****

**Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie**

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

**Sprawozdanie Robotyka Kosmiczna**

Autor:  
 **Dawid Lisek**

Nr indeksu: 402382

Kierunek studiów: **Automatyka i Robotyka**

Specjalizacja: **Komputerowe Systemy Sterowania**Grupa: pn 13:15 - 15:45

1. **Cel ćwiczenia.**

Celem ćwiczenia było rozwiązanie zadania prostego kinematyki na poziomie położeń oraz prędkości. Należało wyprowadzić analityczne zależności na pozycję, orientację, prędkość liniową oraz prędkość kątową. Następnie należało rozwiązać odwrotne zadanie kinematyki. Całość została zrealizowana w programie Matlab.

1. **Analityczne zależności na pozycję i orientację członu roboczego w funkcji pozycji kątowych przegubów.**

Obraz zawierający wykres

Opis wygenerowany automatycznie

Na podstawie powyższego modelu manipulatora płaskiego należało wyprowadzić wzory na pozycję oraz orientację członu roboczego w funkcji pozycji kątowych przegubów . Wyprowadzone wzory:

1. **Analityczne zależności na prędkość liniową i kątową członu roboczego w funkcji prędkości kątowych i pozycji kątowych przegubów.**

Następnie na podstawie powyższych równań należało wyprowadzić wzory na prędkość liniową oraz kątową członu roboczego. W tym celu należało obliczyć poniższe pochodne cząstkowe:

Obraz zawierający kalendarz

Opis wygenerowany automatycznie

Ostatecznie wzory na prędkość liniową oraz kątową członu roboczego przyjmują poniższą postać:

1. **Skrypt źródłowy funkcji "direct\_2DoF" napisany w programie Matlab**

Kolejnym krokiem było napisanie funkcji direct\_2DoF, która realizowała proste zadanie kinematyki. Funkcja na podstawie pozycji kątowych przegubów, prędkości kątowych przegubów oraz długości ramion manipulatora powinna zwracać pozycję, orientację, prędkość liniową oraz prędkość kątową członu roboczego. Funkcja:

Obraz zawierający tekst, osoba, zrzut ekranu

Opis wygenerowany automatycznie

Argument q oraz dq był przekazywany jako 2 kolumnowy wektor, który zawierał pozycję oraz prędkość kątową poszczególnych przegubów. Dodatkowo do obliczeń zostało wykorzystane mnożenie tablicowe dostępne w pakiecie Matlab.

1. **Wybrane parametry wejściowe funkcji „Trajectory\_Generation” dla dwóch przegubów manipulatora oraz wykres wygenerowanej trajektorii (przebiegi pozycji kątowych i prędkości kątowych przegubów).**

Następnie należało wywołać funkcję „Trajectory\_Generation” oraz wygenerować trajektorię pozycji oraz prędkości kątowych przegubów manipulatora:

Funkcja została wywołana dla dwóch przegubów z poniższymi argumentami:

* Przegub 1

q\_p\_1 = 0;

dq\_p\_1 = 0.1;

q\_k\_1 = 1/2 \* pi;

dq\_k\_1 = 0.1;

Tk\_1 = 10;

Ta\_1 = 4;

Tb\_1 = 4;

dt\_1 = 0.01;

V\_1 = 0.2;

[q\_1,dq\_1,ddq\_1, t\_1] = Trajectory\_Generation(solved, q\_p\_1, dq\_p\_1, q\_k\_1, dq\_k\_1, Tk\_1, Ta\_1, Tb\_1, dt\_1, V\_1);

* Przegub 2

q\_p\_2 = 0;

dq\_p\_2 = 0.2;

q\_k\_2 = 3/2 \* pi;

dq\_k\_2 = 0.1;

Tk\_2 = 10;

Ta\_2 = 2;

Tb\_2 = 4;

dt\_2 = 0.01;

V\_2 = 0.5;

[q\_2,dq\_2,ddq\_2, t\_2] = Trajectory\_Generation(solved, q\_p\_2, dq\_p\_2, q\_k\_2, dq\_k\_2, Tk\_2, Ta\_2, Tb\_2, dt\_2, V\_2);

Uzyskane przebiegi pozycji, prędkości oraz przyspieszenia kątowego przegubów manipulatora:

Obraz zawierający wykres

Opis wygenerowany automatycznie

1. **Przebieg pozycji członu roboczego, przebieg orientacji członu roboczego, prędkość liniowa członu roboczego oraz prędkość kątowa członu roboczego dla wygenerowanej w pkt. 5 trajektorii.**

Obraz zawierający wykres

Opis wygenerowany automatycznie

1. **Skrypt źródłowy funkcji "inverse\_2DoF" napisany w programie Matlab.**

Następnie należało stworzyć funkcję ”inverse\_2DoF” realizującą odwrotne zadanie kinematyki dla naszego modelu manipulatora.

Funkcja:

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. **Parametry wejściowe funkcji „Trajectory\_Generation”.**

Jako parametry wejściowe funkcji „Trajectory\_Generation” zostały przyjęte dane identycznie jak w punkcie nr 5.

1. **Wykres wygenerowanej w pkt 8 trajektorii oraz wykresy pozycji kątowych przegubów wyznaczone poprzez rozwiązanie zadania odwrotnego.**

Obraz zawierający wykres

Opis wygenerowany automatycznie

Rozwiązanie zadania kinematyki odwrotnej:

Obraz zawierający wykres

Opis wygenerowany automatycznie

1. **Skrypt źródłowy programu głównego (gdzie są podane parametry i wywoływane są funkcje „Trajectory\_Generation”, "direct\_2DoF" oraz "inverse\_2DoF", i w którym znajdują się procedury wygenerowania wykresów).**

solved = Z;

q\_p\_1 = 0;

dq\_p\_1 = 0.1;

q\_k\_1 = 1/2 \* pi;

dq\_k\_1 = 0.1;

Tk\_1 = 10;

Ta\_1 = 4;

Tb\_1 = 4;

dt\_1 = 0.01;

V\_1 = 0.2;

[q\_1,dq\_1,ddq\_1, t\_1] = Trajectory\_Generation(solved, q\_p\_1, dq\_p\_1, q\_k\_1, dq\_k\_1, Tk\_1, Ta\_1, Tb\_1, dt\_1, V\_1);

solved = Z;

q\_p\_2 = 0;

dq\_p\_2 = 0.2;

q\_k\_2 = 3/2 \* pi;

dq\_k\_2 = 0.1;

Tk\_2 = 10;

Ta\_2 = 2;

Tb\_2 = 4;

dt\_2 = 0.01;

V\_2 = 0.5;

[q\_2,dq\_2,ddq\_2, t\_2] = Trajectory\_Generation(solved, q\_p\_2, dq\_p\_2, q\_k\_2, dq\_k\_2, Tk\_2, Ta\_2, Tb\_2, dt\_2, V\_2);

figure;

subplot(2, 3, 1);

plot(t\_1, q\_1, t\_1(1), q\_p\_1, 'o', Tk\_1, q\_k\_1, 'o')

legend('Pozycja', 'Pozycja początkowa', 'Pozycja końcowa', "Location","best")

title('Pozycja przegub 1')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

subplot(2, 3, 2);

plot(t\_1, dq\_1, t\_1(1), dq\_p\_1, 'o', Tk\_1, dq\_k\_1, 'o')

legend('Prędkość', 'Prędkość początkowa', 'Prędkość końcowa', "Location","best")

title('Prędkość przegub 1')

xlabel('t')

ylabel('Prędkość kątowa [rad/s]')

grid on

axis auto

subplot(2, 3, 3);

plot(t\_1, ddq\_1)

legend('Przyspieszenie', "Location","best")

title('Przyspieszenie przegub 1')

xlabel('t')

ylabel('Przyspieszenie kątowe [rad/s{^2}]')

grid on

axis auto

subplot(2, 3, 4);

plot(t\_2, q\_2, t\_2(1), q\_p\_2, 'o', Tk\_2, q\_k\_2, 'o')

legend('Pozycja', 'Pozycja początkowa', 'Pozycja końcowa', "Location","best")

title('Pozycja przegub 2')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

subplot(2, 3, 5);

plot(t\_2, dq\_2, t\_2(1), dq\_p\_2, 'o', Tk\_2, dq\_k\_2, 'o')

legend('Prędkość', 'Prędkość początkowa', 'Prędkość końcowa', "Location","best")

title('Prędkość przegub 2')

xlabel('t')

ylabel('Prędkość kątowa [rad/s]')

grid on

axis auto

subplot(2, 3, 6);

plot(t\_2, ddq\_2)

legend('Przyspieszenie', "Location","best")

title('Przyspieszenie przegub 2')

xlabel('t')

ylabel('Przyspieszenie kątowe [rad/s{^2}]')

grid on

axis auto

t = t\_1;

q = double([q\_1; q\_2]');

dq = double([dq\_1; dq\_2]');

L1 = 0.6;

L2 = 0.5;

[Pee, Psiee, Vee, Omee] = direct\_2DoF(q, dq, L1, L2);

figure

subplot(2, 2, 1)

plot(Pee(:, 1), Pee(:, 2))

title('Pozycja członu roboczego')

xlabel('x [m]')

ylabel('y [m]')

grid on

axis auto

subplot(2, 2, 2)

plot(t, Psiee)

title('Orientacja członu roboczego [rad]')

xlabel('t')

ylabel('Orientacja [rad]')

grid on

axis auto

subplot(2, 2, 3)

plot(Vee(:, 1), Vee(:, 2))

title('Prędkość liniową członu roboczego [m/s]')

xlabel('V{\_x} [m/s]')

ylabel('V{\_y} [m/s]')

grid on

axis auto

subplot(2, 2, 4)

plot(t, Omee)

title('Prędkość kątowa członu roboczego')

xlabel('t')

ylabel('Prędkość kątowa [rad/s]')

grid on

axis auto

[sol1, sol2] = inverse\_2DoF(Pee, L1, L2);

figure

subplot(5, 1, 1)

plot(Pee(:, 1), Pee(:, 2))

title('Pozycja członu roboczego')

xlabel('x [m]')

ylabel('y [m]')

grid on

axis auto

subplot(5, 1, 2)

plot(t, sol1(:, 1))

title('Pozycja kątowa przegubu 1 (rozwiązanie 1)')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

subplot(5, 1, 3)

plot(t, sol1(:, 2))

title('Pozycja kątowa przegubu 2 (rozwiązanie 1)')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

subplot(5, 1, 4)

plot(t, sol2(:, 1))

title('Pozycja kątowa przegubu 1 (rozwiązanie 2)')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

subplot(5, 1, 5)

plot(t, sol2(:, 2))

title('Pozycja kątowa przegubu 2 (rozwiązanie 2)')

xlabel('t')

ylabel('Pozycja kątowa [rad]')

grid on

axis auto

1. **Wnioski.**

Z wykresów wynika, że symulacja zadania kinematyki prostej oraz odwrotnej dla manipulatora o dwóch przegubach obrotowych zostało rozwiązane poprawnie. Zadanie proste kinematyki zamienia współrzędne złączowe robota na współrzędne kartezjańskie, natomiast zadanie odwrotne współrzędne kartezjańskie na współrzędne złączowe. Warto wspomnieć, że rozwiązanie zadania kinematyki odwrotnej nie jest jednoznaczne i możliwa jest wielokrotność takich rozwiązań co wynika z wykresów wykonanych dla zadania odwrotnego kinematyki.

W funkcji Trajectory\_Generation końcowa pozycja kątowa przegubu 1 wynosi 1/2 pi, zaś przegubu nr 2 3/2 pi czyli odpowiednio 90 oraz 270 stopni. W rozwiązaniu nr 2 zadania kinematyki odwrotnej pozycja kątowa przegubu nr 1 pokrywa się z wynikami uzyskanymi w funkcji „Trajectory\_Generation”., zaś przegub nr 2 został obrócony o kąt -1/2 pi co jest równoważne kątowi 3/2 pi. Końcówka robota osiąga dokładnie ten sam punkt we współrzędnych kartezjańskich lecz robi to w dwa różne sposoby. Rozwiązanie nr 1 zadania kinematyki odwrotnej przedstawia kolejny sposób, w którym robot osiąga zadaną pozycję.

Podsumowując zadanie kinematyki odwrotnej jest dużo trudniejsze od prostego zadania kinematyki ze względu wielokrotność rozwiązań. Powyższe ćwiczenie pozwoliło mi na zapoznanie się z symulacją oraz rozwiązywaniem zadania kinematyki prostej i odwrotnej dla manipulatora o dwóch przegubach obrotowych.