****

**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

Rysunek 1 Schemat manipulatora 2DoF z danymi o członach

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

Równanie 1 Wyprowadzone wzory na pozycję i orientację członu roboczego

Gdzie:

xee - pozycja członu roboczego na osi x,

yee - pozycja członu roboczego na osi y,

Ψee - orientacja członu roboczego,

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:

Równanie 2 Wzory na prędkości liniowe członu roboczego

Obraz zawierający kalendarz

Opis wygenerowany automatycznie

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

Równanie 3 Wyprowadzone wzory na prędkości liniowe oraz prędkość kątową członu roboczego

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

Kolejnym krokiem było napisanie funkcji direct\_2DoF, która realizowała zadanie proste 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

Listing 1 Skrypt źródłowy funkcji "direct\_2DoF"

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 Trajectory\_Generation została wywołana dla dwóch przegubów z poniższymi argumentami:

* Dane wejściowe dla przegubu nr 1

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Listing 2 Dane wejściowe funkcji Trajectory\_Generation dla przegubu nr 1

* Dane wejściowe dla przegubu nr 2

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Listing 3 Dane wejściowe funkcji Trajectory\_Generation dla przegubu nr 2

Uzyskane przebiegi pozycji, prędkości oraz przyspieszenia kątowego przegubów manipulatora z funkcji Trajectory\_Generation:

Obraz zawierający tekst, diagram, linia, Równolegle

Opis wygenerowany automatycznie

Rysunek 2 Wygenerowane przebiegi pozycji, prędkości i przyspieszenia kątowego przegubów

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

Na podstawie pozycji oraz prędkości kątowych przegubów zostały wygenerowane przebiegi pozycji, orientacji, prędkości liniowej oraz prędkości kątowej członu roboczego uzyskane z funkcji Direct\_2DoF.

Obraz zawierający tekst, diagram, linia, Wykres

Opis wygenerowany automatycznie

Rysunek 3 Wygenerowane przebiegi pozycji, orientacji, prędkości liniowej oraz prędkości kątowej członu roboczego

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

Kolejnym zadaniem było stworzenie funkcji ”inverse\_2DoF” realizującą odwrotne zadanie kinematyki dla naszego modelu manipulatora. Funkcja jako parametry przyjmowała pozycję członu roboczego wygenerowaną przy pomocy funkcji direct\_2DoF oraz długości członów manipulatora.

Skrypt źródłowy funkcji inverse\_2DoF:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Listing 4 Skrypt źródłowy funkcji inverse\_2DoF

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

Funkcja inverse\_2DoF została wywołana w następujący sposób:

1. Na początku zostały wygenerowane trajektorie pozycji kątowej oraz prędkości kątowej dla dwóch przegubów manipulatora. W tym celu została wywołana funkcja Trajectory\_Generation z poniższymi parametrami wejściowymi.

Dane wejściowe dla przegubu nr 1

q\_p\_1 = 0; % pozycja początkowa przegubu nr 1 [rad]

dq\_p\_1 = 0.0; % prędkość początkowa przegubu nr 1 [rad]

q\_k\_1 = 1/2 \* pi; % pozycja końcowa przegubu nr 1 [rad/s]

dq\_k\_1 = 0.0; % prędkość końcowa przegubu nr 1 [rad/s]

Tk\_1 = 10; % Czas trwania ruchu przegubu [s]

Ta\_1 = 4; % Czas trwania fazy przyspieszania [s]

Tb\_1 = 4; % Czas trwania fazy hamowania [s]

dt\_1 = 0.01; % Krok czasowy [s]

V\_1 = 0.2; % Prędkość przegubu w fazie drugiej [rad/s]

[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);

Dane wejściowe dla przegubu nr 2

q\_p\_2 = 0; % pozycja początkowa przegubu nr 1 [rad]

dq\_p\_2 = 0.0; % prędkość początkowa przegubu nr 1 [rad]

q\_k\_2 = 1/4 \* pi; % pozycja końcowa przegubu nr 1 [rad/s]

dq\_k\_2 = 0.0; % prędkość końcowa przegubu nr 1 [rad/s]

Tk\_2 = 10; % Czas trwania ruchu przegubu [s]

Ta\_2 = 2; % Czas trwania fazy przyspieszania [s]

Tb\_2 = 4; % Czas trwania fazy hamowania [s]

dt\_2 = 0.01; % Krok czasowy [s]

V\_2 = 0.1; % Prędkość przegubu w fazie drugiej [rad/s]

[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);

1. Następnie w celu uzyskania przebiegu pozycji członu roboczego manipulatora wywołana została funkcja direct\_2DoF w poniższy sposób:

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);

1. Ostatnim krokiem było wywołanie funkcji inverse\_2DoF dla przebiegu pozycji członu roboczego wygenerowanego przy pomocy funkcji direct\_2DoF w poprzednim kroku.

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

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

Obraz zawierający tekst, diagram, linia, Równolegle

Opis wygenerowany automatycznie

Rysunek 4 Wygenerowane przebiegi pozycji, prędkości i przyspieszenia kątowego przegubów manipulatora

Rozwiązanie zadania kinematyki odwrotnej:

Obraz zawierający tekst, diagram, Równolegle, linia

Opis wygenerowany automatycznie

Rysunek 5 Uzyskane pozycje kątowe przegubów z rozwiązania zadania kinematyki odwrotnej

Dodatkowo na powyższym wykresie w celach porównawczych pomarańczową linią zostały narysowane przebiegi pozycji kątowych przegubów wygenerowane z funkcji Trajectory\_Generation.

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; % pozycja początkowa przegubu nr 1 [rad]

dq\_p\_1 = 0.0; % prędkość początkowa przegubu nr 1 [rad]

q\_k\_1 = 1/2 \* pi; % pozycja końcowa przegubu nr 1 [rad/s]

dq\_k\_1 = 0.0; % prędkość końcowa przegubu nr 1 [rad/s]

Tk\_1 = 10; % Czas trwania ruchu przegubu [s]

Ta\_1 = 4; % Czas trwania fazy przyspieszania [s]

Tb\_1 = 4; % Czas trwania fazy hamowania [s]

dt\_1 = 0.01; % Krok czasowy [s]

V\_1 = 0.2; % Prędkość przegubu w fazie drugiej [rad/s]

[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; % pozycja początkowa przegubu nr 1 [rad]

dq\_p\_2 = 0.0; % prędkość początkowa przegubu nr 1 [rad]

q\_k\_2 = 1/4 \* pi; % pozycja końcowa przegubu nr 1 [rad/s]

dq\_k\_2 = 0.0; % prędkość końcowa przegubu nr 1 [rad/s]

Tk\_2 = 10; % Czas trwania ruchu przegubu [s]

Ta\_2 = 2; % Czas trwania fazy przyspieszania [s]

Tb\_2 = 4; % Czas trwania fazy hamowania [s]

dt\_2 = 0.01; % Krok czasowy [s]

V\_2 = 0.1; % Prędkość przegubu w fazie drugiej [rad/s]

[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 [s]'); 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 [s]')

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 [s]')

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 [s]'); 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 [s]')

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 [s]')

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); hold on; plot(t, sol1(:, 1)); plot(t, q\_1); hold off

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

legend('Rozwiązanie zadania odwrotnego', 'Pozycja przegubu z funkcji Trajectory Generation', 'Location','best')

xlabel('t [s]'); ylabel('Pozycja kątowa [rad]'); grid on; axis auto

subplot(5, 1, 3); hold on; plot(t, sol1(:, 2)); plot(t, q\_2); hold off

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

legend('Rozwiązanie zadania odwrotnego', 'Pozycja przegubu z funkcji Trajectory Generation', 'Location','best')

xlabel('t [s]'); ylabel('Pozycja kątowa [rad]'); grid on; axis auto

subplot(5, 1, 4); hold on; plot(t, sol2(:, 1)); plot(t, q\_1); hold off

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

legend('Rozwiązanie zadania odwrotnego', 'Pozycja przegubu z funkcji Trajectory Generation', 'Location','best')

xlabel('t [s]'); ylabel('Pozycja kątowa [rad]'); grid on; axis auto

subplot(5, 1, 5); hold on; plot(t, sol2(:, 2)); plot(t, q\_2); title('Pozycja kątowa przegubu 2 (rozwiązanie 2)')

legend('Rozwiązanie zadania odwrotnego', 'Pozycja przegubu z funkcji Trajectory Generation', 'Location','best')

xlabel('t [s]'); 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 końcówki członu roboczego, natomiast zadanie odwrotne współrzędne kartezjańskie końcówki członu roboczego na współrzędne złączowe.

W funkcji Trajectory\_Generation końcowa pozycja kątowa przegubu 1 wynosi 1/2 \* pi, zaś przegubu nr 2 1/4 \* pi czyli odpowiednio 90 oraz 45 stopni. W rozwiązaniu nr 1 zadania kinematyki odwrotnej pozycja kątowa przegubów pokrywają się z wynikami uzyskanymi w funkcji „Trajectory\_Generation”. Rozwiązanie nr 2 również zostało wyznaczone poprawnie, ponieważ pozycja kątowa przegubu nr 2 jest odbiciem wykresu z funkcji Trajectory\_Generation od osi X. Końcówka robota osiąga dokładnie ten sam punkt we współrzędnych kartezjańskich lecz robi to na dwa różne sposoby.

Podsumowując, zadanie kinematyki odwrotnej jest dużo trudniejsze od prostego zadania kinematyki, lecz pozwala na sprawdzenie poprawności wygenerowanego rozwiązania kinematyki prostej co jest bardzo ważne w przypadku obliczeń numerycznych. Powyższe ćwiczenie pozwoliło mi na zapoznanie się z symulacją oraz rozwiązywaniem zadania kinematyki prostej i odwrotnej dla manipulatora o dwóch przegubach obrotowych.