

Meccanica (per Ing. Informatica) [088804]

Analisi cinematica MATLAB - Problema 1 - TdE 14/02/2024

Anno accademico 2022-2023 - Politecnico di Milano - Redaelli Luca

1 Introduzione al problema

L'analisi cinematica è effettuata in relazione al problema 1 del tema d'esame del 14 febbraio 2024. Il testo, privato della relativa parte di dinamica, è di seguito riportato.

In figura è presente un manovellismo. La cerniera in O è vincolata a terra, mentre quella in B nel centro di un disco di raggio R che rotola senza strisciare e senza resistenza su un piano orizzontale.

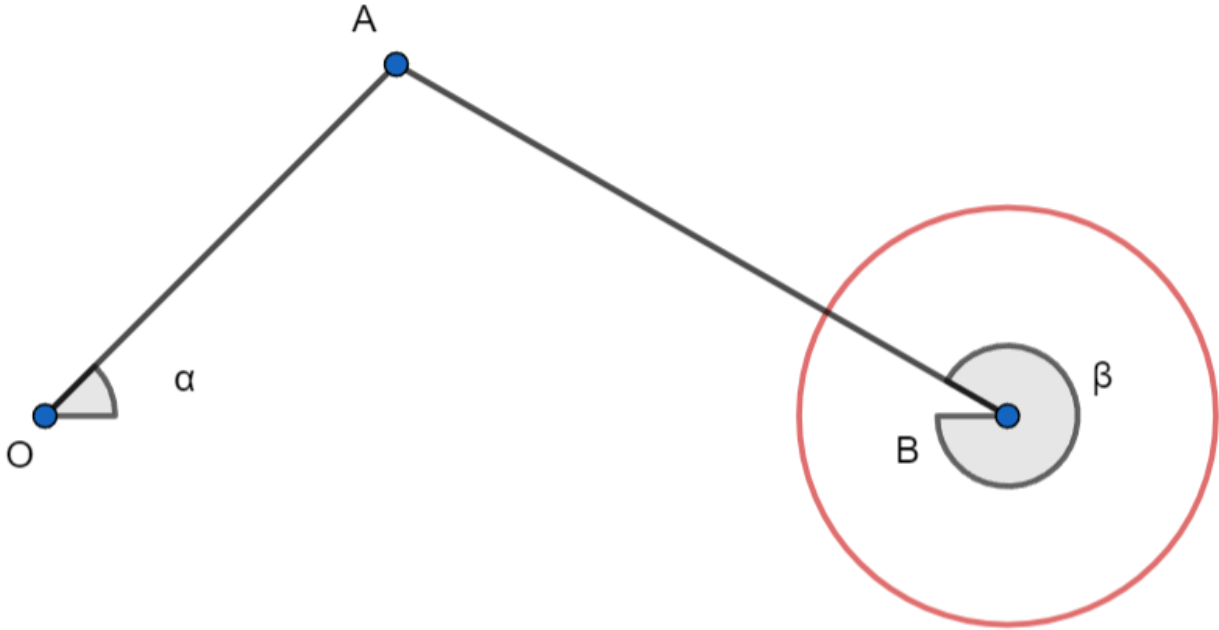


Figura 1: Manovellismo ordinario centrato

Sono noti, dal testo del problema, i seguenti dati:

- $\overline{OA} = 1 \text{ [m]} = a$
- $\overline{AB} = \sqrt{2} \text{ [m]} = b$
- $\angle BOA = 45 \text{ [}^\circ\text{]} = \alpha$
- $\angle OBA = 30 \text{ [}^\circ\text{]} = 2\pi - \beta$
- $\dot{\alpha} = 0.5 \text{ [rad/s]}$
- $\ddot{\alpha} = 1 \text{ [rad/s}^2\text{]}$

2 Analisi cinematica del manovellismo

2.1 Chiusura cinematica

Il meccanismo per la trasformazione del moto rotatorio continuo della manovella OA nel moto traslatorio alternativo del piede di biella B è un manovellismo ordinario centrato. L'analisi cinematica del meccanismo si basa sulla seguente equazione di chiusura:

$$(A - O) + (B - A) = (B - O) \quad (1)$$

che, riscritta sfruttando l'equazione di Eulero, risulta:

$$ae^{i\alpha} + be^{i\beta} = ce^{i\gamma} \quad (2)$$

Si mostra graficamente l'equazione di chiusura.

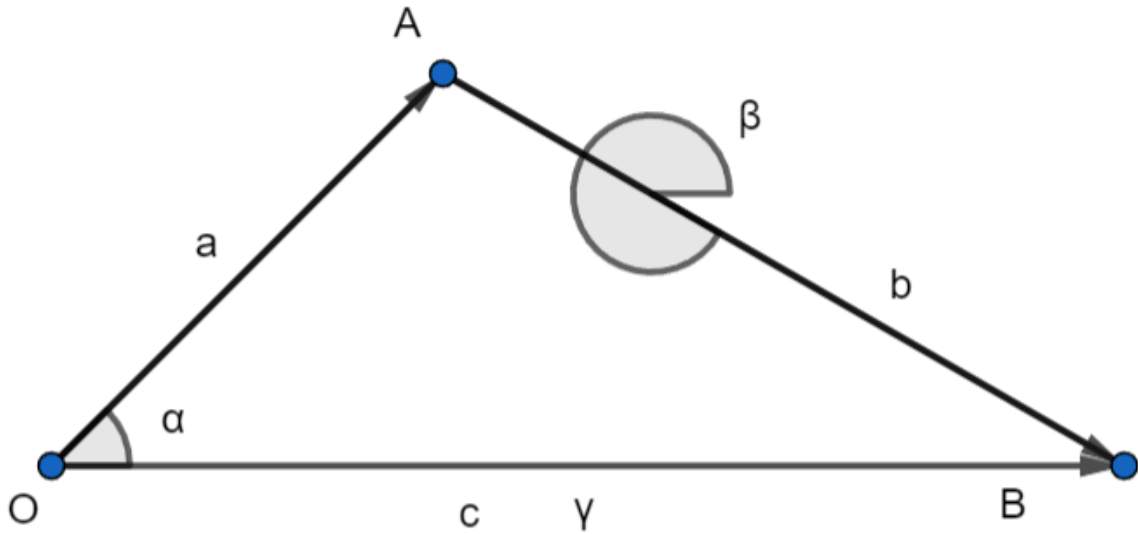


Figura 2: Chiusura cinematica del manovellismo

2.2 Analisi delle posizioni

Dalla soluzione del sistema algebrico non lineare che deriva dalla proiezione sugli assi Reale ed Immaginario dell'equazione (2) si ottiene:

$$\begin{cases} a\cos(\alpha) + b\cos(\beta) = c\cos(\gamma) \\ a\sin(\alpha) + b\sin(\beta) = c\sin(\gamma) \end{cases} \quad (3)$$

Conoscendo i dati assegnati dal problema risulta semplice ricavare i valori di β e c in funzione dell'angolo α assegnato. In modo particolare:

$$\begin{cases} c(\alpha) = a\cos(\alpha) + b\cos(\beta) \\ \beta(\alpha) = \sin^{-1}\left(\frac{-a\sin(\alpha)}{b}\right) \end{cases} \quad (4)$$

Tramite script MATLAB è possibile rappresentare in un grafico le posizioni del piede di biella c e la posizione angolare della biella β in funzione della rotazione della manovella α .

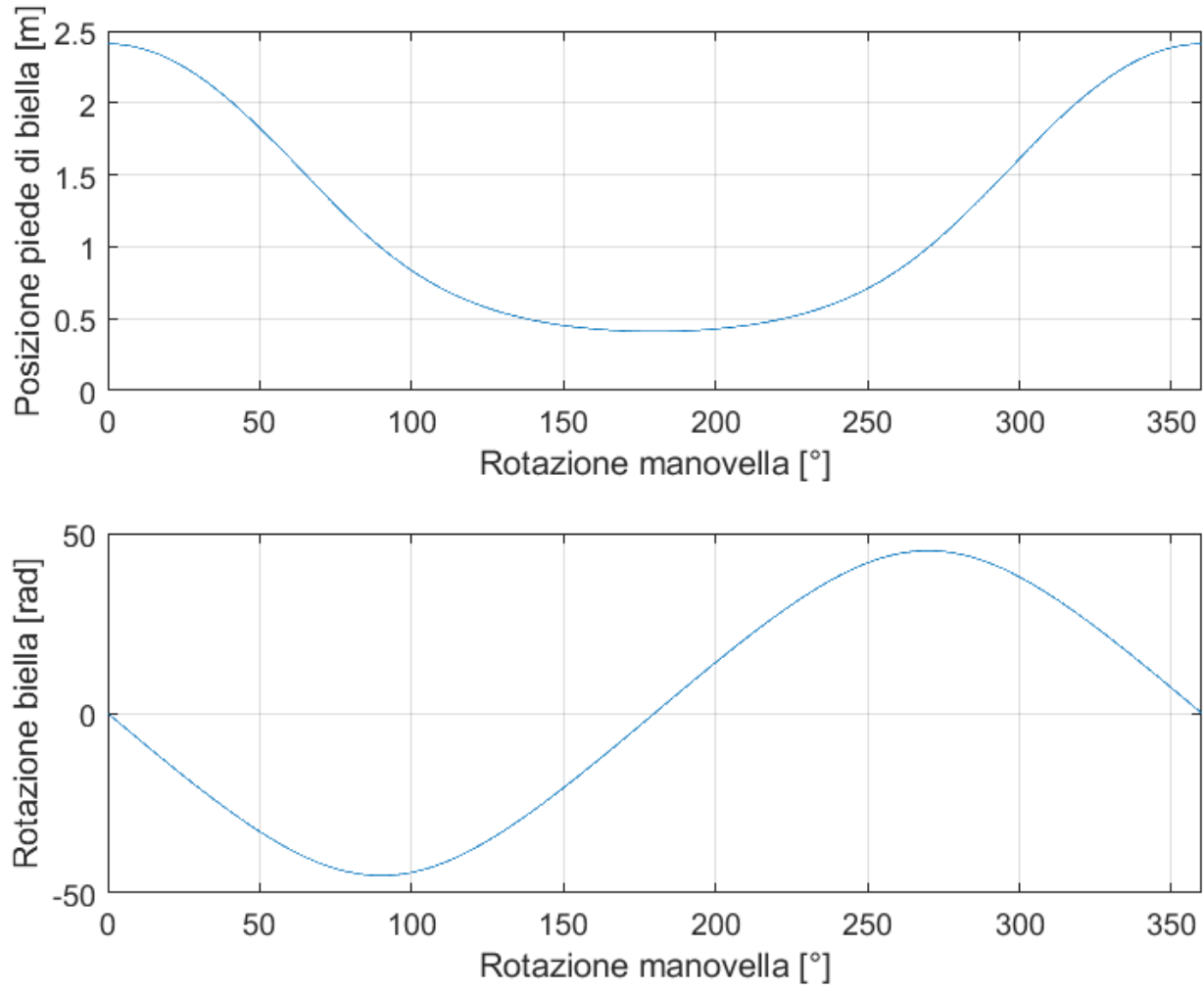


Figura 3: Grafici delle posizioni

2.3 Analisi delle velocità

Derivando rispetto al tempo l'equazione (3) si ottiene:

$$\begin{cases} -\dot{\alpha}a\sin(\alpha) - \dot{\beta}b\sin(\beta) = \dot{c}\cos(\gamma) \\ \dot{\alpha}a\cos(\alpha) + \dot{\beta}b\cos(\beta) = \dot{c}\sin(\gamma) \end{cases} \quad (5)$$

dalla quale è comodo ricavare e risolvere il sistema matriciale sfruttando i dati forniti dal problema e quelli ottenuti al punto precedente.

$$\begin{bmatrix} \cos(\gamma) & b\sin(\beta) \\ \sin(\gamma) & -b\cos(\beta) \end{bmatrix} \begin{bmatrix} \dot{c} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\dot{\alpha}a\sin(\alpha) \\ \dot{\alpha}a\cos(\alpha) \end{bmatrix} \quad (6)$$

Ancora una volta, tramite script MATLAB, è possibile rappresentare in un grafico le velocità del piede di biella \dot{c} e la velocità angolare della biella $\dot{\beta}$ in funzione della rotazione della manovella α .

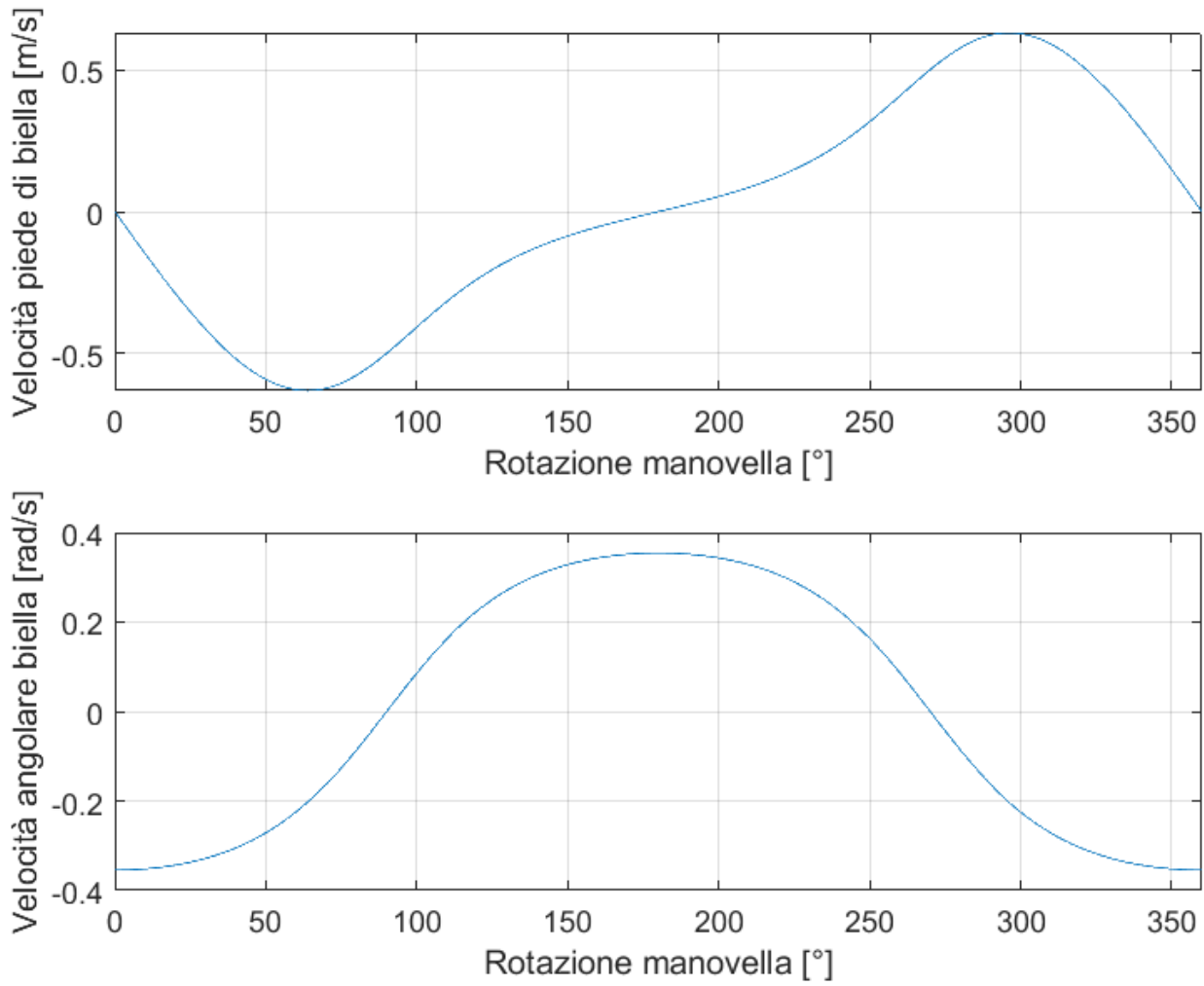


Figura 4: Grafici delle velocità

2.4 Analisi delle accelerazioni

Derivando rispetto al tempo l'equazione (5) si ottiene:

$$\begin{cases} -\ddot{\alpha} \sin(\alpha) - \dot{\alpha}^2 \cos(\alpha) - \ddot{\beta} b \sin(\beta) - \dot{\beta}^2 b \cos(\beta) = \ddot{c} \cos(\gamma) \\ \ddot{\alpha} \cos(\alpha) - \dot{\alpha}^2 \sin(\alpha) + \ddot{\beta} b \cos(\beta) - \dot{\beta}^2 b \sin(\beta) = \ddot{c} \sin(\gamma) \end{cases} \quad (7)$$

dalla quale è comodo ricavare e risolvere il sistema matriciale sfruttando i dati forniti dal problema e quelli ottenuti al punto precedente.

$$\begin{bmatrix} \cos(\gamma) & b \sin(\beta) \\ \sin(\gamma) & -b \cos(\beta) \end{bmatrix} \begin{bmatrix} \ddot{c} \\ \ddot{\beta} \end{bmatrix} = \begin{bmatrix} -\ddot{\alpha} \sin(\alpha) - \dot{\alpha}^2 \cos(\alpha) - \dot{\beta}^2 b \cos(\beta) \\ \ddot{\alpha} \cos(\alpha) - \dot{\alpha}^2 \sin(\alpha) - \dot{\beta}^2 b \sin(\beta) \end{bmatrix} \quad (8)$$

Conseguentemente, tramite script MATLAB, è possibile rappresentare in un grafico le accelerazioni del piede di biella \ddot{c} e l'accelerazione angolare della biella $\ddot{\beta}$ in funzione della rotazione della manovella α .

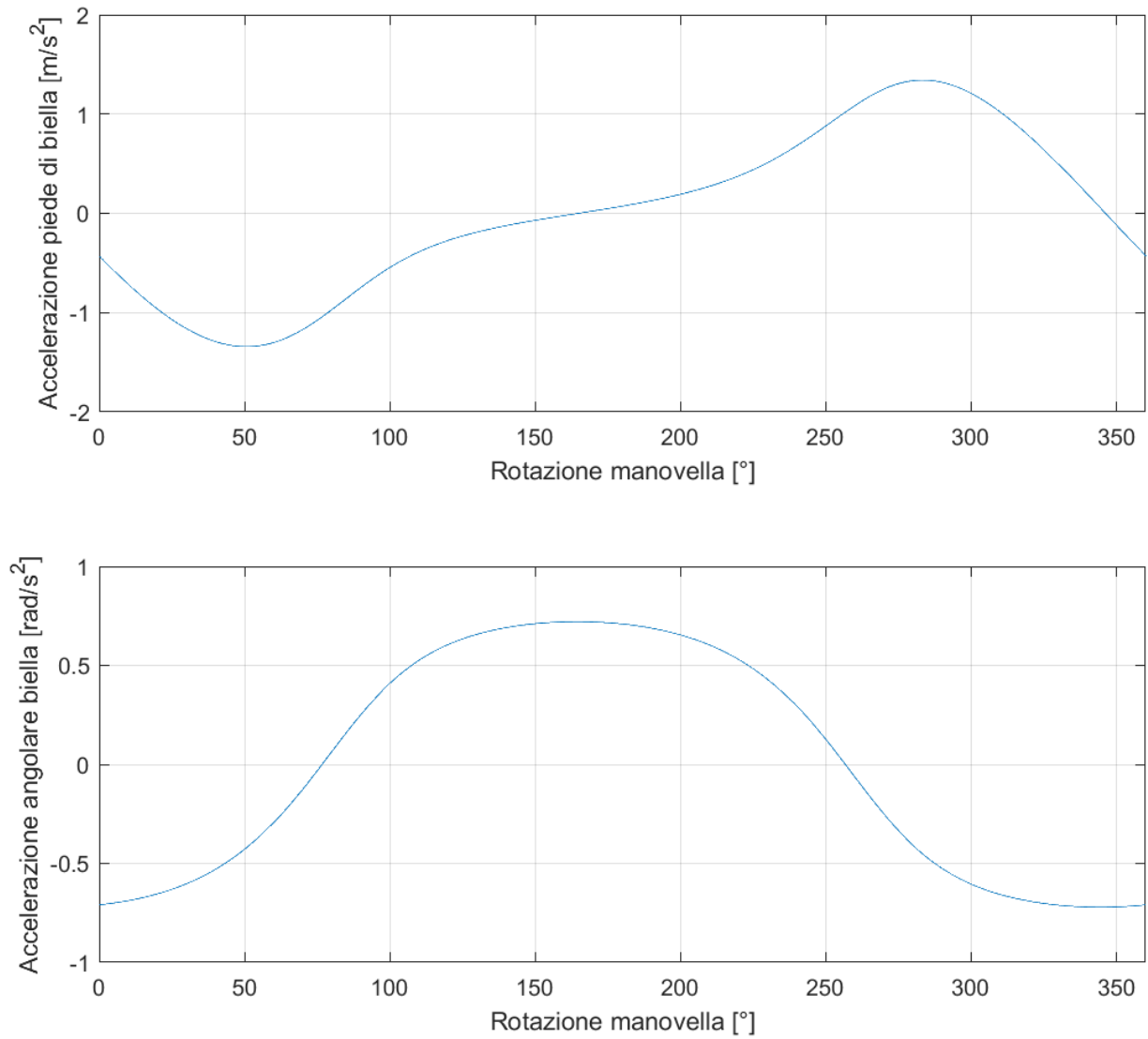


Figura 5: Grafici delle accelerazioni

2.5 Animazione del moto

Sempre tramite script MATLAB è possibile mostrare una breve animazione del comportamento del sistema. Sono di seguito mostrati alcuni snapshot dell'animazione.

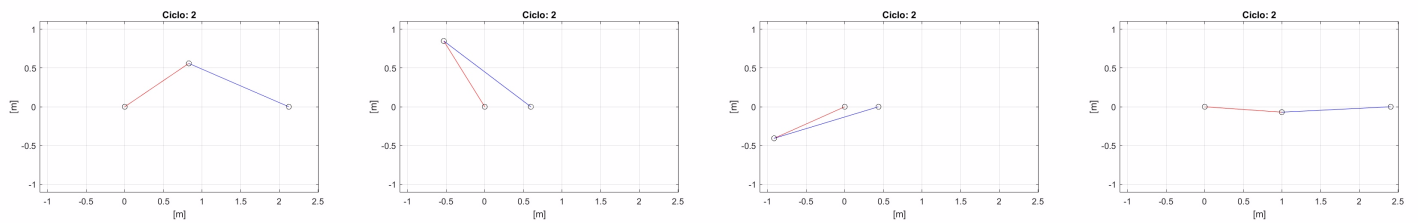


Figura 6: Istantanee dal sistema

3 Codice MATLAB utilizzato

Viene di seguito riportato lo script MATLAB utilizzato per le analisi cinematiche del problema.

Si ringrazia il Prof. Stefano Arrigoni per la base di codice fornita durante le esercitazioni del corso. Il codice è stato adattato al problema in questione ed ulteriormente sviluppato al fine di poter selezionare manualmente anche l'accelerazione angolare da fornire al sistema.

```
1  clc
2  close all
3  clear all
4
5  %% Inserimento dati iniziali
6
7  corretto = 1;
8  while corretto == 1
9      a = input('Raggio manovella in [m]: ');
10     b = input('Lunghezza biella in [m]: ');
11
12     if a < b
13         corretto = 0;
14     else
15         disp('La manovella non puo essere piu lunga della biella.')
16     end
17 end
18
19 % Definizione della discretizzazione dell'analisi
20
21 alfa_d = 0:0.1:36000;
22 alfa = alfa_d * (pi / 180);
23
24 alfap = input('Assegna velocita angolare manovella [rad/s]: ');
25 alfapp = input('Assegna accelerazione angolare manovella [rad/s^2]: ');
26
27 %% Rappresentazione della configurazione iniziale
28
29 beta_a = asin(-(a * sin(alfa(451))) / b);
30 c_a = a * cos(alfa(451)) + b * cos(beta_a);
31 gamma = 0;
32 OA = a * exp(1i * alfa(451));
33 AB = b * exp(1i * beta_a);
34 OB = c_a * exp(1i * gamma);
35 figure()
36
37 % Rappresentazione delle cerniere
38 plot(0, 0, 'ko')
39 xlabel('x [m]')
40 ylabel('y [m]')
41 title('Configurazione a alpha = 45')
42 hold on
43
44 % Posizioni di O, A e B
45 O = 0;
46 A = OA;
47 B = OB;
48
49 % Rappresentazione delle posizioni O, A e B
50 plot(real(O), imag(O), 'ko')
51 plot(real(A), imag(A), 'ko')
52 plot(real(B), imag(B), 'ko')
```

```

53
54 % Etichette
55 offset = 0.05;
56 text(real(O), imag(O) + offset, 'O', 'HorizontalAlignment', 'right', 'VerticalAlignment', 'bottom')
57 text(real(A), imag(A) + offset, 'A', 'HorizontalAlignment', 'right', 'VerticalAlignment', 'bottom')
58 text(real(B) + offset, imag(B), 'B', 'HorizontalAlignment', 'right', 'VerticalAlignment', 'bottom')
59
60 % Rappresentazione delle aste
61 plot([0 real(A)], [0 imag(A)], 'r-', 'LineWidth', 3)
62 plot([real(A) real(A + AB)], [imag(A) imag(A + AB)], 'g-', 'LineWidth', 3)
63 plot([0 real(B)], [0 imag(B)], 'b-', 'LineWidth', 3)
64
65 % Estensione degli assi
66 offset_extend = 0.1;
67 xlim([min([0 real(O) real(A) real(A + AB) real(B)]), max([0 real(O) real(A) real(A + AB) real(B)])])
68 ylim([min([0 imag(O) imag(A) imag(A + AB) imag(B)]), max([0 imag(O) imag(A) imag(A + AB) imag(B)])])
69
70 axis equal
71 grid on
72
73 %% Analisi della posizione
74
75 beta = asin(-(a * sin(alfa)) / b);
76 c = a * cos(alfa) + b * cos(beta);
77
78 %% Analisi della velocita
79
80 for i = 1:length(alfa)
81     A = [1, b * sin(beta(i));
82         0, -b * cos(beta(i))];
83
84     B = [-alfap * a * sin(alfa(i));
85         alfap * a * cos(alfa(i))];
86
87     xp = inv(A) * B;
88
89     cp(i) = xp(1);
90     betap(i) = xp(2);
91 end
92
93 %% Analisi dell'accelerazione
94
95 for i = 1:length(alfa)
96     A = [1, b * sin(beta(i));
97         0, -b * cos(beta(i))];
98
99     B = [-alfapp * a * sin(alfa(i)) - alfap ^ 2 * a * cos(alfa(i)) - betap(i) ^ 2 * b * cos(beta(i));
100         alfapp * a * cos(alfa(i)) - alfap ^ 2 * a * sin(alfa(i)) - betap(i) ^ 2 * b * sin(beta(i))];
101
102     xpp = inv(A) * B;
103
104     cpp(i) = xpp(1);
105     betapp(i) = xpp(2);
106 end
107
108 %% Visualizzazione risultati
109
110 visualizza = 1;
111 while visualizza == 1
112     k = menu('Visualizzazione risultati', 'Grafici posizione', ...
113             'Grafici velocita', 'Grafici accelerazione', ...

```



```

114         'Animazione del moto', 'Fine');
115
116 % Grafico posizione
117 if k == 1
118     figure(1)
119     subplot(211)
120     plot(alfa * (180 / pi), c)
121     xlabel('Rotazione manovella')
122     ylabel('Posizione piede di biella [m]')
123     ax = axis;
124     axis([0, 360, ax(3), ax(4)])
125     grid on
126
127     subplot(212)
128     plot(alfa * (180 / pi), beta * (180 / pi))
129     xlabel('Rotazione manovella')
130     ylabel('Rotazione biella [rad]')
131     ax = axis;
132     axis([0, 360, ax(3), ax(4)])
133     grid on
134
135 % Grafico velocita
136 elseif k == 2
137     figure(2)
138     subplot(211)
139     plot(alfa * (180 / pi), cp)
140     xlabel('Rotazione manovella')
141     ylabel('Velocita piede di biella [m/s]')
142     ax = axis;
143     axis([0, 360, ax(3), ax(4)])
144     grid on
145
146     subplot(212)
147     plot(alfa * (180 / pi), betap)
148     xlabel('Rotazione manovella')
149     ylabel('Velocita angolare biella [rad/s]')
150     ax = axis;
151     axis([0, 360, ax(3), ax(4)])
152     grid on
153
154 % Grafico accelerazione
155 elseif k == 3
156     figure(3)
157     subplot(211)
158     plot(alfa * (180 / pi), cpp)
159     xlabel('Rotazione manovella')
160     ylabel('Accelerazione piede di biella [m/s^2]')
161     ax = axis;
162     axis([0, 360, ax(3), ax(4)])
163     grid on
164
165     subplot(212)
166     plot(alfa * (180 / pi), betapp)
167     xlabel('Rotazione manovella')
168     ylabel('Accelerazione angolare biella [rad/s^2]')
169     ax = axis;
170     axis([0, 360, ax(3), ax(4)])
171     grid on
172
173 % Animazione del moto
174 elseif k == 4

```

```

175 figure(4)
176 set(gcf, 'DoubleBuffer', 'on')
177
178 ngiri = 3;
179 nxframe = 2;
180 nframe = ngiri * 360 / nxframe;
181
182 for i = 1:nframe
183     igrado = nxframe * i;
184     n = floor(igrado / 360);
185     igrado = igrado - 360 * n;
186
187     ic = find(alfa * 180 / pi >= igrado);
188
189     x_manov = a * cos(alfa(ic(1)));
190     y_manov = a * sin(alfa(ic(1)));
191
192     x_biell = c(ic(1));
193     y_biell = 0;
194
195     plot([0, x_manov], [0, y_manov], 'r', ...
196          [x_manov, x_biell], [y_manov, y_biell], 'b', ...
197          [0, x_manov, x_biell], [0, y_manov, y_biell], 'ok')
198
199     axis([-a - b, a + b, -a - b, a + b])
200     axis equal;
201     title(sprintf('Cielo: %d', n + 1))
202     xlabel(' [m] ')
203     ylabel(' [m] ')
204     xlim([-a - 0.1, a + b + 0.1])
205     ylim([-a - 0.1, a + 0.1])
206     grid on
207     drawnow
208 end
209 else
210     visualizza = 0;
211     close all
212 end
213 end

```

4 Conclusioni, riferimenti e licenza d'uso

Il report è disponibile al seguente link: <https://github.com/lucaredaelli/Report-Cinematica-Meccanica>

All'interno della repository è possibile reperire:

- Lo script MATLAB utilizzato per l'analisi
- Le immagini dei grafici (a dimensione reale)
- L'animazione del manovellismo
- Il file .pdf del report
- Il file .tex del report

4.1 Licenza

Copyright (c) 2023 Luca Redaelli

Permission is hereby granted, free of charge, to any person obtaining a copy of this report and software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE REPORT AND SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.