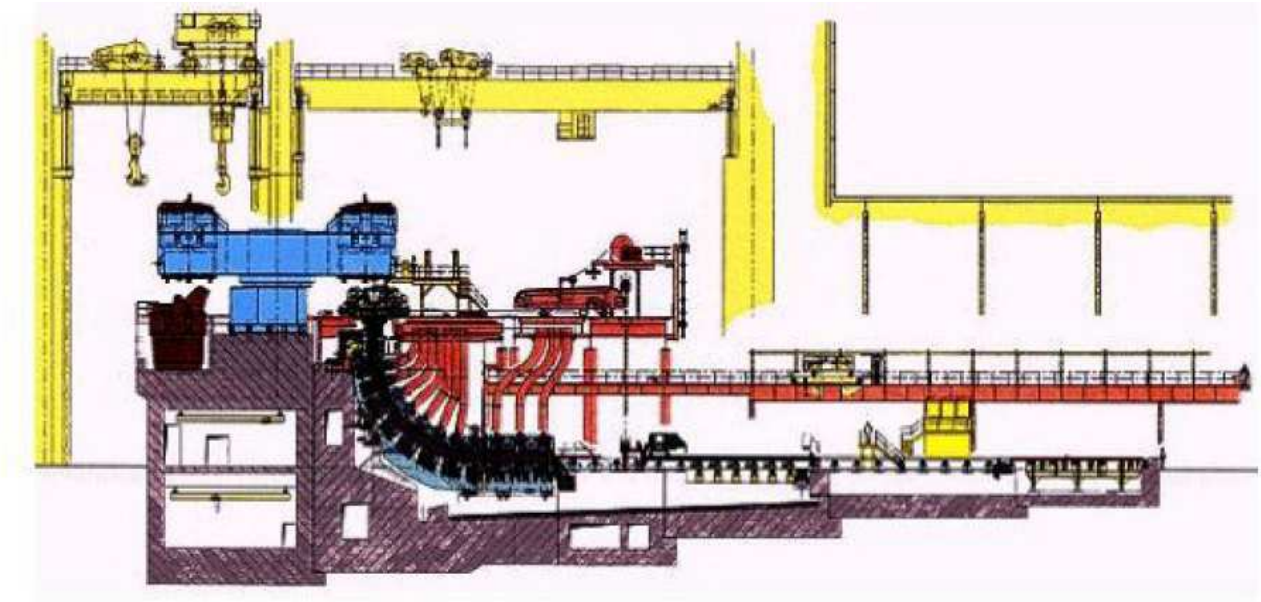
Università degli Studi di Udine

Colata continua

Interfaccia per la risoluzione di esercizi realizzata in Matlab.

--- Fonderia ---

Prof. Fabio Miani



Massimo Cignolini 89378

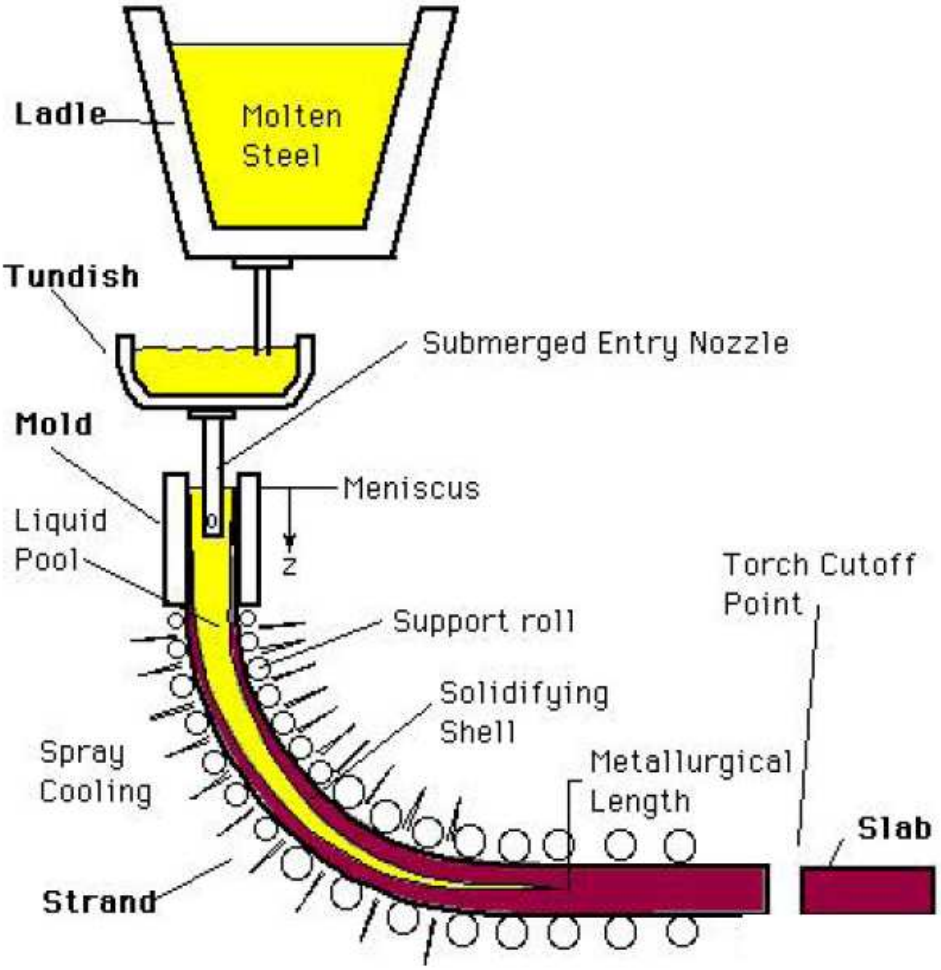
Filippo Tabaccanti 89391

Mirko Mazzei 89837

Obiettivi dell'elaborato:

Realizzare un'interfaccia in Matlab in grado di risolvere un esercizio di Colata continua proposto dalla SteelUniversity (www.steeluniversity.com).
Fornire un punto di partenza per un'eventuale sviluppo successivo di una simulazione grafica con animazioni in grado di mostrare all'utente la realizzazione step by step di una colata.
Fornire all'utente medio una rapida visione teorica sull'argomento in maniera tale che possa utilizzare in maniera più critica il software realizzato.

Introduzione: La colata continua



La colata continua è un processo di produzione industriale in cui l'acciaio allo stato liquido (presente nella siviera o ladle) viene fatto passare, sfruttando la forza di gravità, attraverso una forma di rame, raffreddata all'esterno da getti d'acqua, chiamata lingottiera (mold).

Grazie al raffreddamento forzato, il metallo si solidifica in superficie mentre rimane liquido nella parte interna della sua sezione. Tuttavia, questa parte esterna solidificata fornisce al pezzo una stabilità tale da poter farlo scendere attraverso un percorso curvo e da poter sorreggere il peso dell'acciaio liquido all'interno. In questo percorso il metallo continua ad essere raffreddato forzatamente attraverso degli spruzzi d'acqua diretti.

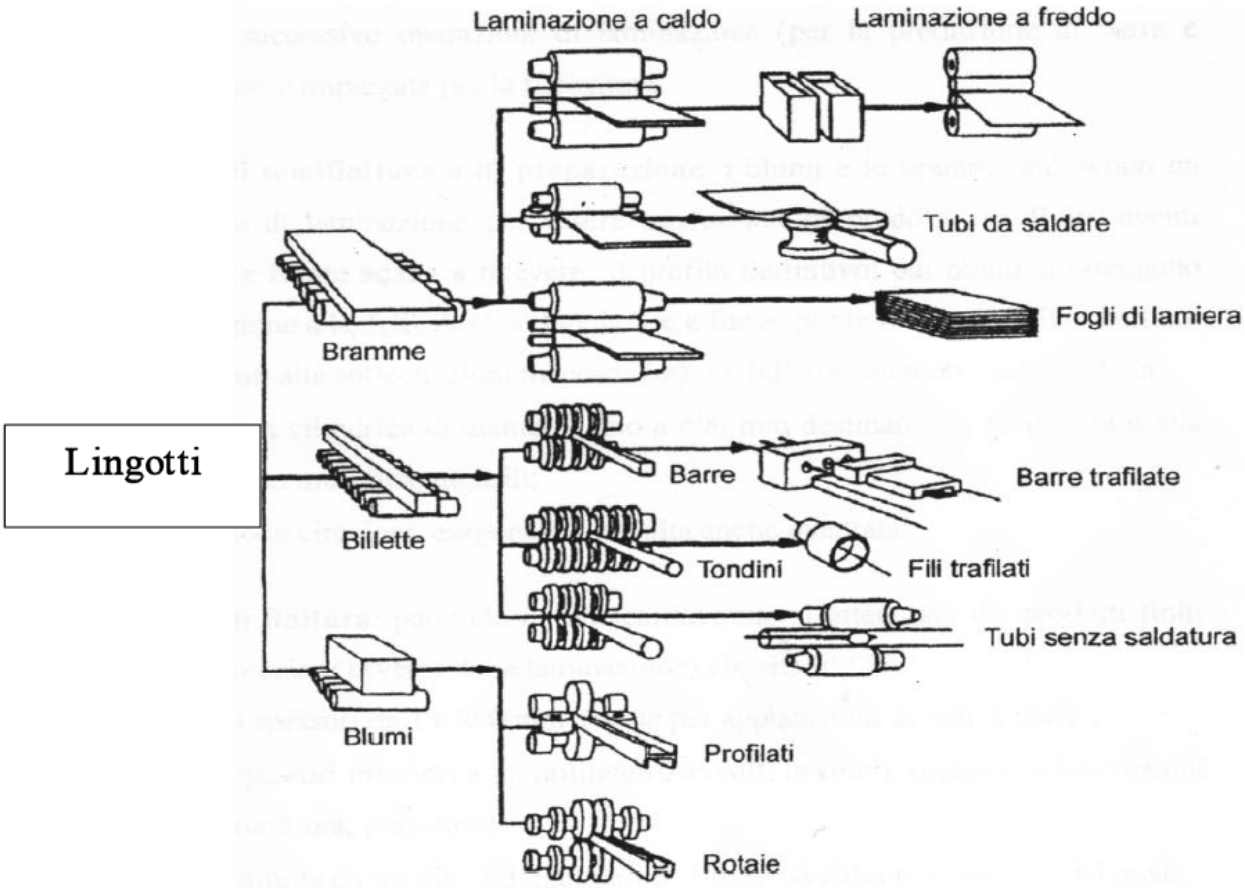
Durante tutto il processo di colata la lingottiera oscilla verticalmente per permettere la separazione dell'acciaio dal supporto in rame stesso. Questa operazione è ulteriormente facilitata dall'introduzione di una polvere (mold powder). La barra semilavorata di acciaio (strand) viene quindi trasportata attraverso un percorso con rulli che la porta ad una posizione orizzontale. Giunta in orizzontale, gran parte della sezione del pezzo colato è ormai solidificata.

Il pezzo colato va tagliato a misura mediante una fiamma ad ossigeno e lasciato raffreddare del tutto per effetto dell'aria.

Solitamente, una macchina a colata continua dispone di più linee di colata, ciascuna attrezzata di lingottiera, percorso di raffreddamento e taglio ad ossigeno. Le diverse linee vengono alimentate da un contenitore di distribuzione, detto paniera (tundish).

I semilavorati ottenuti dal processo di colata continua, a seconda della loro sezione, vengono chiamati blumi, billette o bramme (slab, blooms e billets), e sono destinati ad ulteriori processi di produzione industriale. (vedi figura seguente)

Questo processo nacque negli anni 50 allo scopo di incrementare la produzione industriale dell'acciaio.



Interfaccia di selezione:

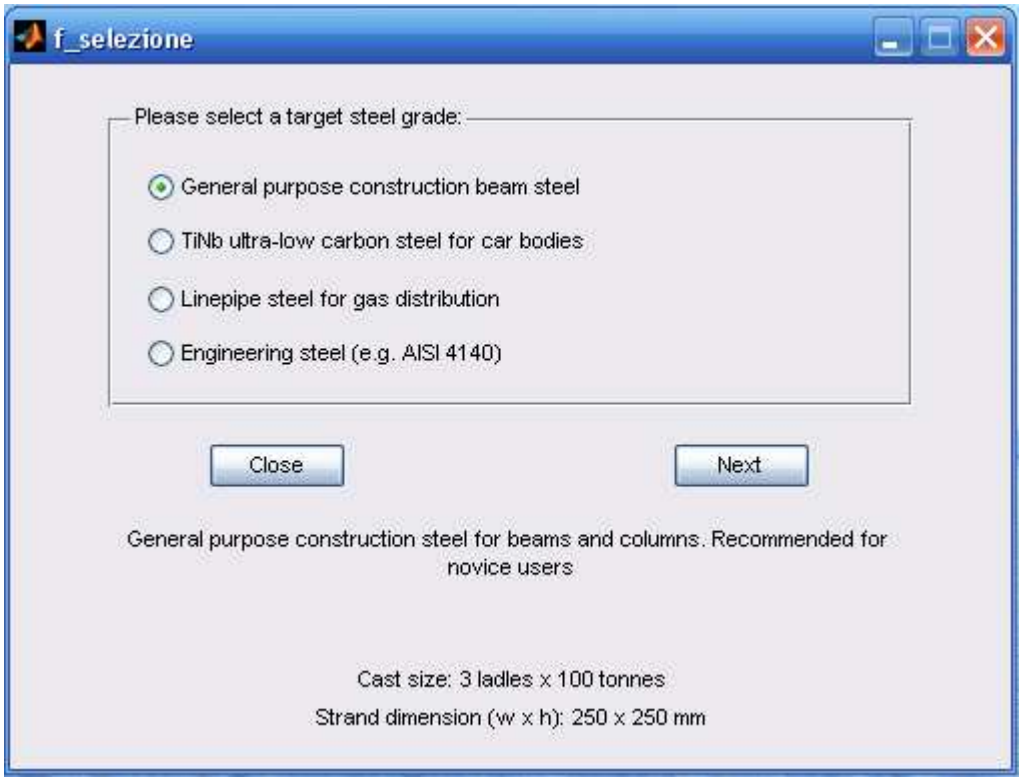
La prima interfaccia che si presenta all'utente permette di selezionare 4 diverse produzioni possibili di acciaio che possono essere realizzate attraverso 3 tipi di macchine. Ciascuna di queste macchine realizza un determinato tipo di semilavorato: slab, bloom e billet:

1) **construction steel grade:** ottenuto con una colata di tipo bloom con una sezione di 250 × 250 mm. Vi può essere un numero moderato di inclusioni senza che vi siano per questo problemi di qualità.

2) **TiNb ultra-low carbon steel (ULC steel):** E' un acciaio che viene impiegato nella costruzione di parti automobilistiche che deve avere una percentuale di carbonio inferiore allo 0.0035 %. Questo acciaio viene prodotto con una colata di tipo slab che permette di ottenere un semilavorato con sezione 1200 × 230 mm. E' importante in questo caso ridurre il più possibile il numero di eventuali inclusioni.

3) **linepipe steel (LPS steel):** semilavorato in acciaio impiegato nella distribuzione di gas. Necessita di alta resistenza alle fratture e bassissimo livello di impurità (S, P, H, O e N). Anche in questo caso, come nel precedente, viene impiegata una colata di tipo slab che permette di ottenere un semilavorato con sezione 1200 × 230 mm.

4) **engineering steel:** è un tipo di acciaio che viene prodotto, ad alta velocità, in billets di sezione 130 × 130 mm.



Sorgente “f_selezione”:

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

% Copyright 2002-2003 The MathWorks, Inc.

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

% Last Modified by GUIDE v2.5 30-Sep-2009 22:05:37

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

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

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

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

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_selezione wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global tipo;
tipo = 1;
testo1 = 'General purpose construction steel for beams and columns. Recommended
for novice users';
testo2 = 'Cast size: 3 ladles x 100 tonnes';
testo3 = 'Strand dimension (w x h): 250 x 250 mm';
set(handles.text1,'String',testo1);
set(handles.text2,'String',testo2);
set(handles.text3,'String',testo3);
% --- Outputs from this function are returned to the command line.
function varargout = f_selezione_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --------------------------------------------------------------------
function uipanel1_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to uipanel1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global tipo;
global testo;

if hObject == handles.radiobutton1
    tipo = 1;
    testo1 = 'General purpose construction steel for beams and columns.
Recommended for novice users';
    testo2 = 'Cast size: 3 ladles x 100 tonnes';
    testo3 = 'Strand dimension (w x h): 250 x 250 mm';
elseif hObject == handles.radiobutton2
    tipo = 2;
    testo1 = 'A TiNb ultra-low carbon for car bodies';
    testo2 = 'Cast size: 3 ladles x 250 tonnes';
    testo3 = 'Strand dimension (w x h): 230 x 1200 mm';
elseif hObject == handles.radiobutton3
    tipo = 3;
    testo1 = 'The linepipe steel for oil and gas distribution is a very
demanding grade as the combination of high strength and high fracture toughness
demands extremely low levels of impurities (S, P, H, O and N) and inclusions.
Recommended for more experienced users.';
    testo2 = 'Cast size: 3 ladles x 250 tonnes';
    testo3 = 'Strand dimension (w x h): 230 x 1200 mm';
elseif hObject == handles.radiobutton4
    tipo = 4;
    testo1 = 'A heat-treatable medium-carbon CrMo engineering steel provides an
example of a more highly alloyed grade. It contains significant Cr and Mo
additions, and also requires a low hydrogen content.';
    testo2 = 'Cast size: 3 ladles x 100 tonnes';
    testo3 = 'Strand dimension (w x h): 130 x 130 mm';
else tipo = 1;
```

```
     testo1 = 'General purpose construction steel for beams and columns.
Recommended for novice users';
     testo2 = 'Cast size: 3 ladles x 100 tonnes';
     testo3 = 'Strand dimension (w x h): 250 x 250 mm';
end

set(handles.text1,'String',testo1);
set(handles.text2,'String',testo2);
set(handles.text3,'String',testo3);

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global tipo;

if tipo == 1
    f_dati1;
elseif tipo == 2
    f_dati2;
elseif tipo == 3
    f_dati3;
elseif tipo == 4
    f_dati4;
end
```

Interfaccia dati:

Durante la colata di tipo slab si utilizza un processo chiamato soft reduction. La soft reduction deve essere realizzata nella zona in cui il metallo è ancora liquido all'interno della strand, ovvero nella metallurgical length, e questo richiede una scelta oculata della velocità di colata e di quella di raffreddamento.
La soft reduction avviene tramite 5 rulli da 400 mm di diametro posti in una zona lunga approssimativamente 2 metri.
La zona di soft reduction può quindi essere individuata nella lunghezza della strand in un intervallo da 27 a 31 metri nel caso del Linepipe steel (LPS) e fra i 21 e 24 metri nel caso dell'Ultra-low carbon steel (ULC).

Casting Speed and Secondary Cooling Rate:

E' di fondamentale importanza la scelta della casting speed e del cooling rate infatti questi dati influenzano molti parametri di colata e ne determinano quindi la qualità. Ad esempio, un parametro che viene direttamente determinato da questa scelta è la metallurgical length, ovvero, come spiegato in precedenza la distanza alla quale la strand diventa totalmente solida.
La metallurgical length è determinata da una complessa combinazione di composizione, casting speed, cooling rate e dimensioni della strand. Essa viene indicata nell'interfaccia dati. I valori che vengono indicati a seconda dei casi sono stati ottenuti consultando delle tabelle specifiche:

Cooling Rate / kg water per kg steel	Casting Speed / m min^{-1}			
	1.2	1.4	1.6	1.8
0.3	22.48	26.55	30.43	34.65
0.4	21.78	25.57	29.10	33.12
0.5	20.96	24.43	27.55	31.26
0.6	20.04	23.17	25.57	29.22

Table 6-3 Metallurgical lengths for ultra-low carbon steel cast in the slab caster, 1200 × 230 mm.

Cooling Rate / kg water per kg steel	Casting Speed / m min^{-1}					
	1.0	1.2	1.4	1.6	1.8	2.0
0.4	19.03	23.06	27.23	31.55	36.06	40.73
0.5	18.30	22.16	26.16	30.30	34.62	39.10
0.6	17.67	21.38	25.23	29.22	33.36	37.70
0.7	17.11	20.70	24.43	28.30	32.28	36.47
0.8	16.63	20.10	23.70	27.46	31.35	35.40

Table 6-4 Metallurgical lengths for linepipe steel cast in the slab caster, 1200 × 230 mm.

Cooling Rate / kg water per kg steel	Casting Speed / m min^{-1}					
	1.0	1.2	1.4	1.6	1.8	2.0
0.4	20.17	24.50	28.98	33.65	38.55	43.56
0.5	19.40	23.56	27.86	32.35	37.02	41.87
0.6	18.75	22.74	26.88	31.20	35.70	40.37
0.7	18.17	22.02	26.04	30.21	34.56	39.10
0.8	17.65	21.40	25.30	29.33	33.57	37.97

Mold Oscillation Settings:

Il mold viene fatto oscillare per permettere il distacco della strand. Questa operazione viene facilitata dell'uso di mold powder fra la strand ed il mold.

Stroke, *S* [mm]: di solito è compreso fra 3 e 10 mm. Il suo aumento provoca una crescita proporzionale del negative strip time, dell'oscillation mark depth e del consumo di mold powder.

Frequency, *f* [min-1]: i mold oscillator producono delle oscillazioni a frequenza compresa fra i 100 e i 250 cicli al minuto. Incrementando la frequenza diminuiscono gli oscillation mark depth, il negative strip time ed il consumo di mold powder.

Mold acceleration, *ma* [m*s-2]: accelerazione massima del mold calcolata ricavando la formula dal moto armonico avendo come dati S ed f:

$$ma = (S / 2) / 1000 * (2 * \pi * (f / 60))^2 \qquad [m*s-2]$$

Negative strip time, *t*N [s]: è il periodo di tempo in cui la velocità del mold è maggiore della casting speed.

$$t_N = \frac{60}{\pi f} \arccos \frac{1000\, v_{cast}}{\pi f S} \quad [s]$$

f = frequency, min^{-1}
S = stroke, mm
v_{cast} = casting speed, m min^{-1}

Oscillation mark depth, *d* [mm]: nell'oscillazione si determina una riduzione della qualità della superficie dell'acciaio, chiamata oscillation mark depth, sottoforma di segni che si presentano periodicamente e che spesso sono fonte di crack trasversali. Il negative strip time è la fonte principale della formazione di questi difetti.
Allo scopo di minimizzare la profondità di tali difetti è necessario prestare particolare attenzione al setting dei parametri di oscillazione. In particolare il negative strip time dovrebbe essere vicino a 0.11 s e associato ad uno stroke adatto.
Il massimo valore di oscillation mark depth accettabile per acciaio ULC è 0.25 mm mentre nel caso degli altri tre tipi di acciaio è 0.60 mm.

$$d = 0.065 \cdot 1.145^{S} \cdot (200 \cdot 0.9^{S})^{t_N} \quad [mm]$$

t_N = negative strip time, s

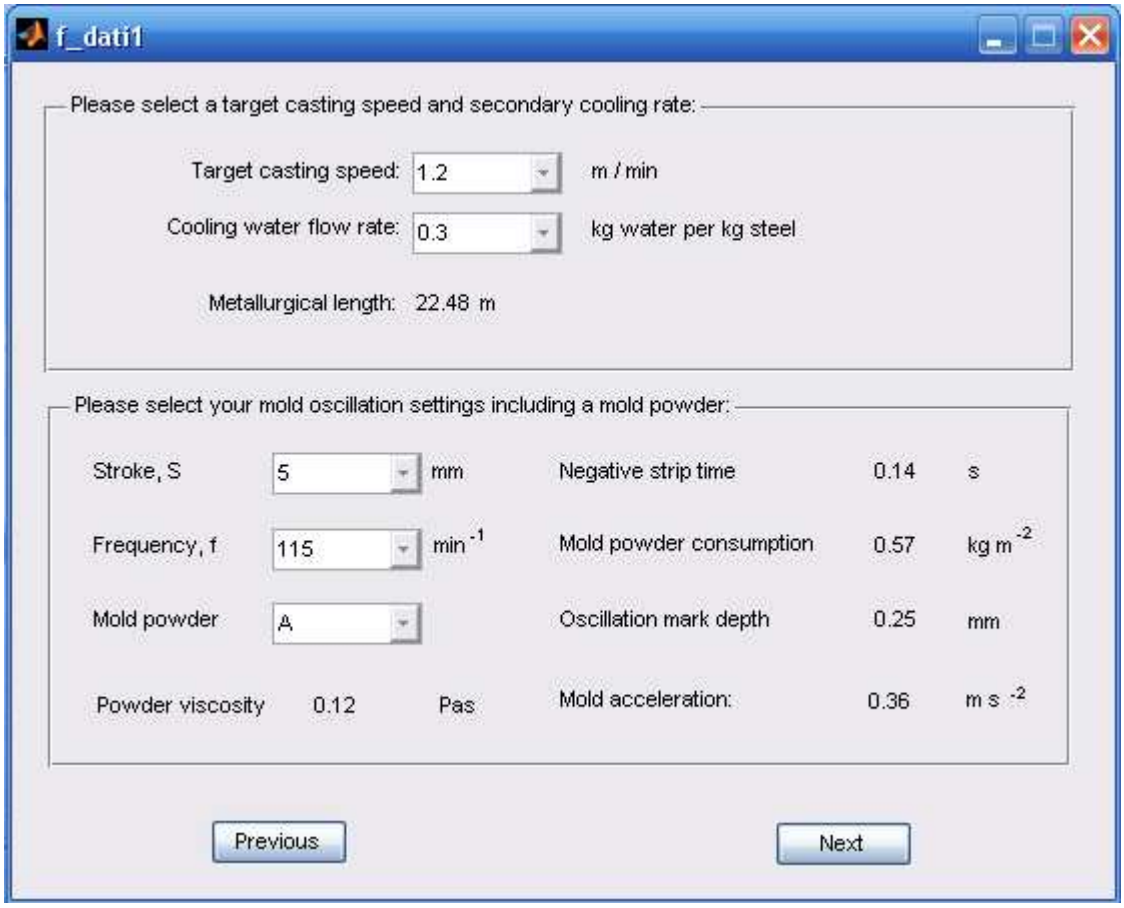
Mold powder: è un materiale sintetico che viene continuamente versato sulla superficie dell'acciaio fuso durante la colata. Tale polvere si scioglie e scorre fra la parete del mold e la superficie della strand. Scegliendo il corretto tipo di polvere si contribuisce a determinare un buon grado di qualità della superficie dell'acciaio. Esso influenza anche l'oscillation mark depth e la mold powder consumption.
Le funzioni principali della mold powder sono:

1) *lubrificare la zona fra mold e strand*
2) *migliorare il trasferimento di calore fra mold e strand*
3) *fornire isolamento termico alla superficie della molten pool*
4) *protegge l'acciaio liquido dalla re-ossidazione*
5) *assorbire eventuali inclusioni che risalgono sulla superficie del metallo*

Mold powder consumption dipende dal tipo di polvere ma anche dal setting dell'oscillazione e dalla casting speed. Tale consumo viene misurato in massa per unità d'area della superficie della strand.
Il movimento del mold determina, quindi, anche la mold powder consumption.
Una mold powder consumption troppo bassa potrebbe causare un fenomeno di sticking tra la strand ed il mold che potrebbe portare ad una rottura della strand. Per evitare questo la mold powder consumption deve essere superiore a 0.30 kg m-2 (tranne nel caso dell'engineering steel dove anche 0.15 kg m-2 è un valore adeguato).
Una delle caratteristiche più importanti della mold powder è la **break temperature**. Essa è definita come la soglia di temperatura alla quale la viscosità della polvere aumenta notevolmente, ovvero il punto in cui la lubrificazione raggiunge il break-down.

$$Q = \frac{1.7 \times t_N}{\sqrt{\eta} \times v_c} \ [\text{kg m}^{-2}]$$

Q = mold powder consumption per unit area, kg m^{-2}
t_N = negative strip time, s
η = mold powder viscosity, Pa s
v_c = casting speed, m min^{-1}



Sorgente “f_dati1.m”:

```
function varargout = f_dati1(varargin)
% F_DATI1 M-file for f_dati1.fig
%      F_DATI1, by itself, creates a new F_DATI1 or raises the existing
%      singleton*.
%
%      H = F_DATI1 returns the handle to a new F_DATI1 or the handle to
%      the existing singleton*.
%
%      F_DATI1('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in F_DATI1.M with the given input arguments.
%
%      F_DATI1('Property','Value',...) creates a new F_DATI1 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before f_dati1_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to f_dati1_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help f_dati1

% Last Modified by GUIDE v2.5 30-Sep-2009 17:47:27
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @f_dati1_OpeningFcn, ...
                   'gui_OutputFcn',  @f_dati1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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


% --- Executes just before f_dati1 is made visible.
function f_dati1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to f_dati1 (see VARARGIN)

% Choose default command line output for f_dati1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_dati1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global target;
global water;
global S;
global viscos;
global f;
global mp;
global ma;

target = 1.2;
water = 0.3;
mp = 'A';
S = 3;
viscos = 0.12;
f = 100;

metri = lunghezza;

set(handles.text1,'String',metri);
set(handles.text2,'String',viscos);
[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = f_dati1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
global target;
data = get(handles.popupmenu1,'Value');
switch data
    case 1
        target = 1.2;
    case 2
        target = 1.4;
    case 3
        target = 1.6;
    case 4
        target = 1.8;
    otherwise
        target = 1.2;
end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2
```

```
global water;
data = get(handles.popupmenu2,'Value');
switch data
    case 1
        water = 0.3;
    case 2
        water = 0.4;
    case 3
        water = 0.5;
    case 4
        water = 0.6;
    otherwise
        water = 0.3;
end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

%-----------
function [metri] = lunghezza()
global target;
global water;

switch target
    case 1.2
        if water == 0.3
            metri = '22.48';
        elseif water == 0.4
            metri = '21.78';
        elseif water == 0.5
            metri = '20.96';
        elseif water == 0.6
            metri = '20.04';
        else metri = '-----';
        end
    case 1.4
        if water == 0.3
            metri = '26.55';
        elseif water == 0.4
            metri = '25.57';
        elseif water == 0.5
            metri = '24.43';
        elseif water == 0.6
            metri = '23.17';
        else metri = '-----';
        end
    case 1.6
        if water == 0.3
```

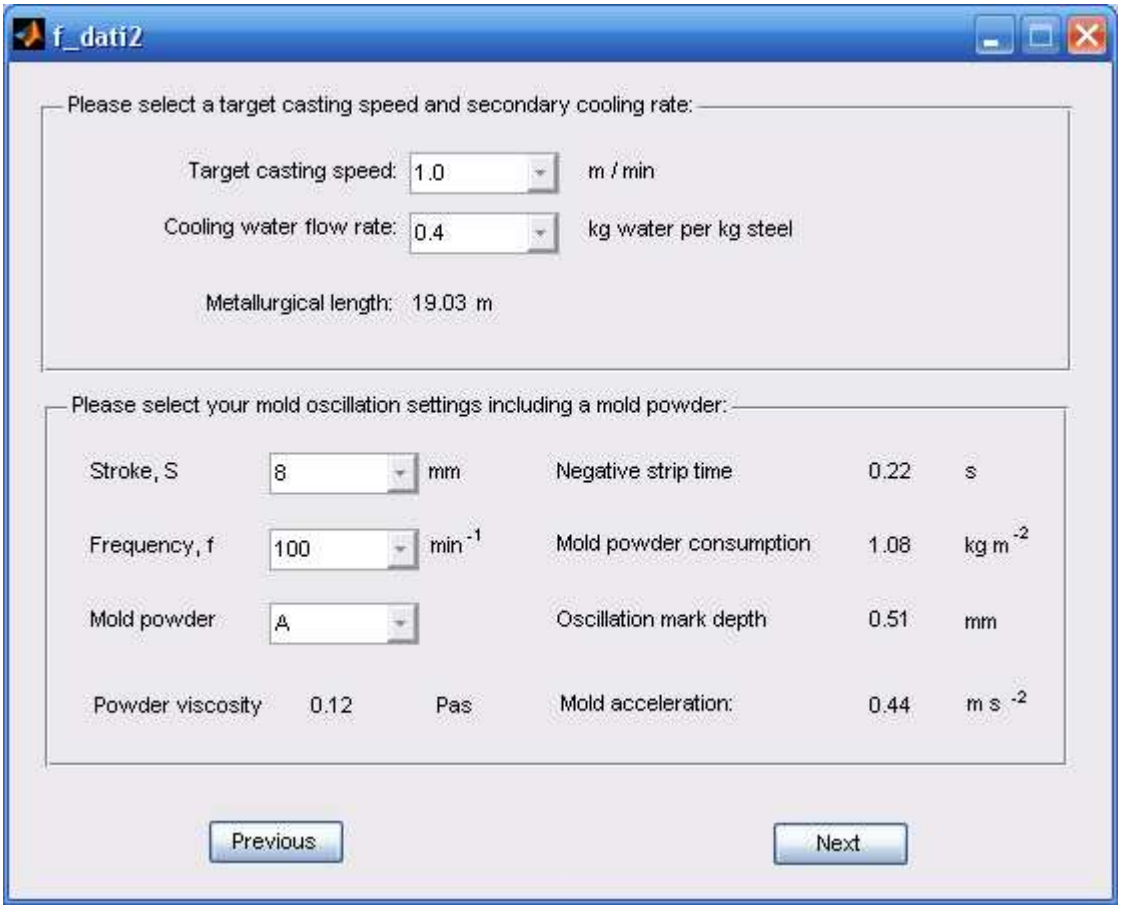
```
            metri = '30.43';
        elseif water == 0.4
            metri = '29.10';
        elseif water == 0.5
            metri = '27.55';
        elseif water == 0.6
            metri = '25.57';
        else metri = '-----';
        end
    case 1.8
        if water == 0.3
            metri = '34.65';
        elseif water == 0.4
            metri = '33.12';
        elseif water == 0.5
            metri = '31.26';
        elseif water == 0.6
            metri = '29.22';
        else metri = '-----';
        end
    otherwise
        metri = '-----';
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f_temperature;

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3
global S;
global ma;
stroke = get(handles.popupmenu3,'Value');
switch stroke
    case 1
        S = 3;
    case 2
        S = 4;
    case 3
        S = 5;
    case 4
        S = 6;
    case 5
        S = 7;
    case 6
```

```
        S = 8;
    case 7
        S = 9;
    case 8
        S = 10;
    case 9
        S = 11;
    case 10
        S = 12;
    case 11
        S = 13;
    case 12
        S = 14;
    case 13
        S = 15;
    otherwise
        S = 3;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu4
global f;
global ma;
freq = get(handles.popupmenu4,'Value');
switch freq
    case 1
        f = 100;
    case 2
        f = 105;
    case 3
        f = 110;
    case 4
        f = 115;
```

```
    case 5
        f = 120;
    case 6
        f = 125;
    case 7
        f = 130;
    case 8
        f = 135;
    case 9
        f = 140;
    case 10
        f = 145;
    case 11
        f = 150;
    case 12
        f = 155;
    case 13
        f = 160;
    case 14
        f = 165;
    case 15
        f = 170;
    case 16
        f = 175;
    case 17
        f = 180;
    case 18
        f = 185;
    case 19
        f = 190;
    case 20
        f = 195;
    case 21
        f = 200;
    case 22
        f = 205;
    case 23
        f = 210;
    case 24
        f = 215;
    case 25
        f = 220;
    case 26
        f = 225;
    case 27
        f = 230;
    case 28
        f = 235;
    case 29
        f = 240;
    case 30
        f = 245;
    case 31
        f = 250;
    otherwise
        f = 100;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
```

```
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu5
global viscos;
global vi;
global mp;
global ma;

vi = get(handles.popupmenu5,'Value');
switch vi
    case 1
        mp = 'A';
        viscos = 0.12;
    case 2
        mp = 'B';
        viscos = 0.21;
    case 3
        mp = 'C';
        viscos = 0.19;
    case 4
        mp = 'D';
        viscos = 0.10;
    case 5
        mp = 'E';
        viscos = 0.003;
    otherwise
        mp = 'A';
        viscos = 0.12;
end

set(handles.text2,'String',viscos);

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% Funzione che calcola i parametri fondamentali
function [tn, d, Q, ma] = calcola()
global S;
global f;
global target;
global tn;
global d;
global viscos;
global Q;
global ma;

tmp = (60 / (pi * f)) * acos ((1000 * target) / (pi * f * S));
tmp = tmp * 100;
tmp = round(tmp);
tn = tmp / 100;

tmp2 = 0.065 * (1.145^S) * (200 * (0.9^S))^tn;
tmp2 = tmp2 * 100;
tmp2 = round(tmp2);
d = tmp2 / 100;

tmp3 = (1.7 * tn) / (sqrt(viscos) * target);
tmp3 = tmp3 * 100;
tmp3 = round (tmp3);
Q = tmp3 / 100;

tmp4 = (S / 2000) * ( 2 * pi * f / 60)^2;
tmp4 = tmp4 * 100;
tmp4 = round(tmp4);
ma = tmp4 / 100;
```



Sorgente “f_dati2.m”:

```
function varargout = f_dati2(varargin)
% F_DATI2 M-file for f_dati2.fig
%      F_DATI2, by itself, creates a new F_DATI2 or raises the existing
%      singleton*.
%
%      H = F_DATI2 returns the handle to a new F_DATI2 or the handle to
%      the existing singleton*.
%
%      F_DATI2('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in F_DATI2.M with the given input arguments.
%
%      F_DATI2('Property','Value',...) creates a new F_DATI2 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before f_dati2_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to f_dati2_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help f_dati2

% Last Modified by GUIDE v2.5 30-Sep-2009 17:47:27
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @f_dati2_OpeningFcn, ...
                   'gui_OutputFcn',  @f_dati2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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


% --- Executes just before f_dati2 is made visible.
function f_dati2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to f_dati2 (see VARARGIN)

% Choose default command line output for f_dati2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_dati2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global target;
global water;
global S;
global viscos;
global f;
global mp;
global ma;

target = 1.0;
water = 0.4;
mp = 'A';
S = 3;
viscos = 0.12;
f = 100;

metri = lunghezza;

set(handles.text1,'String',metri);
set(handles.text2,'String',viscos);
[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = f_dati2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
global target;
data = get(handles.popupmenu1,'Value');
switch data
    case 1
        target = 1.0;
    case 2
        target = 1.2;
    case 3
        target = 1.4;
    case 4
        target = 1.6;
    case 5
        target = 1.8;
    case 6
        target = 2.0;
    otherwise
        target = 1.0;

end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2
global water;
data = get(handles.popupmenu2,'Value');
switch data
    case 1
        water = 0.4;
    case 2
        water = 0.5;
    case 3
        water = 0.6;
    case 4
        water = 0.7;
    case 5
        water = 0.8;
    otherwise
        water = 0.4;
end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

%-----------
function [metri] = lunghezza()
global target;
global water;

switch target
    case 1.0
        if water == 0.4
            metri = '19.03';
        elseif water == 0.5
            metri = '18.30';
        elseif water == 0.6
            metri = '17.76';
        elseif water == 0.7
            metri = '17.11';
        elseif water == 0.8
            metri = '16.63';
        else metri = '-----';
        end
    case 1.2
        if water == 0.4
            metri = '23.06';
```

```
    elseif water == 0.5
        metri = '22.16';
    elseif water == 0.6
        metri = '21.38';
    elseif water == 0.7
        metri = '20.70';
    elseif water == 0.8
        metri = '20.10';
    else metri = '-----';
    end
case 1.4
    if water == 0.4
        metri = '27.23';
    elseif water == 0.5
        metri = '26.16';
    elseif water == 0.6
        metri = '25.23';
    elseif water == 0.7
        metri = '24.43';
    elseif water == 0.8
        metri = '23.70';
    else metri = '-----';
    end
case 1.6
    if water == 0.4
        metri = '31.55';
    elseif water == 0.5
        metri = '30.30';
    elseif water == 0.6
        metri = '29.22';
    elseif water == 0.7
        metri = '28.30';
    elseif water == 0.8
        metri = '27.46';
    else metri = '-----';
    end
case 1.8
    if water == 0.4
        metri = '36.06';
    elseif water == 0.5
        metri = '34.62';
    elseif water == 0.6
        metri = '33.36';
    elseif water == 0.7
        metri = '32.28';
    elseif water == 0.8
        metri = '31.35';
    else metri = '-----';
    end
case 2.0
    if water == 0.4
        metri = '40.73';
    elseif water == 0.5
        metri = '39.10';
    elseif water == 0.6
        metri = '37.70';
    elseif water == 0.7
        metri = '36.47';
    elseif water == 0.8
        metri = '35.40';
    else metri = '-----';
    end
otherwise
```

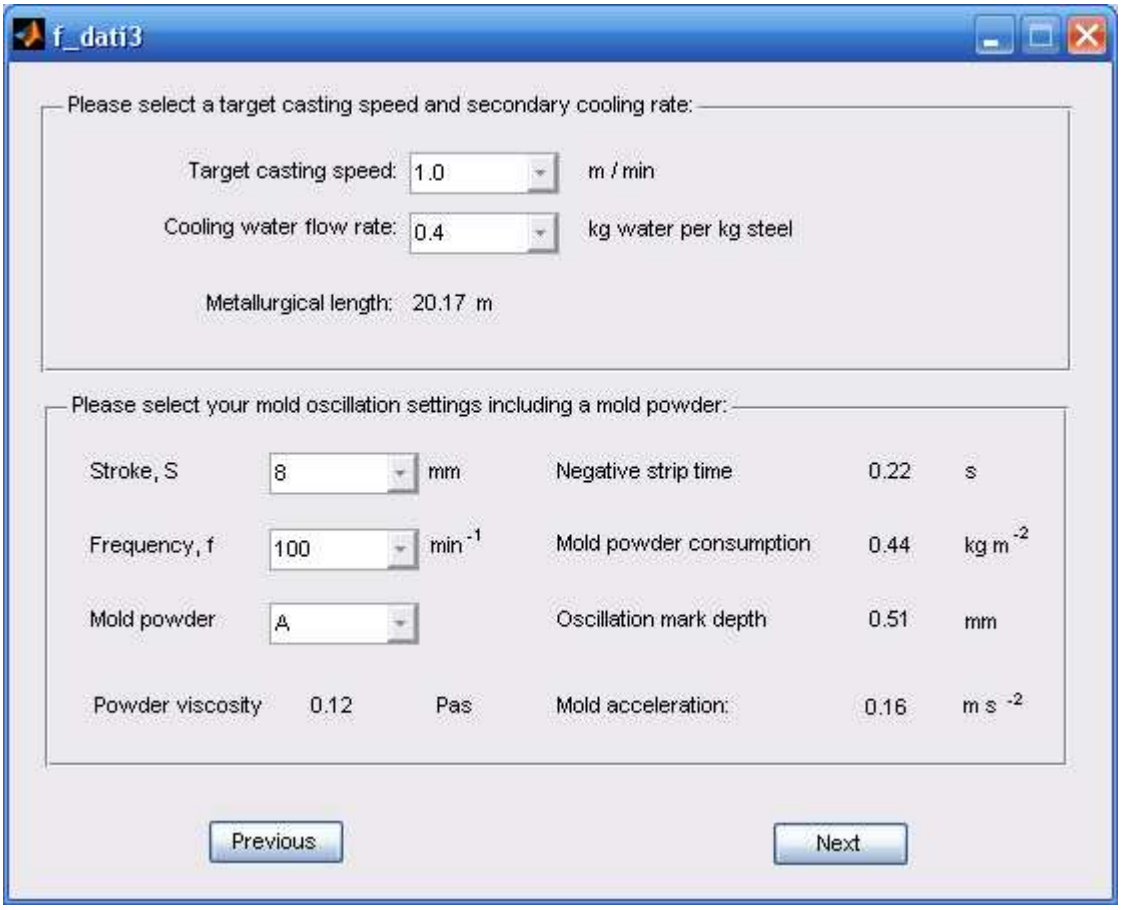
```
        metri = '-----';
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f_temperature;

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3
global S;
stroke = get(handles.popupmenu3,'Value');
switch stroke
    case 1
        S = 3;
    case 2
        S = 4;
    case 3
        S = 5;
    case 4
        S = 6;
    case 5
        S = 7;
    case 6
        S = 8;
    case 7
        S = 9;
    case 8
        S = 10;
    case 9
        S = 11;
    case 10
        S = 12;
    case 11
        S = 13;
    case 12
        S = 14;
    case 13
        S = 15;
    otherwise
        S = 3;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
```

```
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu4
global f;
global ma;

freq = get(handles.popupmenu4,'Value');
switch freq
    case 1
        f = 100;
    case 2
        f = 105;
    case 3
        f = 110;
    case 4
        f = 115;
    case 5
        f = 120;
    case 6
        f = 125;
    case 7
        f = 130;
    case 8
        f = 135;
    case 9
        f = 140;
    case 10
        f = 145;
    case 11
        f = 150;
    case 12
        f = 155;
    case 13
        f = 160;
    case 14
        f = 165;
    case 15
```

```
        f = 170;
    case 16
        f = 175;
    case 17
        f = 180;
    case 18
        f = 185;
    case 19
        f = 190;
    case 20
        f = 195;
    case 21
        f = 200;
    case 22
        f = 205;
    case 23
        f = 210;
    case 24
        f = 215;
    case 25
        f = 220;
    case 26
        f = 225;
    case 27
        f = 230;
    case 28
        f = 235;
    case 29
        f = 240;
    case 30
        f = 245;
    case 31
        f = 250;
    otherwise
        f = 100;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu5
global viscos;
global vi;
global mp;

vi = get(handles.popupmenu5,'Value');
switch vi
    case 1
        mp = 'A';
        viscos = 0.12;
    case 2
        mp = 'B';
        viscos = 0.21;
    case 3
        mp = 'C';
        viscos = 0.19;
    case 4
        mp = 'D';
        viscos = 0.10;
    case 5
        mp = 'E';
        viscos = 0.003;
    otherwise
        mp = 'A';
        viscos = 0.12;
end

set(handles.text2,'String',viscos);

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% Funzione che calcola i parametri fondamentali
function [tn, d, Q, ma] = calcola()
global S;
global f;
global target;
global tn;
global d;
global viscos;
```

```
global Q;
global ma;

tmp = (60 / (pi * f)) * acos ((1000 * target) / (pi * f * S));
tmp = tmp * 100;
tmp = round(tmp);
tn = tmp / 100;

tmp2 = 0.065 * (1.145^S) * (200 * (0.9^S))^tn;
tmp2 = tmp2 * 100;
tmp2 = round(tmp2);
d = tmp2 / 100;


tmp3 = (1.7 * tn) / (sqrt(viscos) * target);
tmp3 = tmp3 * 100;
tmp3 = round (tmp3);
Q = tmp3 / 100;

tmp4 = (S / 2000) * ( 2 * pi * f / 60)^2;
tmp4 = tmp4 * 100;
tmp4 = round(tmp4);
ma = tmp4 / 100;
```



Sorgente “f_dati3.m”:

```
function varargout = f_dati3(varargin)
% F_DATI3 M-file for f_dati3.fig
%      F_DATI3, by itself, creates a new F_DATI3 or raises the existing
%      singleton*.
%
%      H = F_DATI3 returns the handle to a new F_DATI3 or the handle to
%      the existing singleton*.
%
%      F_DATI3('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in F_DATI3.M with the given input arguments.
%
%      F_DATI3('Property','Value',...) creates a new F_DATI3 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before f_dati3_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to f_dati3_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help f_dati3

% Last Modified by GUIDE v2.5 30-Sep-2009 17:47:27
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @f_dati3_OpeningFcn, ...
                   'gui_OutputFcn',  @f_dati3_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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


% --- Executes just before f_dati3 is made visible.
function f_dati3_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to f_dati3 (see VARARGIN)

% Choose default command line output for f_dati3
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_dati3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global target;
global water;
global S;
global viscos;
global f;
global mp;
global ma;

target = 1.0;
water = 0.4;
mp = 'A';
S = 3;
viscos = 0.12;
f = 100;

metri = lunghezza;

set(handles.text1,'String',metri);
set(handles.text2,'String',viscos);
[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = f_dati3_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
global target;
data = get(handles.popupmenu1,'Value');
switch data
    case 1
        target = 1.0;
    case 2
        target = 1.2;
    case 3
        target = 1.4;
    case 4
        target = 1.6;
    case 5
        target = 1.8;
    case 6
        target = 2.0;
    otherwise
        target = 1.0;

end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2
global water;
data = get(handles.popupmenu2,'Value');
switch data
    case 1
        water = 0.4;
    case 2
        water = 0.5;
    case 3
        water = 0.6;
    case 4
        water = 0.7;
    case 5
        water = 0.8;
    otherwise
        water = 0.4;
end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

%-----------
function [metri] = lunghezza()
global target;
global water;

switch target
    case 1.0
        if water == 0.4
            metri = '20.17';
        elseif water == 0.5
            metri = '19.40';
        elseif water == 0.6
            metri = '18.75';
        elseif water == 0.7
            metri = '18.17';
        elseif water == 0.8
            metri = '17.65';
        else metri = '-----';
        end
    case 1.2
        if water == 0.4
            metri = '24.50';
```

```
    elseif water == 0.5
        metri = '23.56';
    elseif water == 0.6
        metri = '22.74';
    elseif water == 0.7
        metri = '22.02';
    elseif water == 0.8
        metri = '21.40';
    else metri = '-----';
    end
case 1.4
    if water == 0.4
        metri = '28.98';
    elseif water == 0.5
        metri = '27.86';
    elseif water == 0.6
        metri = '26.88';
    elseif water == 0.7
        metri = '26.04';
    elseif water == 0.8
        metri = '25.30';
    else metri = '-----';
    end
case 1.6
    if water == 0.4
        metri = '33.65';
    elseif water == 0.5
        metri = '32.35';
    elseif water == 0.6
        metri = '31.20';
    elseif water == 0.7
        metri = '30.21';
    elseif water == 0.8
        metri = '29.33';
    else metri = '-----';
    end
case 1.8
    if water == 0.4
        metri = '38.55';
    elseif water == 0.5
        metri = '37.02';
    elseif water == 0.6
        metri = '35.70';
    elseif water == 0.7
        metri = '34.56';
    elseif water == 0.8
        metri = '33.57';
    else metri = '-----';
    end
case 2.0
    if water == 0.4
        metri = '43.56';
    elseif water == 0.5
        metri = '41.87';
    elseif water == 0.6
        metri = '40.37';
    elseif water == 0.7
        metri = '39.10';
    elseif water == 0.8
        metri = '37.97';
    else metri = '-----';
    end
otherwise
```

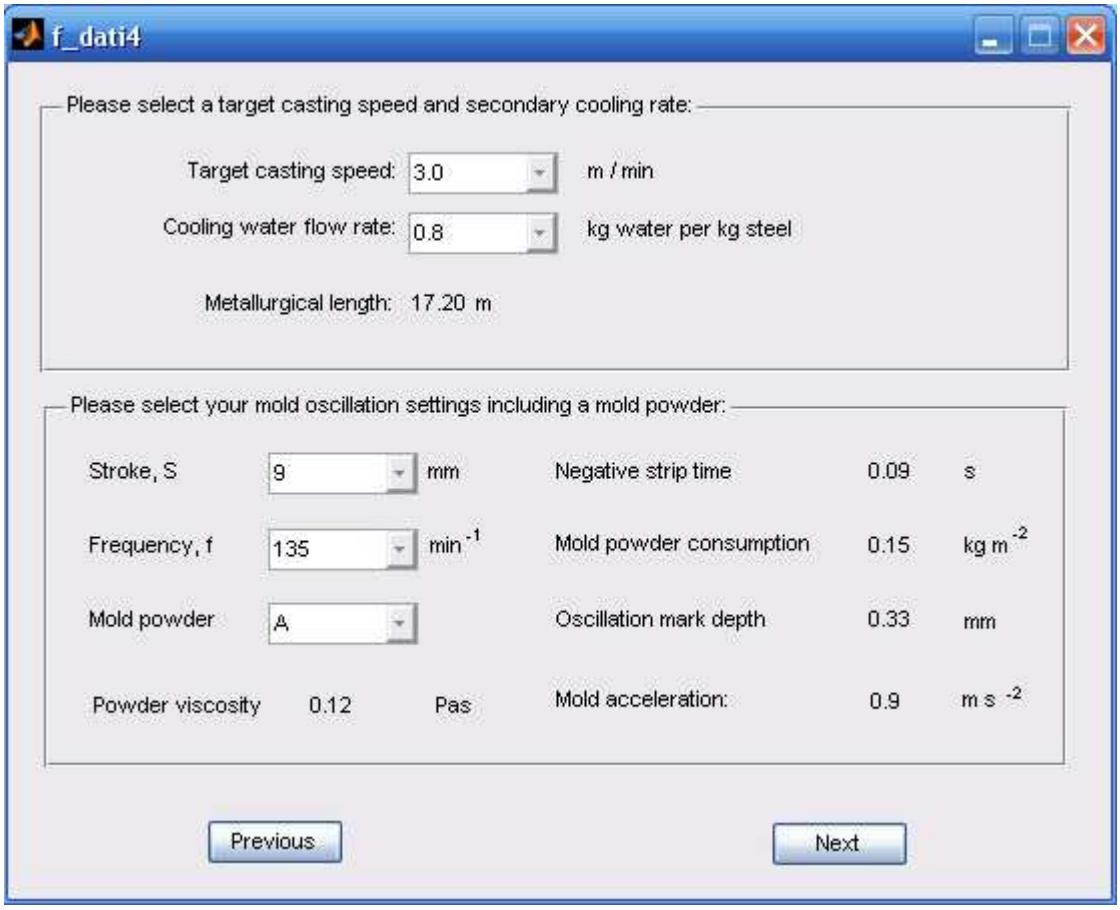
```
        metri = '-----';
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f_temperature;

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3
global S;
global ma;

stroke = get(handles.popupmenu3,'Value');
switch stroke
    case 1
        S = 3;
    case 2
        S = 4;
    case 3
        S = 5;
    case 4
        S = 6;
    case 5
        S = 7;
    case 6
        S = 8;
    case 7
        S = 9;
    case 8
        S = 10;
    case 9
        S = 11;
    case 10
        S = 12;
    case 11
        S = 13;
    case 12
        S = 14;
    case 13
        S = 15;
    otherwise
        S = 3;
end
```

```
[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text5,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu4
global f;
global ma;

freq = get(handles.popupmenu4,'Value');
switch freq
    case 1
        f = 100;
    case 2
        f = 105;
    case 3
        f = 110;
    case 4
        f = 115;
    case 5
        f = 120;
    case 6
        f = 125;
    case 7
        f = 130;
    case 8
        f = 135;
    case 9
        f = 140;
    case 10
        f = 145;
    case 11
        f = 150;
    case 12
        f = 155;
    case 13
        f = 160;
```

```
    case 14
        f = 165;
    case 15
        f = 170;
    case 16
        f = 175;
    case 17
        f = 180;
    case 18
        f = 185;
    case 19
        f = 190;
    case 20
        f = 195;
    case 21
        f = 200;
    case 22
        f = 205;
    case 23
        f = 210;
    case 24
        f = 215;
    case 25
        f = 220;
    case 26
        f = 225;
    case 27
        f = 230;
    case 28
        f = 235;
    case 29
        f = 240;
    case 30
        f = 245;
    case 31
        f = 250;
    otherwise
        f = 100;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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


% --- Executes on selection change in popupmenu5.
```

```
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu5
global viscos;
global vi;
global mp;
global ma;

vi = get(handles.popupmenu5,'Value');
switch vi
    case 1
        mp = 'A';
        viscos = 0.12;
    case 2
        mp = 'B';
        viscos = 0.21;
    case 3
        mp = 'C';
        viscos = 0.19;
    case 4
        mp = 'D';
        viscos = 0.10;
    case 5
        mp = 'E';
        viscos = 0.003;
    otherwise
        mp = 'A';
        viscos = 0.12;
end

set(handles.text2,'String',viscos);

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% Funzione che calcola i parametri fondamentali
function [tn, d, Q, ma] = calcola()
global S;
global f;
```

```
global target;
global tn;
global d;
global viscos;
global Q;
global ma;

tmp = (60 / (pi * f)) * acos ((1000 * target) / (pi * f * S));
tmp = tmp * 100;
tmp = round(tmp);
tn = tmp / 100;

tmp2 = 0.065 * (1.145^S) * (200 * (0.9^S))^tn;
tmp2 = tmp2 * 100;
tmp2 = round(tmp2);
d = tmp2 / 100;


tmp3 = (1.7 * tn) / (sqrt(viscos) * target);
tmp3 = tmp3 * 100;
tmp3 = round (tmp3);
Q = tmp3 / 100;

tmp4 = (S / 2000) * ( 2 * pi * f / 60)^2;
tmp4 = tmp4 * 100;
tmp4 = round(tmp4);
ma = tmp4 / 100;
```



Sorgente “f_dati4”:

```
function varargout = f_dati4(varargin)
% F_DATI4 M-file for f_dati4.fig
%      F_DATI4, by itself, creates a new F_DATI4 or raises the existing
%      singleton*.
%
%      H = F_DATI4 returns the handle to a new F_DATI4 or the handle to
%      the existing singleton*.
%
%      F_DATI4('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in F_DATI4.M with the given input arguments.
%
%      F_DATI4('Property','Value',...) creates a new F_DATI4 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before f_dati4_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to f_dati4_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help f_dati4

% Last Modified by GUIDE v2.5 30-Sep-2009 17:47:27
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @f_dati4_OpeningFcn, ...
                   'gui_OutputFcn',  @f_dati4_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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


% --- Executes just before f_dati4 is made visible.
function f_dati4_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to f_dati4 (see VARARGIN)

% Choose default command line output for f_dati4
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_dati4 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global target;
global water;
global S;
global viscos;
global f;
global mp;
global ma;

target = 3.0;
water = 0.8;
mp = 'A';
S = 3;
viscos = 0.12;
f = 100;

metri = lunghezza;

set(handles.text1,'String',metri);
set(handles.text2,'String',viscos);
[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = f_dati4_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
global target;
data = get(handles.popupmenu1,'Value');
switch data
    case 1
        target = 3.0;
    case 2
        target = 4.0;
    case 3
        target = 5.0;
    otherwise
        target = 3.0;

end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell
array
```

```
%        contents{get(hObject,'Value')} returns selected item from popupmenu2
global water;
data = get(handles.popupmenu2,'Value');
switch data
    case 1
        water = 0.8;
    case 2
        water = 0.9;
    case 3
        water = 1.0;
    case 4
        water = 1.1;
    case 5
        water = 1.2;
    otherwise
        water = 0.8;
end

metri = lunghezza;
set(handles.text1,'String',metri);

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

%-----------
function [metri] = lunghezza()
global target;
global water;

switch target
    case 3.0
        if water == 0.8
            metri = '17.20';
        elseif water == 0.9
            metri = '16.70';
        elseif water == 1.0
            metri = '16.20';
        elseif water == 1.1
            metri = '15.70';
        elseif water == 1.2
            metri = '15.20';
        else metri = '-----';
        end
    case 4.0
        if water == 0.8
            metri = '22.40';
        elseif water == 0.9
            metri = '21.53';
        elseif water == 1.0
            metri = '20.73';
        elseif water == 1.1
```

```
            metri = '19.86';
        elseif water == 1.2
            metri = '19.06';
        else metri = '-----';
        end
    case 5.0
        if water == 0.8
            metri = '28.00';
        elseif water == 0.9
            metri = '26.83';
        elseif water == 1.0
            metri = '25.66';
        elseif water == 1.1
            metri = '24.42';
        elseif water == 1.2
            metri = '23.33';
        else metri = '-----';
        end
    otherwise
        metri = '-----';
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
f_temperature;

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3
global S;
global ma;

stroke = get(handles.popupmenu3,'Value');
switch stroke
    case 1
        S = 3;
    case 2
        S = 4;
    case 3
        S = 5;
    case 4
        S = 6;
    case 5
        S = 7;
    case 6
        S = 8;
```

```
    case 7
        S = 9;
    case 8
        S = 10;
    case 9
        S = 11;
    case 10
        S = 12;
    case 11
        S = 13;
    case 12
        S = 14;
    case 13
        S = 15;
    otherwise
        S = 3;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu4 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu4
global f;
global ma;

freq = get(handles.popupmenu4,'Value');
switch freq
    case 1
        f = 100;
    case 2
        f = 105;
    case 3
        f = 110;
    case 4
        f = 115;
```

```
    case 5
        f = 120;
    case 6
        f = 125;
    case 7
        f = 130;
    case 8
        f = 135;
    case 9
        f = 140;
    case 10
        f = 145;
    case 11
        f = 150;
    case 12
        f = 155;
    case 13
        f = 160;
    case 14
        f = 165;
    case 15
        f = 170;
    case 16
        f = 175;
    case 17
        f = 180;
    case 18
        f = 185;
    case 19
        f = 190;
    case 20
        f = 195;
    case 21
        f = 200;
    case 22
        f = 205;
    case 23
        f = 210;
    case 24
        f = 215;
    case 25
        f = 220;
    case 26
        f = 225;
    case 27
        f = 230;
    case 28
        f = 235;
    case 29
        f = 240;
    case 30
        f = 245;
    case 31
        f = 250;
    otherwise
        f = 100;
end

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
```

```
set(handles.text32,'String',ma);

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu5 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu5
global viscos;
global vi;
global mp;
global ma;

vi = get(handles.popupmenu5,'Value');
switch vi
    case 1
        mp = 'A';
        viscos = 0.12;
    case 2
        mp = 'B';
        viscos = 0.21;
    case 3
        mp = 'C';
        viscos = 0.19;
    case 4
        mp = 'D';
        viscos = 0.10;
    case 5
        mp = 'E';
        viscos = 0.003;
    otherwise
        mp = 'A';
        viscos = 0.12;
end

set(handles.text2,'String',viscos);

[tn, d, Q, ma] = calcola;
set(handles.text3,'String',tn);
set(handles.text4,'String',d);
set(handles.text5,'String',Q);
set(handles.text32,'String',ma);
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

% Funzione che calcola i parametri fondamentali
function [tn, d, Q, ma] = calcola()
global S;
global f;
global target;
global tn;
global d;
global viscos;
global Q;
global ma;

tmp = (60 / (pi * f)) * acos ((1000 * target) / (pi * f * S));
tmp = tmp * 100;
tmp = round(tmp);
tn = tmp / 100;

tmp2 = 0.065 * (1.145^S) * (200 * (0.9^S))^tn;
tmp2 = tmp2 * 100;
tmp2 = round(tmp2);
d = tmp2 / 100;


tmp3 = (1.7 * tn) / (sqrt(viscos) * target);
tmp3 = tmp3 * 100;
tmp3 = round (tmp3);
Q = tmp3 / 100;

tmp4 = (S / 2000) * ( 2 * pi * f / 60)^2;
tmp4 = tmp4 * 100;
tmp4 = round(tmp4);
ma = tmp4 / 100;
```

Interfaccia temperature:

Lo scopo della simulazione è quello di ottenere una colata costituita da una sequenza di tre ladle. Si può scegliere il tempo di arrivo di ciascun ladle nonché la temperatura degli stessi. Occorre tenere presente un abbassamento della temperatura pari a 0.5 gradi centigradi al minuto. Il tempo necessario per svuotare un ladle deve corrispondere al tempo di arrivo del successivo ladle. Esso dipende dal numero di strand, dalla sezione dello strand e dalla casting speed. I ladle devono arrivare alla giusta temperatura tenendo conto dell'abbassamento della stessa al passare dei minuti.

$$\tau = \frac{m_{\text{ladle}}}{\dot{M}_{\text{T}}} = \frac{m_{\text{ladle}}}{n \cdot \rho_{\text{liq}} \cdot w \cdot t \cdot v_{\text{c}}} \ [\text{min}]$$

w = strand width, m

t = thickness of the strand, m

v_c = casting speed, m min^{-1}

n = number of strands

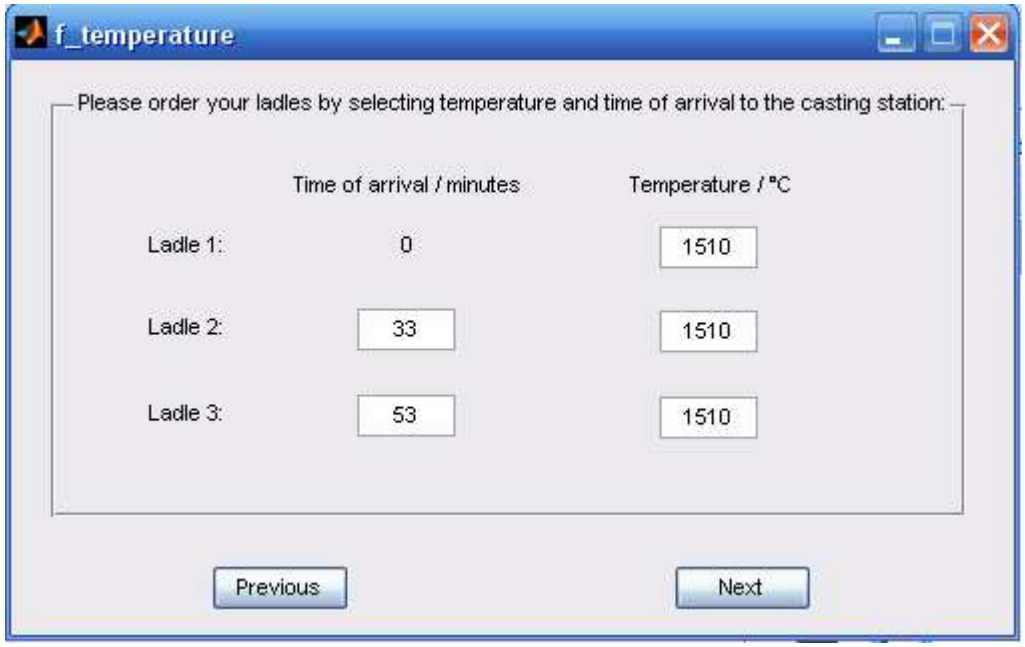
ρ_{liq} = liquid steel density, 7400 kg m^{-3}

m_{ladle} = mass of liquid steel to be teemed from the ladle, kg. Note that teeming automatically stops when slag is detected at the slidegate, typically when the steel level reaches 5 %.

E' di fondamentale importanza evitare che l'acciaio si solidifichi durante la colata. Questo avviene se si raggiunge la soglia di temperatura Tliq (liquidus temperature). Questa soglia dipende dalla composizione dell'acciaio. Per evitare il fenomeno della solidificazione, la differenza fra la temperatura del metallo e la liquidus temperature, chiamata superheat, deve essere di almeno dieci gradi centigradi. Aumentando la superheat si riduce però la superficie solida dell'acciaio e in questo modo può accadere che, se è troppo sottile per sopportare il peso dell'acciaio liquido, si verifichi una rottura della strand. Il massimo valore della superheat è di 50 gradi centigradi per le colate slab mentre è di 60 gradi centigradi per quelle bloom e billet.

$$T_{\text{liq}} = 1537 - 78\%C - 7.6\%Si - 4.9\%Mn - 34.4\%P - 38\%S$$

	Construction steel	TiNb ULC steel for car bodies	Linepipe steel	Engineering steel
C	0.1450	0.0030	0.0700	0.4150
Si	0.2000	0.2100	0.1800	0.4000
Mn	1.4000	0.7500	1.0500	0.7500
P	<0.0250	0.0650	<0.0120	0.0350
S	<0.0200	<0.0120	<0.0030	0.0350
Cr	<0.1000	<0.0500	<0.0600	1.0500
Al	0.0350	0.0450	0.0300	0.0225
B	<0.0005	0.0030	<0.0050	0.0050
Ni	<0.1500	<0.0800	<0.0500	0.3000
Nb	0.0500	0.0200	0.0150	0.0000
Ti	<0.0100	0.0300	<0.0100	0.0000
V	<0.0100	-	<0.0100	0.0100
Mo	<0.0400	<0.0100	<0.0100	0.2250
As	-	<0.0010	-	0.0000
Ca	-	-	<0.0050	0.0000
N	<0.0050	<0.0040	<0.0045	0.0050
H	<0.0005	<0.0005	<0.0002	0.0002
O	<0.0010	<0.0005	<0.0007	0.0005



Sorgente “f_temperature.m”:

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

% Copyright 2002-2003 The MathWorks, Inc.

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

% Last Modified by GUIDE v2.5 30-Sep-2009 18:46:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
```

```
                   'gui_OpeningFcn', @f_temperature_OpeningFcn, ...
                   'gui_OutputFcn',  @f_temperature_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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


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

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_temperature wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global ta2;
global ta3;
global te1;
global te2;
global te3;

ta2 = 33;
ta3 = 53;
te1 = 1510;
te2 = 1510;
te3 = 1510;

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

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



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

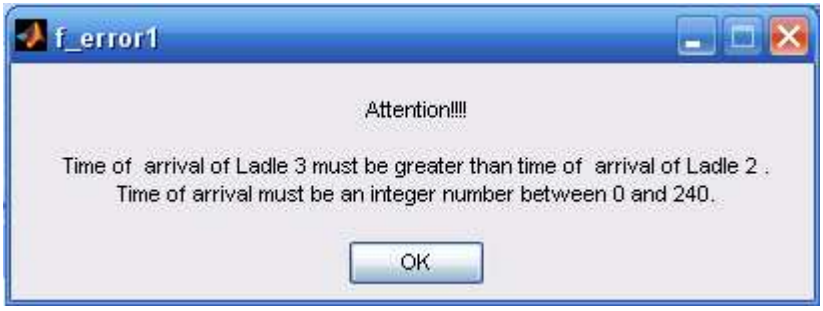
```
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
global ta2;
global ta3;

ta2 = get(handles.edit1,'String');
ta2 = str2num(ta2);
if length(ta2) == 0
    ta2 = 33;
    f_error1;
end
ta2 = round(ta2);
set(handles.edit1,'String',ta2);

if ta2 < 0
    set(handles.edit1,'String','0');
    ta2 = 0;
    f_error1;
elseif ta2 > ta3
    set(handles.edit1,'String',ta3);
    ta2 = ta3;
    f_error1;
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double
global ta2;
global ta3;

ta3 = get(handles.edit2,'String');
ta3 = str2num(ta3);
if length(ta3) == 0
    ta3 = 53;
    f_error1;
end
ta3 = round(ta3);
set(handles.edit2,'String',ta3);
```

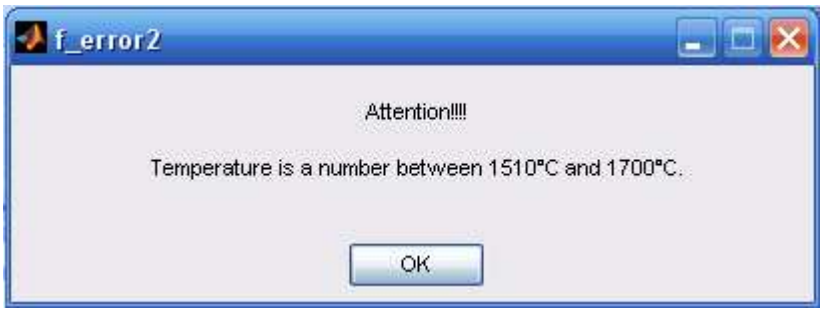
```
if ta3 < ta2
    set(handles.edit2,'String',ta2);
    ta3 = ta2;
    f_error1;
elseif ta3 > 240
    set(handles.edit2,'String','240');
    ta3 = 240;
    f_error1;
end

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
te1 = get(handles.edit3,'String');
te1 = str2num(te1);
if length(te1) == 0
    te1 = 1510;
    f_error2;
end
te1 = round(te1);
set(handles.edit3,'String',te1);

if te1 < 1510
    set(handles.edit3,'String','1510');
    f_error2;
elseif te1 > 1700
    set(handles.edit3,'String','1700');
    f_error2;
end

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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

```
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double
te2 = get(handles.edit4,'String');
te2 = str2num(te2);
if length(te2) == 0
    te2 = 1510;
    f_error2;
end
te2 = round(te2);
set(handles.edit4,'String',te2);

if te2 < 1510
    set(handles.edit4,'String','1510');
    f_error2;
elseif te2 > 1700
    set(handles.edit4,'String','1700');
    f_error2;
end

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double
te3 = get(handles.edit5,'String');
te3 = str2num(te3);
if length(te3) == 0
    te3 = 1510;
    f_error2;
end
te3 = round(te3);
set(handles.edit5,'String',te3);
```

```
if te3 < 1510
    set(handles.edit5,'String','1510');
    f_error2;
elseif te3 > 1700
    set(handles.edit5,'String','1700');
    f_error2;
end

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

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



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global ta2;
global ta3;
global te1;
global te2;
global te3;

ta2 = get(handles.edit1,'String');
ta3 = get(handles.edit2,'String');
te1 = get(handles.edit3,'String');
te2 = get(handles.edit4,'String');
te3 = get(handles.edit5,'String');

f_riassunto;
```

Interfacce errori:

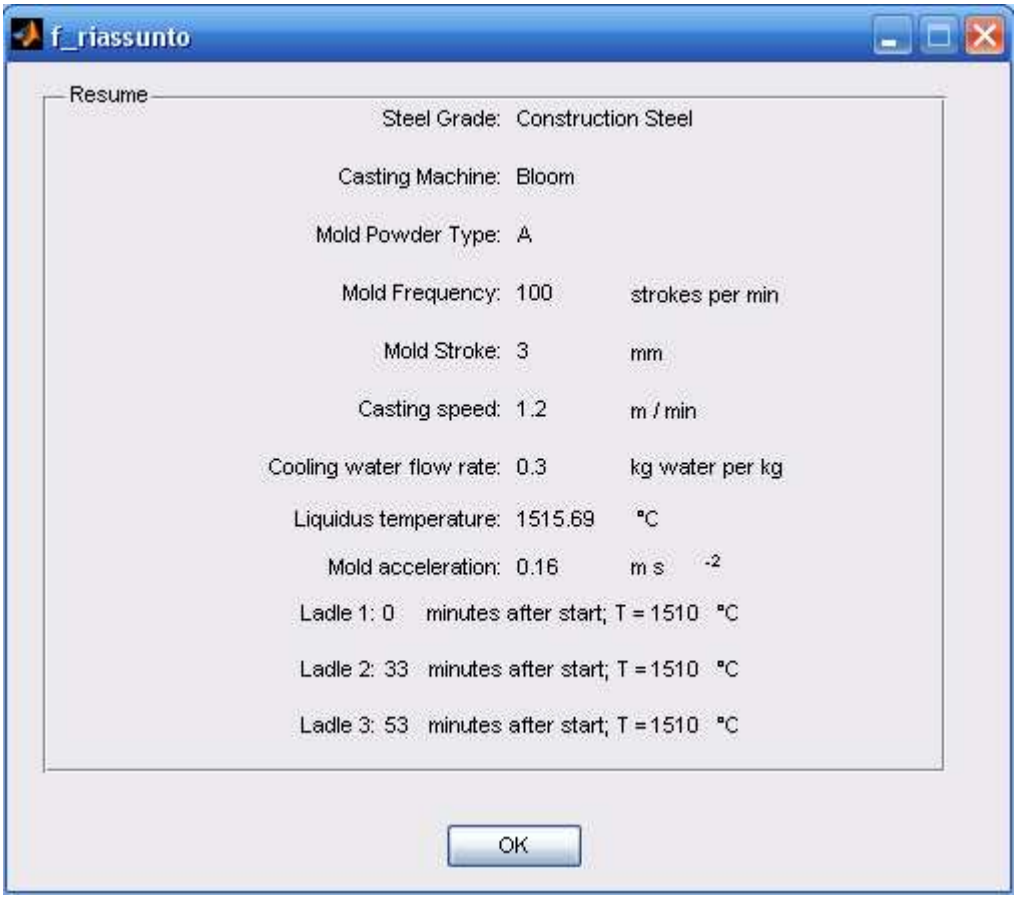




Sono stati previsti dei controlli negli inserimenti.
La prima interfaccia controlla che il tempo di arrivo del terzo ladle sia maggiore del primo inoltre essa verifica anche che l'inserimento sia un numero intero compreso fra 0 e 240.
La seconda interfaccia serve a segnalare un inserimento delle temperature fuori dal range 1510-1500°C. Anche in questo caso viene effettuato il controllo sul numero inserito affinché sia un intero.

Interfaccia riassunto:

In sostanza questa interfaccia permette di visualizzare i risultati principali dell'esercizio, riepilogando i dati immessi.



Sorgente "f_riassunto.m":

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

```
% Copyright 2002-2003 The MathWorks, Inc.

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

% Last Modified by GUIDE v2.5 06-Oct-2009 13:13:51

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

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


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

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes f_riassunto wait for user response (see UIRESUME)
% uiwait(handles.figure1);
global tipo;
global mp;
global f;
global S;
global target;
global water;
global ta2;
global ta3;
global te1;
global te2;
global te3;
global Tliq;
global ma;

text36_CreateFcn(hObject, eventdata, handles);

switch tipo
```

```
    case 1
        set(handles.text1,'String','Construction Steel');
        set(handles.text2,'String','Bloom');
    case 2
        set(handles.text1,'String','Ultra-low Carbon Steel  ');
        set(handles.text2,'String','Slab');
    case 3
        set(handles.text1,'String','Linepipe Steel  ');
        set(handles.text2,'String','Slab');
    case 4
        set(handles.text1,'String','Engineering steel');
        set(handles.text2,'String','Billet');
    otherwise
        set(handles.text1,'String','-------');
        set(handles.text2,'String','-------');
end

set(handles.text3,'String',mp);
set(handles.text4,'String',f);
set(handles.text5,'String',S);
set(handles.text6,'String',target);
set(handles.text7,'String',water);
set(handles.text8,'String',ta2);
set(handles.text9,'String',ta3);
set(handles.text10,'String',te1);
set(handles.text11,'String',te2);
set(handles.text12,'String',te3);
set(handles.text36,'String',Tliq);
set(handles.text38,'String',ma);

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

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


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;



% --- Executes during object creation, after setting all properties.
function text36_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text36 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
global Tliq;
global tipo;

switch tipo
    case 1
        c = 0.145;
```

```
        si = 0.2;
        mn = 1.4;
        p = 0.025;
        s = 0.02;
    case 2
        c = 0.003;
        si = 0.21;
        mn = 0.75;
        p = 0.065;
        s = 0.012;
    case 3
        c = 0.07;
        si = 0.18;
        mn = 1.05;
        p = 0.012;
        s = 0.003;
    case 4
        c = 0.4150;
        si = 0.4;
        mn = 0.75;
        p = 0.035;
        s = 0.035;
    otherwise
        c = 0.145;
        si = 0.2;
        mn = 1.4;
        p = 0.025;
        s = 0.02;
end

Tliq = 1537 - (78 * c) - (7.6 * si) - (4.9 * mn) - (34.4 * p) - (38 * s);
```

Conclusioni:

Il software realizzato è facilmente scalabile e permette di raccogliere dei dati che potrebbero essere successivamente impiegati allo scopo di realizzare una simulazione della effettiva produzione di una colata di acciaio.
Infatti, l'interfaccia così costruita permette all'utente di impostare i tempi di arrivo e le temperature dei ladle e questi dati possono essere utilizzati come punto di partenza per l'eventuale futura simulazione.
Le funzioni sono state progettate in modo tale da permettere una forte riusabilità, ossia sono state realizzate in maniera modulare, secondo i principi dell'ingegneria del software, affinché modificandone una non sia necessario riscrivere altre parti di codice in altri punti del programma.
Sono stati previsti dei controlli per migliorare la versione del software disponibile su internet, in particolare sono state aggiunte delle interfacce che verificano il corretto svolgimento degli inserimenti.
E' stata aggiunta la formula della mold acceleration che non è presente nella guida della SteelUniversity.