     % Create input_d
299     app.input_d = uieditfield(app.programa, 'text');
300     app.input_d.ValueChangedFcn = createCallbackFcn(app, @input_d_Callback, ...
301         true);
302     app.input_d.HorizontalAlignment = 'center';
303     app.input_d.FontSize = 11;
304     app.input_d.Position = [159 552 120 22];

306     % Create text_R
307     app.text_R = uilabel(app.programa);
308     app.text_R.HorizontalAlignment = 'center';
309     app.text_R.VerticalAlignment = 'top';
310     app.text_R.FontSize = 11;
311     app.text_R.Position = [27 488 112 31];
312     app.text_R.Text = {''; 'Radi hèlix (mm)'};

314     % Create input_R
315     app.input_R = uieditfield(app.programa, 'text');
316     app.input_R.ValueChangedFcn = createCallbackFcn(app, @input_R_Callback, ...
317         true);
318     app.input_R.HorizontalAlignment = 'center';
319     app.input_R.FontSize = 11;
320     app.input_R.Position = [159 497 120 22];

322     % Create text_pas
323     app.text_pas = uilabel(app.programa);
324     app.text_pas.HorizontalAlignment = 'center';
325     app.text_pas.VerticalAlignment = 'top';
326     app.text_pas.FontSize = 11;
327     app.text_pas.Position = [27 432 112 31];
328     app.text_pas.Text = {''; 'Pas (mm)'};

330     % Create input_pas
331     app.input_pas = uieditfield(app.programa, 'text');
332     app.input_pas.ValueChangedFcn = createCallbackFcn(app, ...
333         @input_pas_Callback, true);
334     app.input_pas.HorizontalAlignment = 'center';
335     app.input_pas.FontSize = 11;
336     app.input_pas.Position = [159 441 120 22];

338     % Create tria_material
339     app.tria_material = uidropdown(app.programa);
340     app.tria_material.Items = {};
341     app.tria_material.ValueChangedFcn = createCallbackFcn(app, ...
342         @tria_material_Callback, true);
343     app.tria_material.FontSize = 11;
344     app.tria_material.BackgroundColor = [1 1 1];
345     app.tria_material.Position = [159 384 120 22];
346     app.tria_material.Value = {};
```

```
345     app.text_material = uilabel(app.programa);
346     app.text_material.HorizontalAlignment = 'center';
347     app.text_material.VerticalAlignment = 'top';
348     app.text_material.FontSize = 11;
349     app.text_material.Position = [27 381 112 31];
350     app.text_material.Text = {''; 'Material'};

352     % Create slider_resolucio
353     app.slider_resolucio = uislider(app.programa);
354     app.slider_resolucio.Limits = [200 1000];
355     app.slider_resolucio.MajorTicks = [];
356     app.slider_resolucio.ValueChangedFcn = createCallbackFcn(app, ...
        @slider_resolucio_Callback, true);
357     app.slider_resolucio.MinorTicks = [];
358     app.slider_resolucio.FontSize = 11;
359     app.slider_resolucio.Position = [500 673 150 3];
360     app.slider_resolucio.Value = 250;

362     % Create text_opcionsfinal
363     app.text_opcionsfinal = uilabel(app.programa);
364     app.text_opcionsfinal.HorizontalAlignment = 'center';
365     app.text_opcionsfinal.VerticalAlignment = 'top';
366     app.text_opcionsfinal.FontSize = 11;
367     app.text_opcionsfinal.Position = [27 658 112 31];
368     app.text_opcionsfinal.Text = {''; 'Finals de l''espira'};

370     % Create Tancaments
371     app.Tancaments = uibuttongroup(app.programa);
372     app.Tancaments.SelectionChangedFcn = createCallbackFcn(app, ...
        @TancamentsSelectionChanged, true);
373     app.Tancaments.BorderType = 'none';
374     app.Tancaments.BackgroundColor = [1 1 1];
375     app.Tancaments.Position = [131 632 230 70];

377     % Create TancatsButton
378     app.TancatsButton = uiradiobutton(app.Tancaments);
379     app.TancatsButton.Text = 'Tancats';
380     app.TancatsButton.Position = [11 38 122 22];

382     % Create ObertsButton
383     app.ObertsButton = uiradiobutton(app.Tancaments);
384     app.ObertsButton.Text = 'Oberts';
385     app.ObertsButton.Position = [11 8 122 22];
386     app.ObertsButton.Value = true;

388     % Create RectificatsButton
389     app.RectificatsButton = uiradiobutton(app.Tancaments);
390     app.RectificatsButton.Text = 'Rectificats';
391     app.RectificatsButton.Position = [101 38 122 22];

393     % Create DoblementtancatsButton
394     app.DoblementtancatsButton = uiradiobutton(app.Tancaments);
395     app.DoblementtancatsButton.Text = 'Doblement tancats';
396     app.DoblementtancatsButton.Position = [101 8 122 22];

398     % Create boto_tria
399     app.boto_tria = uibutton(app.programa, 'state');
400     app.boto_tria.ValueChangedFcn = createCallbackFcn(app, ...
        @boto_triaValueChanged, true);
401     app.boto_tria.Text = 'Mostra la gràfica';
```

```
402     app.boto_tria.Position = [81 328 154 34];
404     % Show the figure after all components are created
405     app.programa.Visible = 'on';
406     end
407     end

409     % App creation and deletion
410     methods (Access = public)

412         % Construct app
413         function app = GUI_App
414             % Create UIFigure and components
415             createComponents(app)
416             % Register the app with App Designer
417             registerApp(app, app.programa)
418             % Execute the startup function
419             runStartupFcn(app, @GUI_OpeningFcn)
420             if nargin == 0
421                 clear app
422             end
423         end

425         % Code that executes before app deletion
426         function delete(app)
427             % Delete UIFigure when app is deleted
428             delete(app.programa)
429         end
430     end
431 end
```

Bibliografia

- [1] ACXESS SPRING, *Compression Spring End Types* [en línia]. Access Spring, 2019. [Consulta: 20 maig 2019]
Disponible a: <<https://www.accessspring.com/compression-spring-end-types.html>>
- [2] ACXESS SPRING, *Spring Creator* [en línia]. Access Spring, 2019. [Consulta: 28 abril 2019]
Disponible a: <<https://www.accessspring.com/espanol/helical-spring-calculations.html>>
- [3] APPLE DEVELOPER, *Xcode 10* [en línia]. Cupertino, Estats Units: Apple Inc., 2019
[Consulta: 23 maig 2019]
Disponible a: <<https://developer.apple.com/xcode/>>
- [4] CARLES RIBA ROMEVA, *Disseny i càlcul de molles* [en línia]. Barcelona: Edicions UPC, 1992. ISBN 9788498800852 [Consulta: 10 febrer 2019]
Disponible a: <<https://upcommons.upc.edu/bitstream/handle/2099.3/36327/9788498800852.pdf>>
- [5] COLLINS ENGLISH DICTIONARY, *Graphical user interface* [en línia]. HarperCollins Publishers, 2019. [Consulta: 18 maig 2019]
Disponible a: <<https://www.collinsdictionary.com/dictionary/english/graphical-user-interface>>
- [6] DIAMOND WIRE SPRING COMPANY, *Coiling compression springs* [en línia]. Diamond Wire Spring Company, 2019. [Consulta: 30 abril 2019].
Disponible a: <<https://www.diamondwire.com/about-springs/spring-manufacturing-process/coiling.html>>
- [7] INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR, *Design of Machine Elements I: Module 7 Design of Springs*. A: National Programme on Technology Enhanced Learning [en línia]. Government of India, Ministry of Human Resource Development, 2009. [Consulta: 2 maig 2019]
Disponible a: <<https://nptel.ac.in/courses/112105125/19>>
- [8] KING SPRINGS, *King Springs* [en línia]. Brisbane: King Springs, 2019.
Disponible a: <<https://kingsprings.com.au/>>
- [9] LEE SPRING, *Counting Coils*. A: Spring-I-Pedia [en línia]. Lee Spring, 2011. [Consulta: 20 maig 2019]
Disponible a: <<http://springipedia.com/compression-counting-coils.asp>>
- [10] LEE SPRING, *Spring Materials (Round Wire)*. A: Spring-I-Pedia [en línia]. Lee Spring, 2011. [Consulta: 3 maig 2019]
Disponible a: <<http://springipedia.com/material-spring-materials.asp>>

- [11] MATLAB®, *MATLAB® App Designer* [en línia]. The MathWorks, Inc, 2019. [Consulta: 27 maig 2019]
Disponible a: <<https://www.mathworks.com/products/matlab/app-designer.html>>
- [12] MATLAB®, *Create Standalone Application from MATLAB®* [en línia]. The MathWorks, Inc, 2019. [Consulta: 31 maig 2019]
Disponible a: <<https://es.mathworks.com/help/compiler/create-and-install-a-standalone-application-from-matlab-code.html>>
- [13] MATLAB®, *MATLAB® Runtime* [en línia]. The MathWorks, Inc, 2019. [Consulta: 31 maig 2019]
Disponible a:
<<https://es.mathworks.com/products/compiler/matlab-runtime.html>>
- [14] MATWEB, *MatWeb Material Property Data* [en línia]. MatWeb, 2019. [Consulta: 22 maig 2019]
Disponible a: <<http://www.matweb.com/index.aspx>>
- [15] MICROSOFT.COM OPERATIONS, *The GUI versus the Command Line: Which is better? (Part 1) A: Microsoft TechNet* [en línia]. Microsoft, 2007. [Consulta: 25 maig 2019]
Disponible a: <<https://blogs.technet.microsoft.com/mscom/2007/03/12/the-gui-versus-the-command-line-which-is-better-part-1/>>
- [16] PETRICH, MARTIN; SCHRODIN, BENEDIKT; KLETZIN, ULF. *Material Selection Method for Composite Springs* [en línia]. Ilmenau: ilmedia, 59th Ilmenau Scientific Colloquium, 2017. [Consulta: 2 maig 2019]
Disponible a: <https://www.db-thueringen.de/servlets/MCRFileNodeServlet/dbt_derivate_00039377/ilm1-2017iwk-140.pdf>
- [17] ROUSE, MARGARET, *integrated development environments (IDEs) A: TechTarget Search Software Quality* [en línia]. Techtarget, 2018. [Consulta: 25 maig 2019]
Disponible a: <<https://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>>
- [18] SARAROLS, MIQUEL; VECIANA, JOAQUIM M^a, *Vehicles: Sistemes de suspensió A: Atenea* [en línia]. ETSEIB, 2018. [Consulta: 29 abril 2019]
- [19] SOFTCATALÀ, *Recull de termes de Softcatalà* [en línia]. Softcatalà, 2010. [Consulta: 31 maig 2019]
Disponible a: <<https://www.softcatala.org/recull-termes-softcatala/>>
- [20] VICTORY SPRINGS, *Comprehensive Spring Design* [en línia]. Edmonton, Canadà: Victory Spring Ltd, 2015. [Consulta: 23 maig 2019]
Disponible a: <<http://www.victoryspring.ca/comprehensive-spring-design.pdf>>
- [21] VOIGT, KARL, *TagTrees: Improving Personal Information Management Using Associative Navigation* [en línia]. Tesi doctoral, Technische Universität Graz, 2012. [Consulta: 24 maig 2019]
Disponible a: <https://www.researchgate.net/publication/268505761_TagTrees_Improving_Personal_Information_Management_Using_Associative_Navigation>

- [22] WHITELEGG, *1.6 mm 5-axis Spring Coiling Machine* [en línia]. Herne Bay, Regne Unit: Whitelegg, 2019. [Consulta: 2 maig 2019]
Disponible a: <<http://www.whitelegg.com/products/wire-forming/spring-and-wireform-machinery/1-6-mm-5-axis-spring-coiling-machine>>
- [23] YAMADA, YOSHIRO; KUWABARA, TOSHIO, Appendix, Comparison between JIS and International Standards. A: *Materials for Springs* [en línia]. Nova York: Springer, 2007, p. 360-362. ISBN 9783540738114 [Consulta: 6 maig 2019]
Disponible a: <<https://books.google.es/books?id=7d0DrrDI1kEC&pg=PA360&lpg=PA360#v=onepage&q&f=false>>