

Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

PRACA DYPLOMOWA INŻYNIERSKA

TYTUŁ PRACY:
Modelowanie ruchu robotów
poruszających się z poślizgiem

AUTOR:
Mateusz Strembicki

PROMOTOR:
dr inż. Katarzyna Zadarnowska

Spis treści

1	Wstęp	3
1.1	Motywacja	4
1.2	Cele pracy	4
1.3	Metodyka badawcza	4
2	Modelowanie	5
2.1	Model kinematyki	6
2.2	Model dynamiki	11
2.3	Badania symulacyjne modelu	15
2.3.1	Model nieholonomiczny (bez poślizgów)	16
2.3.1.1	Prędkości o profilu trapezoidalnym	17
2.3.1.2	Prędkości w postaci szeregu Fouriera	22
2.3.2	Model rzeczywisty (z poślizgiem)	27
2.3.2.1	Prędkości o profilu trapezoidalnym	27
2.3.2.2	Prędkości w postaci szeregu Fouriera	33
2.3.2.3	Duże momenty dla prędkości o profilu trapezoidalnym	38
3	Prawo sterowania	41
3.1	Integrator Brockett’a i postać normalna	42
3.2	Kinematyczne prawo sterowania w zapisie postaci normalnej	46
3.3	Model dynamiki w zapisie postaci normalnej	47
3.4	Dynamiczne prawo sterowania w zapisie postaci normalnej	48
3.5	Testy symulacyjne	51
3.5.1	Wpływ ϕ oraz δ_d na realizację ruchu	51
3.5.2	Zadanie stabilizacji	52
3.5.3	Zadanie śledzenia trajektorii	55
4	Podsumowanie	59
A	Modelowanie tarcia	61
B	Trapezoidalny profil prędkości	63
C	Szereg Fouriera	65
D	Metody Lapunowa	67
D.1	I metoda Lapunowa	67
D.2	II metoda Lapunowa	67
	Bibliografia	68

Rozdział 1

Wstęp

Często podczas projektowania robotów mobilnych zakłada się, że ich ruch odbywa się bez poślizgu, przez co roboty modelowane są jako układy nieholonomiczne, poruszające się w idealnym środowisku/po idealnej nawierzchni, z pominięciem zjawiska tarcia, buksowania czy hamowania kół [15, 13, 8]. Brak uwzględnienia poślizgu powodować może odstępstwa od zakładanych wyników, a modelowany układ może (i najczęściej będzie) się poruszać niezgodnie z rzeczywistością. Z tego powodu ważne jest, aby w równaniach kinematyki i dynamiki uwzględniać wpływ poślizgów na ruch robota, a co za tym idzie, by wciąż eksplorować tematykę układów poruszających się w obecności poślizgów.

Rozważanie tarcia w modelach ruchu to kluczowy krok w dążeniu do precyzyjnego odwzorowania rzeczywistości. Tarcie jest jednym z tych czynników, które mają ogromny wpływ na zachowanie się robotów. Jednak jego nieliniowy charakter sprawia, że modelowanie tego zjawiska może być wyzwaniem. Wiele modeli ruchu bez poślizgów pomija te niuanse z powodu ich złożoności. Dlatego uwzględnienie sił tarcia może stanowić kluczowy krok w kierunku konstruowania bardziej realistycznych symulacji.

Równie istotne są siły dysypatywne, choć czasem mogą być pomijane ze względu na ich pozornie mniejszy wpływ na model. Niemniej jednak, uwzględnienie tych sił może prowadzić do bardziej kompletnego odwzorowania ruchu robotów. Ostatecznie, model, który uwzględnia zarówno siły tarcia, jak i dysypatywne, zdaje się być krokiem w kierunku bardziej wiernego odzwierciedlenia rzeczywistości w symulacjach.

Co więcej, dla nieholonomicznych robotów mobilnych zdefiniowano szereg algorytmów sterowania, które pozwalają realizować różne zadania sterowania pojazdów. Najczęściej stosowanymi w robotyce zadaniami sterowania są:

1. P2P (*ang. point to point*) – stabilizacja – dojazd do zadanego punktu w przestrzeni,
2. śledzenie trajektorii – śledzenie krzywej parametryzowanej czasem,
3. śledzenie ścieżki – śledzenie krzywej parametryzowanej położeniem.

Często łączy się kilka metod sterowań, aby robot mógł realizować wiele zadań jednocześnie [12, 1, 19, 21]. Jeśli chodzi o roboty poruszające się w obecności poślizgów – liczba prac naukowych poświęconych tematyce projektowania algorytmów sterowania jest znacznie uboższa [23, 22, 20, 24, 5].

1.1 Motywacja

Pierwotnym motywem realizacji tematu pracy była chęć zaprojektowania modelu robota poruszającego się z poślizgiem oraz jego fizyczna budowa i oprogramowanie. Jednak po przeanalizowaniu dostępnej literatury stwierdzono, że temat jest bardzo szeroki i już samo zaproponowanie modelu ruchu robota jest zagadnieniem nietrywialnym i może stanowić duże wyzwanie. W związku z powyższym skupiono się na zaproponowaniu modelu czterokołowego robota mobilnego, u którego można wywołać ruch z poślizgiem.

1.2 Cele pracy

Celem pracy jest zaproponowanie modelu kinematyki oraz dynamiki robota mobilnego z napędem na cztery koła, który będzie się poruszał z poślizgiem poprzecznym kół. W szczególności, w ramach pracy autor spróbuje dokonać analizy wpływu poślizgu poprzecznego na możliwości skrętne platformy mobilnej. Możliwości ruchowe wybranej platformy mobilnej zostaną zaimplementowane oraz wizualizowane w wybranym środowisku symulacyjnym.

Dodatkowo, opracowany zostanie algorytm sterowania, który będzie realizować zadanie stabilizacji w punkcie oraz śledzenia trajektorii. Choć to nie było pierwotnym założeniem pracy, autor uważa, że ten punkt jest na tyle ciekawy, że warto podjąć wysiłek w celu jego realizacji.

1.3 Metodyka badawcza

Podczas modelowania ruchu robota wzorowano się na pracy [10]. Ponadto, skorzystano z [4], w celu wprowadzenia poślizgu do równania kinematyki. Podczas definiowania algorytmu sterowania wzorowano się na podejściu postaci normalnej zaproponowanym dla robotów nieholonomicznych w pracach [7, 6] i rozwiniętym w [10] dla robotów poruszających się z poślizgami. Przeprowadzenie badań modelu ruchu i algorytmów sterowania opiera się na zaimplementowaniu i przeprowadzeniu szeregu symulacji modelu ruchu robota nieholonomicznego oraz modelu rzeczywistego (ze składowymi odpowiadającymi za poślizg). Oba modele ruchu porównano pod kątem sprawdzenia

- wpływu poślizgu na trajektorię robota,
- wpływu współczynników tarcia na zachowanie modelu.

Przeprowadzenie badań symulacyjnych w programie **MATLAB** pokaże jak robot, który porusza się z poślizgiem, radzi sobie z zadaniami stabilizacji w punkcie, czy śledzeniem trajektorii. Dodatkowo, w celach wizualizacji zrealizowano animację ruchu robota.

Rozdział 2

Modelowanie

W celu wyprowadzenia modelu matematycznego możliwie najlepiej odzwierciedlającego rzeczywiste zachowanie robota, należy uwzględnić możliwie najwięcej zmiennych wpływających na charakter zachowania robota. Wiadomym jest jednak, że nie da się wykonać modelu, który stuprocentowo będzie oddawał rzeczywiste zachowanie robota, w związku z czym stosuje się szereg zaokrągleń i przybliżeń, które wynikają chociażby z nieliniowości opisów matematycznych, czy braku wiedzy w kwestiach rozwiązywania niektórych zagadnień robotyczno-matematycznych.

Modelując rozważanego w pracy robota wykorzystano dwa rodzaje modeli: kinematyczny i dynamiczny. Pierwszy opisuje ruch robota, a drugi siły oddziaływające na niego. Na podstawie tych dwóch modeli można uzyskać dobre przybliżenie rzeczywistego zachowania robota w terenie.

Wykonanie szeregu symulacji modelu matematycznego pozwala na:

- zaobserwowanie zachowania robota przy określonych warunkach zewnętrznych,
- porównanie trajektorii ruchu/zachowania robota z intuicjami,
- poznanie wpływu wartości parametrów zewnętrznych na ruch robota,
- określenie możliwości doboru współczynników wewnętrznych robota,
- określenie minimalnych oraz maksymalnych parametrów pracy robota,
- porównanie wyników symulacyjnych z uzyskanymi z faktycznego zachowania robota.

Podsumowując, badania symulacyjne to proces testowania modelu, który pomaga dobierać parametry i ulepszać model w kolejnych etapach w taki sposób, by potencjalny prototyp nie został uszkodzony przez np. błędnie dobrane sterowanie na silnikach, co mogłoby skutkować ewentualnym uszkodzeniem elementów mechanicznych.

2.1 Model kinematyki

Rozważać będziemy model kinematyki robota przedstawionego na rysunku 2.1. Robot będzie opisywany we współrzędnych zewnętrznych (X_g, Y_g, Z_g) . Początek układu współrzędnych lokalnych przypisano do środka masy robota i oznaczono jako (x_l, y_l, z_l) . Współrzędne środka masy w układzie globalnym zapisano jako $COM = (X, Y, Z)$.

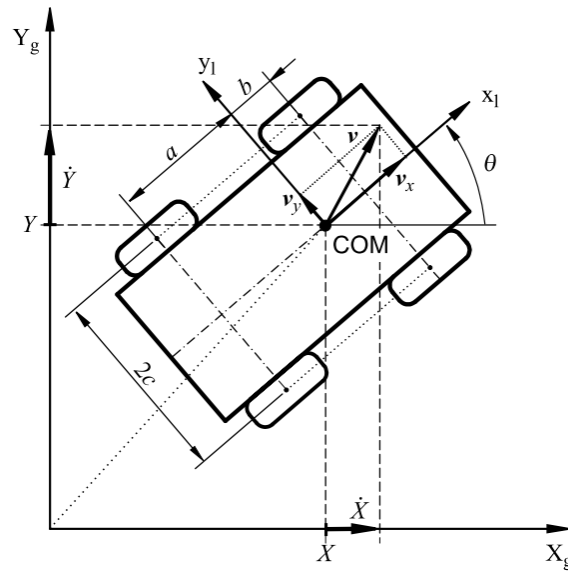
Model kinematyki robota mobilnego oddaje jego zachowanie podczas ruchu na płaszczyźnie w zależności od zadanego sterowania mającego sens prędkości. Uwzględniać będziemy zarówno model idealny (bez poślizgów) oraz model uwzględniający istnienie poślizgów. Konfiguracją q , będziemy oznaczać wektor zawierający informacje o położeniu robota w przestrzeni XY oraz orientacji θ określającej orientację robota względem globalnego układu współrzędnych

$$q = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix}. \quad (2.1)$$

Założono, że robot nie potrafi poruszać się wzdłuż wektora Z , więc nie uwzględnia się go w naszym wektorze stanu. Dzięki temu zabiegowi model opisany jest w przestrzeni dwuwymiarowej, co ułatwia obliczenia.

Zakładając, że robot porusza się po płaszczyźnie ze stałymi prędkościami liniowymi (v_x, v_y) wyrażonymi w lokalnym układzie współrzędnych jako $[v_x, v_y, 0]$ oraz obraca się zgodnie z wektorem prędkości kątowej wokół osi Z_l $[0, 0, \omega]$, każdorazową zmianę położenia można opisać poprzez różniczkę wektora stanu (wzór (2.1))

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix}. \quad (2.2)$$



Rysunek 2.1: Schemat robota ([10, str. 2])

Mając na uwadze rysunek 2.1 oraz wzór wektora zmian stanu (2.2) można napisać równanie

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad (2.3)$$

z którego jasno wynikają zależności pomiędzy zmianą położenia na płaszczyźnie XY a prędkościami lokalnymi robota (v_x, v_y) . Co więcej, ze względu na fakt, iż robot porusza się po powierzchni ekwipotencjalnej można zapisać, że

$$\dot{\theta} = \omega. \quad (2.4)$$

Uwzględniając wzory (2.3) oraz (2.4) zauważyć można, że wektor zmian stanu (wektor prędkości uogólnionych) (2.2) można przedstawić w postaci

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}. \quad (2.5)$$

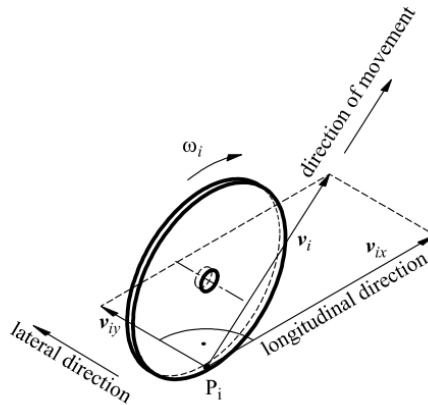
W przypadku idealnym zakłada się, że ruch robota nie podlega poślizgom, a co za tym idzie – m.in. poślizg poprzeczny jest pomijany. Oznacza to, że spełnione jest równanie nieholonomiczne zapisane przy pomocy macierzy Pfaffa

$$\begin{bmatrix} -\sin \theta & \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = 0. \quad (2.6)$$

Wzór (2.6) opisuje ograniczenie na poślizg poprzeczny, które w dalszej części, jako, że będziemy rozważać ruch rzeczywistego robota, zostanie osłabione poprzez uwzględnienie możliwości wystąpienia poślizgu.

W celu określenia prędkości v_x z jaką porusza się robot, posłużono się wzorem opracowanym przez Pacejkę, który opisuje zależność prędkości wzdłużnej v_{xi} koła i -tego, wyrażonej w lokalnym układzie współrzędnych, od promienia tocznego koła r_i (w omawianym przypadku jednakowe dla każdego koła) oraz jego prędkości kątowej ω_i [14]

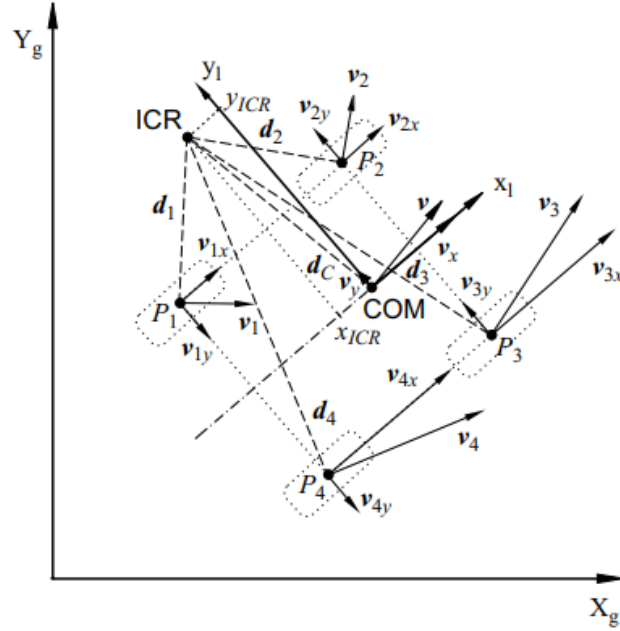
$$v_{xi} = r_i \omega_i. \quad (2.7)$$



Rysunek 2.2: Rozkład prędkości dla i -tego koła (Kozłowski, Pazderski [10, str. 3])

W pracy zakłada się, że grubość kół jest pomijalna, a koła stykają się z podłożem punktowo (rysunek 2.2), gdzie

- v_{ix} – składowa prędkości w osi OX ,
- v_{iy} – składowa prędkości w osi OY ,
- v_i – prędkość wypadkowa,
- P_i – punkt styku koła z powierzchnią,
- ω_i – prędkość kątowna koła.



Rysunek 2.3: Rozkład prędkości w robocie (Kozłowski [10, str. 3])

W celu opracowania modelu kinematycznego robota wymagane jest uwzględnienie wszystkich kół robota. Konieczne jest również zdefiniowanie chwilowego, centralnego punktu obrotu ICR (ang.: Instantaneous Center of Rotation) (rys. 2.3), który w danym momencie stanowi oś obrotu dla obracającego się obiektu [10]. Jest to punkt, wokół którego odbywa się obrót wywołany poślizgiem kół

$$ICR = (x_{ICR}, y_{ICR}) = (-d_{Cx}, -d_{Cy}). \quad (2.8)$$

W momencie obrotu prędkość tego punktu jest zerowa. Odległości d_{Cx} oraz d_{Cy} definiowane są wzorem

$$\frac{\|v_i\|}{\|d_i\|} = \frac{\|v\|}{\|d_C\|} = |\omega|, \quad (2.9)$$

gdzie $\|\cdot\|$ oznacza normę Euklidesową, a d_i to współrzędne położenia kół względem punktu ICR : $d_i = [d_x, d_y]^T$, $d_C = [d_{Cx}, d_{Cy}]^T$ (rysunek 2.3).

Wzór (2.10) jest rozwinięciem wzoru (2.9)

$$\frac{v_{ix}}{-d_{iy}} = \frac{v_x}{-d_{Cy}} = \frac{v_{iy}}{d_{ix}} = \frac{v_y}{d_{Cx}} = \omega. \quad (2.10)$$

Złożenie wzorów (2.8) oraz (2.10) pozwala na wyznaczenie współrzędnych punktu ICR

$$\frac{v_x}{y_{ICR}} = -\frac{v_y}{x_{ICR}} = \omega. \quad (2.11)$$

Ze wzoru (2.10) oraz rysunku 2.3 wynikają zależności

$$\begin{aligned} d_{1y} &= d_{2y} = d_{Cy} + c \\ d_{3y} &= d_{4y} = d_{Cy} - c \\ d_{1x} &= d_{4x} = d_{Cx} - a \\ d_{2x} &= d_{3x} = d_{Cx} + b, \end{aligned} \quad (2.12)$$

przy czym a, b, c są parametrami geometrycznymi robota (tabela 2.1). Zgodnie z ilustracją 2.3 i zależnościami (2.10) oraz (2.12) zapisano również nowe prędkości opisujące ruch robota

$$\begin{aligned} v_L &= v_{1x} = v_{2x} \\ v_R &= v_{3x} = v_{4x} \\ v_F &= v_{2y} = v_{3y} \\ v_B &= v_{1y} = v_{4y}. \end{aligned} \quad (2.13)$$

Korzystając ze wzorów (2.9)–(2.13) wyznaczono szereg równań służących do wyliczenia zależności pomiędzy prędkościami kół a prędkościami (postępową i obrotową) robota

$$\begin{bmatrix} v_L \\ v_R \\ v_F \\ v_B \end{bmatrix} = \begin{bmatrix} 1 & -c \\ 1 & c \\ 0 & -x_{ICR} + b \\ 0 & -x_{ICR} - a \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix}. \quad (2.14)$$

Zakładając, że każde koło jest tego samego rozmiaru uproszczono równanie służące do wyliczenia prędkości obrotowej ω robota

$$\omega_w = \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = \frac{1}{r} \begin{bmatrix} v_L \\ v_R \end{bmatrix}. \quad (2.15)$$

Składając wzory (2.14) oraz (2.15) otrzymuje się równanie na sterowanie robota

$$\begin{aligned} \begin{bmatrix} v_L \\ v_R \end{bmatrix} &= \begin{bmatrix} 1 & -c \\ 1 & c \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \\ \begin{bmatrix} v_x \\ \omega \end{bmatrix} &= \begin{bmatrix} 1 & -c \\ 1 & c \end{bmatrix}^{-1} \begin{bmatrix} r\omega_L \\ r\omega_R \end{bmatrix} \\ \begin{bmatrix} v_x \\ \omega \end{bmatrix} &= \frac{r}{2c} \begin{bmatrix} c & c \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \\ \eta = \begin{bmatrix} v_x \\ \omega \end{bmatrix} &= r \begin{bmatrix} \frac{\omega_L + \omega_R}{2} \\ \frac{-\omega_L + \omega_R}{2c} \end{bmatrix}. \end{aligned} \quad (2.16)$$

Wzór (2.16) pokazuje, że robot na poziomie kinematyki sterowany jest poprzez zmianę prędkości postępowej v_x oraz zmianę prędkości obrotowej ω .

Pomimo tego, że sterowanie odbywa się poprzez prędkość wzdłużną v_x , nie można zapominać o prędkości poprzecznej v_y . Posłuży ona do wprowadzenia do równań kinematyki

ostatniej składowej jaką jest x_{ICR} . Mianowicie, korzystając ze wzoru (2.11) można wy-
prowadzić równanie

$$v_y + x_{ICR}\dot{\theta} = 0, \quad (2.17)$$

które jednoznacznie uzupełni równanie Pfaff'a (wzór (2.6)) o warunek na istnienie poślizgu. Powyższy wzór nie jest całkowalny, więc opisuje nieholonomiczne ograniczenie, które można zapisać w postaci macierzy Pfaffa

$$\begin{bmatrix} -\sin\theta & \cos\theta & x_{ICR} \end{bmatrix} \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T = A(q)\dot{q} = 0. \quad (2.18)$$

Dzięki tej operacji, na bazie równania (2.18) możliwe jest określenie generatorów (pól wektorowych opisujących kierunki ruchu) robota

$$S^T(q)A^T(q) = 0, \quad (2.19)$$

gdzie

$$S(q) = \begin{bmatrix} \cos\theta & x_{ICR}\sin\theta \\ \sin\theta & -x_{ICR}\cos\theta \\ 0 & 1 \end{bmatrix}. \quad (2.20)$$

Wzór (2.20) określa macierz kinematyki dla badanego modelu. Sam model ma postać

$$\dot{q} = S(q)\eta, \quad (2.21)$$

co pokazuje, że aby poznać położenie robota w przestrzeni należy rozwiązać równanie różniczkowe pierwszego rzędu.

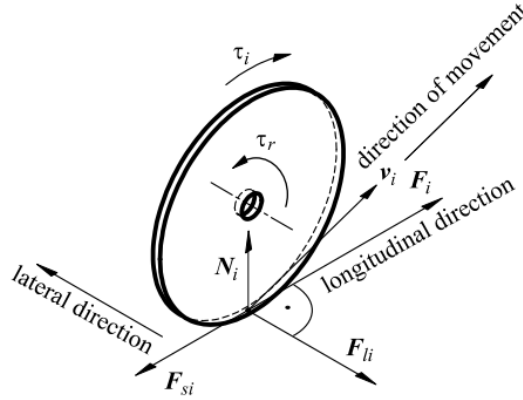
Podsumowując, model kinematyczny pozwala określić położenie robota na płaszczyźnie ekwipotencjalnej. Sterowany jest za pomocą dwóch zmiennych:

- v_x – prędkości liniowa/postępowa robota wzdłuż osi X ,
- ω – prędkości kątowa robota,

które bezpośrednio wpływają na zachowanie obiektu na płaszczyźnie.

2.2 Model dynamiki

Model dynamiki definiuje reakcję na zmianę sił podczas ruchu robota poruszającego się na płaszczyźnie. Właśnie ze względu na te różnice sił w czasie, do określenia modelu dynamiki potrzebny jest model kinematyki. Badany robot posiada napęd na cztery koła co oznacza, że na każde koło można zadać inne sterowanie τ . Każde takie nadanie siły kołu równoznaczne jest z wygenerowaniem sił reakcji działających na dane koło (rys. 2.4),



Rysunek 2.4: Rozkład sił dla i-tego koła (Kozłowski, Pazderski [10, str. 4])

gdzie

- v_i – prędkość wypadkowa,
- F_{si} – siła tarcia dla poślizgu wzdłużnego,
- F_{li} – siła tarcia dla poślizgu poprzecznego,
- F_i – siła wypadkowa,
- N_i – siła reakcji na nacisk na powierzchnię,
- τ_i – moment obrotowy koła.

Siłę wypadkową, która powoduje, że robot wprawiany jest w ruch, można zapisać również jako iloraz momentu obrotowego τ oraz promienia koła r jak we wzorze (2.22)

$$F_i = \frac{\tau_i}{r}. \quad (2.22)$$

Widać, że siła wypadkowa jest wprost proporcjonalna tylko do zadanego momentu obrotowego τ , ponieważ promień koła jest wartością stałą.

Najtrudniejszym zadaniem jest modelowanie sił tarcia ze względu na ich nieliniowy charakter. Zabieg opisujący modelowanie takich sił dla badanego robota opisano w dodatku A. Tutaj przytoczone zostaną jedynie wzory, odpowiednio, na siłę tarcia poprzecznego (równanie (A.3)) oraz wzdłużnego (równanie (A.4)),

$$F_{li} = \mu_{li} m g \widehat{sgn}(v_{yi}),$$

$$F_{si} = \mu_{si} m g \widehat{sgn}(v_{xi}),$$

gdzie μ_{li} oznacza współczynnik tarcia dla koła w kierunku poprzecznym, a μ_{si} – w kierunku wzdłużnym, natomiast g to przyspieszenie ziemskie (wartość przyjętą do obliczeń podano w tabeli 2.1). Warto nakreślić, że siła oporu F_{li} wynika z faktu, iż robot porusza się z poślizgiem i możliwe jest wystąpienie znacznego tarcia w kierunku poprzecznym. Jest to reakcja na poruszanie się pojazdu bokiem (robot nie posiada skrętnych kół, które mogłyby powodować zmniejszenie wartości siły działającej w tym kierunku). Siła F_{si} wynika natomiast oporu jaki należy pokonać, aby pojazd wprawiony został w ruch (lub, żeby wystąpiło buksowanie kół poprzez podanie zbyt dużego momentu obrotowego na koło). Jest to bezpośrednia reakcja na zadanie wartości momentu obrotowego na koło.

W celu wyznaczenia macierzy bezwładności robota dobrze jest posłużyć się równaniem Eulera – Langrange’a. Jako, że model porusza się po płaszczyźnie ekwipotencjalnej, uwzględniona zostanie jedynie energia kinetyczna robota. Lagrangian $L(q, \dot{q})$ uwzględniający własności energetyczne modelu przyjmie postać

$$L(q, \dot{q}) = \frac{1}{2} m v^T v + \frac{1}{2} I \omega^2, \quad (2.23)$$

gdzie m określa masę robota, a I jego moment bezwładności. Zauważyć można, że w równaniu pojawia się forma kwadratowa, która po rozpisaniu $v^T v = v_x^2 + v_y^2 = \dot{X} + \dot{Y}$, pozwala na zapisanie wzoru (2.23), w połączeniu ze wzorem (2.4), w czytelnej postaci

$$L(q, \dot{q}) = \frac{1}{2} m (\dot{X}^2 + \dot{Y}^2) + \frac{1}{2} I \dot{\theta}^2. \quad (2.24)$$

Różniczkując równanie (2.24) można uzyskać postać macierzy bezwładności

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \begin{bmatrix} m \ddot{X} \\ m \ddot{Y} \\ I \ddot{\theta} \end{bmatrix} = M \ddot{q}, \quad (2.25)$$

skąd otrzymano postać macierzy bezwładności

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}. \quad (2.26)$$

Dalej, warto zdefiniować macierz sił tarcia oraz macierz sił dysypatywnych, czyli takich, które są niewielkie jednak powodują, że model staje się możliwie najbardziej przybliżony do rzeczywistości. Takimi siłami będą siły oddziałujące na koła w kierunkach poprzecznym F_{li} i wzdłużnym F_{si} , natomiast siłami dysypatywnymi będzie pływanie momentu sił wokół środka masy M_r

$$F_{rx}(\dot{q}) = \cos \theta \sum_{i=1}^4 F_{si}(v_{xi}) - \sin \theta \sum_{i=1}^4 F_{li}(v_{yi}) \quad (2.27)$$

$$F_{ry}(\dot{q}) = \sin \theta \sum_{i=1}^4 F_{si}(v_{xi}) + \cos \theta \sum_{i=1}^4 F_{li}(v_{yi}) \quad (2.28)$$

$$M_r(\dot{q}) = -a \sum_{i=1,4} F_{li}(v_{yi}) + b \sum_{i=2,3} F_{li}(v_{yi}) + c \left[- \sum_{i=1,2} F_{si}(v_{xi}) + \sum_{i=3,4} F_{si}(v_{xi}) \right]. \quad (2.29)$$

Określenie wektora $R(\dot{q})$ definiującego uogólnione siły oporu polega na wstawieniu sił ze wzorów (2.27), (2.28) i (2.29) w jego odpowiednie komórki

$$R(\dot{q}) = \begin{bmatrix} F_{rx}(\dot{q}) & F_{ry}(\dot{q}) & M_r(\dot{q}) \end{bmatrix}^T. \quad (2.30)$$

Kolejnym krokiem jest zebranie wyrażanych w układzie inercyjnym sił generowanych przez napędy i wpływających na ruch robota

$$F_x = \cos(\theta) \sum_{i=1}^4 F_i,$$

$$F_y = \sin(\theta) \sum_{i=1}^4 F_i$$

oraz momentów obrotowych wokół środka masy

$$M = c(-F_1 - F_2 + F_3 + F_4)$$

w jeden wektor, w celu wyznaczenia macierzy $B(q)$, która jest macierzą przejścia pozwalającą określić wpływ sterowania na model. A zatem po uwzględnieniu wzoru (2.22) otrzymano wektor sił

$$F = \frac{1}{r} \begin{bmatrix} \cos(\theta) \sum_{i=1}^4 \tau_i \\ \sin(\theta) \sum_{i=1}^4 \tau_i \\ c(-\tau_1 - \tau_2 + \tau_3 + \tau_4) \end{bmatrix} = B(q)\tau. \quad (2.31)$$

W tym miejscu warto jest zdefiniować nowe sterowanie τ dla modelu, które zapisać można jako wektor sum odpowiednich sterowań zdefiniowany dla kół lewych i prawych robota

$$\tau = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} = \begin{bmatrix} \tau_1 + \tau_2 \\ \tau_3 + \tau_4 \end{bmatrix}, \quad (2.32)$$

gdzie τ_L oraz τ_R odpowiadają za sterowanie na koła lewe i prawe.

Przekształcając równanie (2.31) do postaci $F\tau^{-1} = B(q)$ można wyliczyć postać macierzy sterowań $B(q)$

$$B(q) = \frac{1}{r} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ -c & c \end{bmatrix}. \quad (2.33)$$

Używając macierzy z równań (2.26), (2.30), (2.33) i (2.32) zdefiniować można model dynamiki robota

$$M(q)\ddot{q} + R(\dot{q}) = B(q)\tau. \quad (2.34)$$

Równanie (2.34) nie uwzględnia nadanych ograniczeń nieholonomicznych wynikających z zależności (2.18). Powołując się na pracę Caracciolo wprowadza się do równania siły zależne od mnożników Lagrange'a mające zapewnić spełnienie ograniczeń nieholonomicznych nałożonych na układ [4]

$$M(q)\ddot{q} + R(\dot{q}) = B(q)\tau + A^T(q)\lambda. \quad (2.35)$$

W dalszym kroku spróbujemy zapisać równania (2.35) w kategoriach prędkości pomocniczych η . W tym celu należy przemnożyć lewostronnie powyższe równanie przez $S^T(q)$ oraz wykorzystać wzór (2.19)

$$S^T(q)M(q)\ddot{q} + S^T(q)R(\dot{q}) = S^T(q)B(q)\tau + S^T(q)A^T(q)\lambda. \quad (2.36)$$

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta}. \quad (2.37)$$

Ingerując we wzór (2.36) poprzez równanie (2.37) otrzymuje się końcowy model ruchu badanego modelu

$$\bar{M}\dot{\eta} + \bar{C}\eta + \bar{R} = \bar{B}\tau, \quad (2.38)$$

przy czym

- macierz sił bezwładności \bar{M}

$$\bar{M} = S^T M S = \begin{bmatrix} m & 0 \\ 0 & mx_{ICR}^2 + I \end{bmatrix}, \quad (2.39)$$

- macierz sił Coriolisa \bar{C}

$$\bar{C} = S^T M \dot{S} = mx_{ICR} \begin{bmatrix} 0 & \dot{\theta} \\ -\dot{\theta} & \dot{x}_{ICR} \end{bmatrix}. \quad (2.40)$$

- macierz sił tarcia i dysypatywnych \bar{R}

$$\bar{R} = S^T R = \begin{bmatrix} F_{rx}(\dot{q}) \\ x_{ICR}F_{ry}(\dot{q}) + M_r(\dot{q}) \end{bmatrix}, \quad (2.41)$$

- macierz sterowań \bar{B}

$$\bar{B} = S^T B = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ -c & c \end{bmatrix}. \quad (2.42)$$

Patrząc na wzór (2.38) widać, że otrzymany model jest kaskadowy, czyli na obiekt wpłynąć można poprzez sterowanie τ , ale tylko pośrednio. Bezpośrednio na robota oddziałuje sterowanie η , które poprzez prędkości pomocnicze wpływa na konfigurację q modelu.

Podsumowując, w powyższym podrozdziale opisano siły oddziałujące na model, które pozwoliły na określenie macierzy sił dysypatywnych (wzór (2.30)). Kolejno, zastosowano równania Euler'a-Lagrange'a do wyznaczenia postaci macierzy bezwładności robota (wzór (2.26)). Wyprowadzono macierz sterowań $B(q)$, pozwalającą na przeniesienie sterowań na prędkości w modelu (wzór (2.33)). Następnie w równaniach dynamiki uwzględniono ograniczenia nieholonomiczne określone w modelu kinematyki (równanie (2.35)). W efekcie uzyskano końcową postać równań (2.38) opisujących wpływ sterowań na prędkości pomocnicze, które dalej wpływają na konfigurację robota.

2.3 Badania symulacyjne modelu

Testy symulacyjne służą do sprawdzenia zachowania się modelu w różnych warunkach początkowych, dla różnych wartości sterowań itp.. Dodatkowo, badając modele opisujące zachowanie układu w rzeczywistych warunkach, przeprowadza się jednocześnie badania na modelu działającym w warunkach idealnych. W szczególności, w celu testowania takiego idealnego (nieholonomicznego) modelu należy przyjąć, że zmienna x_{ICR} jest równa zero. Aby badać model idealny należy również pozbyć się tarcia z modelu dynamiki (w celu wytworzenia warunków idealnych), a zatem macierz $R(q)$, ze wzoru (2.30) również musi zostać wyzerowana.

Wartości parametrów robota oraz sterownika zaczerpnięto z pracy [10, str. 14] i umieszczono w tabeli 2.1.

Tabela 2.1: Tabela zbiorcza wartości parametrów

Parametry geometryczne robota	$a = 39 \text{ mm}$
	$b = 39 \text{ mm}$
	$c = 34 \text{ mm}$
	$r = 26,5 \text{ mm}$
Parametry dynamiczne robota	$m = 1 \text{ kg}$
	$I = 0,0036 \text{ kg} \cdot \text{m}^2$
Współczynniki tarcia	$\mu_l \in <0,2 ; 0,8>$
	$\mu_s \in <0,02 ; 0,18>$
Parametry kontrolera (prawo sterowania)	$k_1 = 1$
	$k_2 = 1$
	$k_3 = 5$
	$k_4 = 2$
	$\alpha_0 = 3.2$
	$\alpha_1 = 0,5$
	$\epsilon_1 = 0,01$
	$\epsilon_2 = 0,02$
	$x_{ICR} = -15 \text{ mm}$
	$\rho = 2$
	$\mu_{l0} = 0,5$
	$\mu_{s0} = 0,1$
	$m_0 = 1,2 \text{ kg}$
	$I_0 = 0,0054 \text{ kg} \cdot \text{m}^2$

Ważne jest, aby testy obu modeli były przeprowadzone dla takich samych warunków początkowych, gdyż pozwoli to na faktyczne zbadanie wpływu poślizgu na wykonany ruch czy pokonaną trajektorię robota.

Warunki początkowe, dla których wykonano testy zebrane zostały w wektorze $W(q, \eta, \tau) = (x, y, \theta, \eta_1, \eta_2, wzm_{\tau_1}, wzm_{\tau_2}, wzm_{\tau_3}, wzm_{\tau_4})$, gdzie wzm_{τ_i} to wartość modulująca sygnał wejściowy (moment) dla koła i -tego. Każda symulacja przeprowadzona dla specyficznych warunków początkowych została oznaczona odpowiednim indeksem. Czas trwania symulacji T wynosi 10 s.

$$W_1(q, \eta, \tau) = [5 \ 2 \ 60^\circ \ 0 \ 0 \ 0,01 \ 0 \ 0,01 \ 0,01]^T \quad (2.43)$$

$$W_2(q, \eta, \tau) = [5 \ 2 \ 60^\circ \ 0 \ 0 \ -0,01 \ 0 \ -0,01 \ 0,01]^T \quad (2.44)$$

$$W_3(q, \eta, \tau) = [0 \ 0 \ 0^\circ \ 0 \ 0 \ 0,01 \ 0,01 \ 0 \ -0,03]^T \quad (2.45)$$

$$W_4(q, \eta, \tau) = [2 \ -3 \ 45^\circ \ 0 \ 0 \ 0,01 \ 0 \ 0,01 \ 0]^T \quad (2.46)$$

Powyższe warunki pozwalają zaprezentować zachowanie robota podczas jazdy po łuku (W_1), po prostej (W_4), obracania się wokół własnej osi (W_3), obracania się wokół punktu kontaktu kół prawych z podłożem (W_2).

2.3.1 Model nieholonomiczny (bez poślizgów)

W pierwszej kolejności przebadano model idealny. W tym celu przyjęto, że macierz $R(q)$ (wzór (2.30)) oraz zmienna x_{ICR} wynoszą zero. Po dokonaniu tych prostych operacji model dynamiki uprościł się do postaci

$$\bar{M}\dot{\eta} + \bar{C}\eta = \bar{B}\tau,$$

a macierz generatorów ruchu $S(q)$ (wzór (2.20))

$$S(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}.$$

Przy takich założeniach badany model to zwykły robot nieholonomiczny typu monocykl.

2.3.1.1 Prędkości o profilu trapezoidalnym

W pierwszym kroku sterowaniem dla obu rodzajów robotów będą momenty generujące prędkości o profilu trapezoidalnym. Takie testy pokażą również jak bardzo, i czy w ogóle, poślizg wpływa na zniekształcenie profilu prędkości.

Szczegółowe omówienie rozpatrywanego charakteru sterowań znajduje się w dodatku B, gdzie wyjaśnione jest jak zadawać sterowanie, aby uzyskać trapezoidalny profil prędkości oraz jakie są jego charakterystyczne cechy.

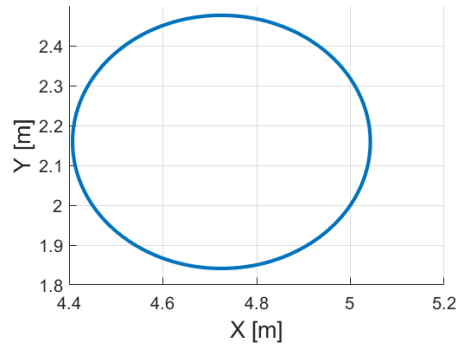
Warunki początkowe W_1 , W_2 oraz W_3 . Wykresy 2.5a, 2.7a i 2.9a pokazują, że przy większym wpływie sterowania jednej ze stron robota, obiekt porusza się po łuku, zakreślając okrąg. Rysunki traektorii $X(t)$ (2.5b – 2.9b) oraz $Y(t)$ (2.5c – 2.9c) zawierają informację dotyczącą liczby okręgów pokonanych przez robota w czasie T – w tym przypadku 9 pełnych okręgów. Przy największej prędkości czas trwania przejazdu wzdłuż jednego okręgu jest najkrótszy. Równanie (2.47) opisuje ruch po okręgu w przypadku modelu idealnego

$$\begin{cases} X = k_x \cos t \\ Y = k_y \sin t \end{cases}, \quad (2.47)$$

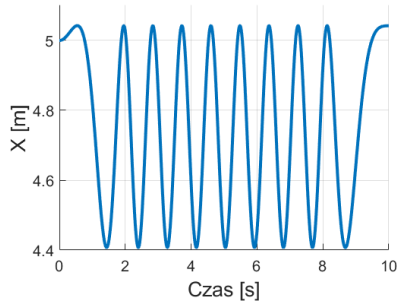
gdzie k_x oraz k_y określają jak duże jest wzmocnienie sygnału. Na podstawie wykresów $\eta(t)$ oraz $\tau(t)$ wnioskujemy, że zastosowany został trapezoidalny profil prędkości do sterowania pojazdem, a rysunki 2.6d – 2.10d pokazują, że ruch odbywał się po okręgu. Świadczy o tym wystąpienie składowej prędkości kątowej ω na wykresach prędkości.

Warunek początkowy W_4 . Tutaj przebadano zachowanie modelu przy sterowaniu, które zapewni ruch wzdłuż prostej (patrz rys. 2.11a). Dzięki użyciu trapezoidalnego profilu prędkości, na wykresach 2.11b oraz 2.11c pojawia się łagodny powolny start i powolne hamowanie modelu w strefach liniowego wzrostu, bądź też spadku prędkości $\eta(t)$ 2.12c, który wygenerowany został w oparciu o całkowanie krzywej $\tau(t)$ z rysunku 2.12d. Wykres 2.12a pokazuje, że robot odbywa się ze stałą początkową orientacją. Skoro ruch nie odbywa się po okręgu, to nie występuje tutaj składowa prędkości kątowej ω . Jest to zgodne z intuicją (rysunek 2.12c).

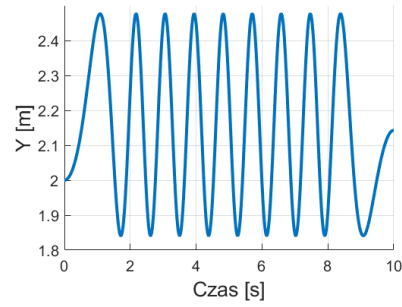
We wszystkich przypadkach (2.6b, 2.8b, 2.10b, 2.12b) wartość x_{ICR} wynosiła zero zgodnie z założeniami przytoczonymi w 2.3.1.



(a) Wykres trajektorii robota na płaszczyźnie XY

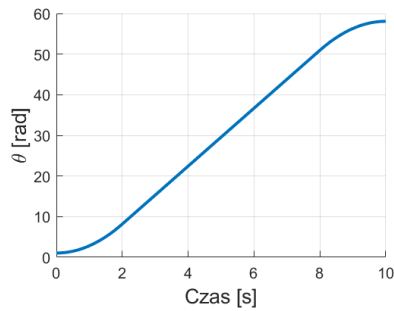


(b) Trajektoria $X(t)$

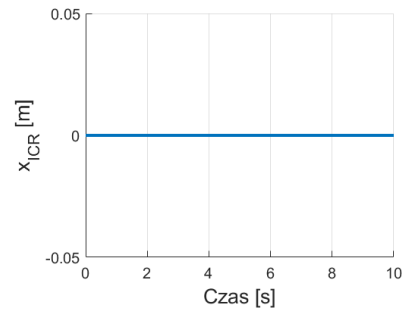


(c) Trajektoria $Y(t)$

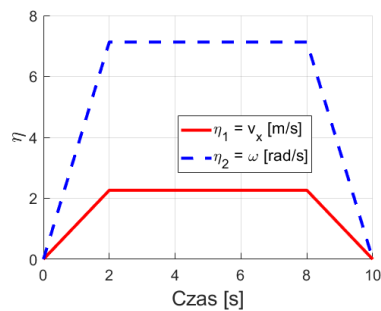
Rysunek 2.5: Warunki początkowe W_1 (2.43): wykresy położenia robota na płaszczyźnie



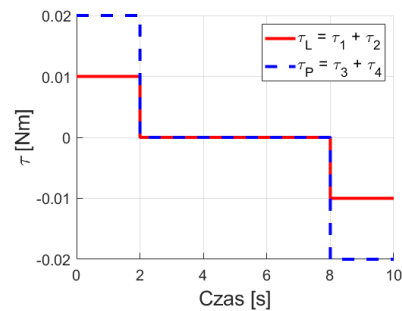
(a) Zmiana orientacji $\theta(t)$



(b) Wartość $x_{ICR}(t)$

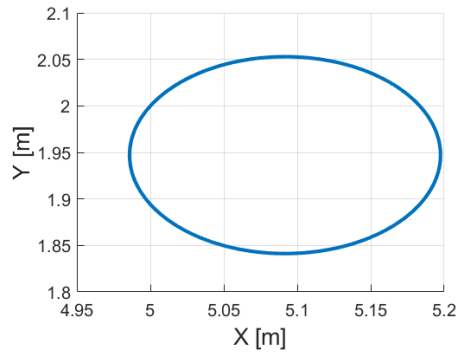
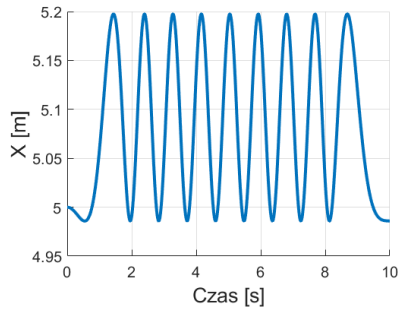
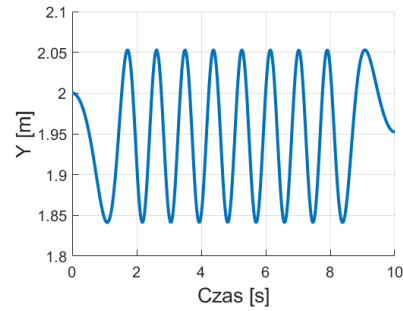
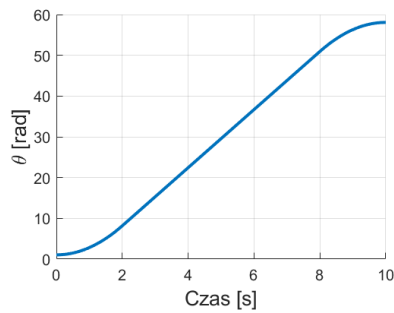
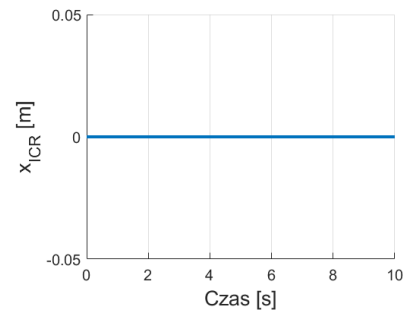
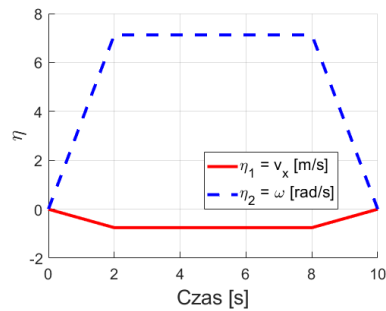
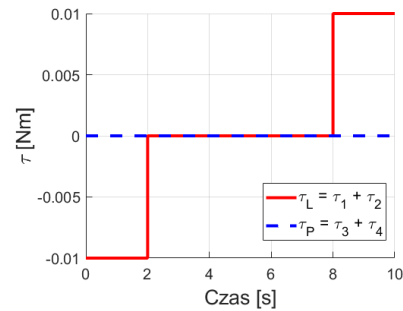


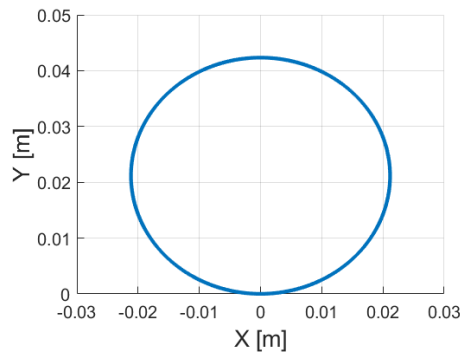
(c) Prędkości $\eta(t)$



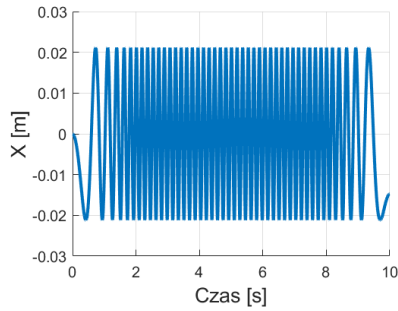
(d) Momenty $\tau(t)$

Rysunek 2.6: Warunki początkowe W_1 (2.43): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie

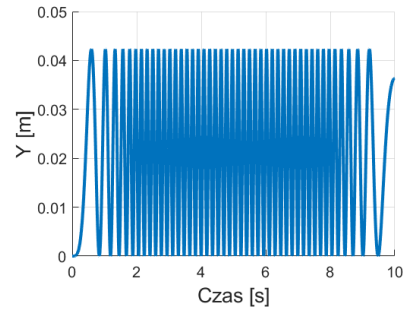
(a) Wykres trajektorii robota na płaszczyźnie XY (b) Trajektoria $X(t)$ (c) Trajektoria $Y(t)$ Rysunek 2.7: Warunki początkowe W_2 (2.44): wykresy położenia robota na płaszczyźnie(a) Zmiana orientacji $\theta(t)$ (b) Wartość $x_{ICR}(t)$ (c) Prędkości $\eta(t)$ (d) Momenty $\tau(t)$ Rysunek 2.8: Warunki początkowe W_2 (2.44): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie



(a) Wykres trajektorii robota na płaszczyźnie XY

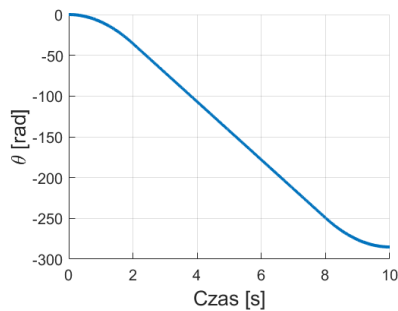


(b) Trajektoria $X(t)$

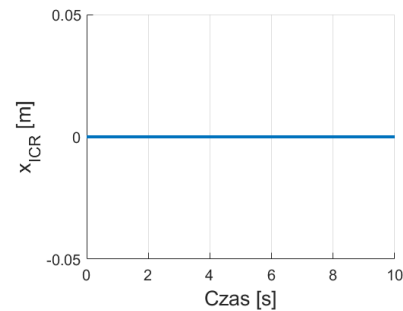


(c) Trajektoria $Y(t)$

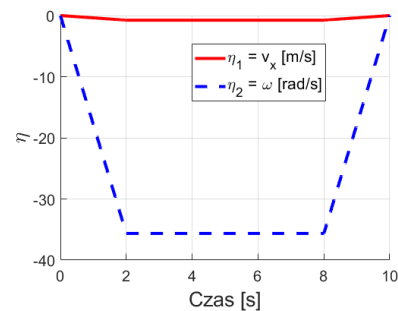
Rysunek 2.9: Warunki początkowe W_3 (2.45): wykresy położenia robota na płaszczyźnie



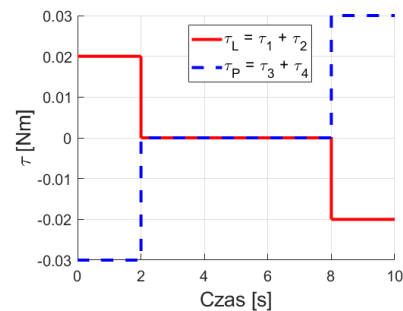
(a) Zmiana orientacji $\theta(t)$



(b) Wartość $x_{ICR}(t)$

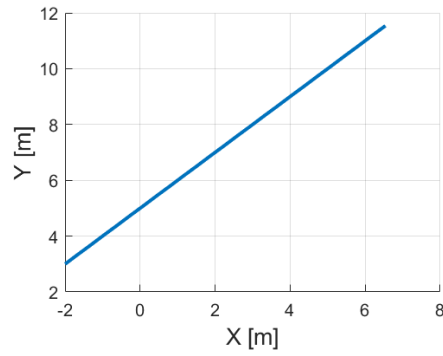
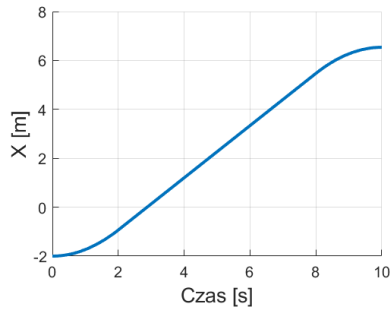
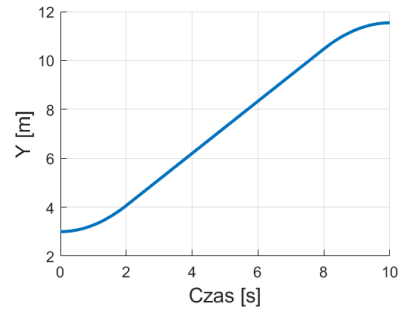
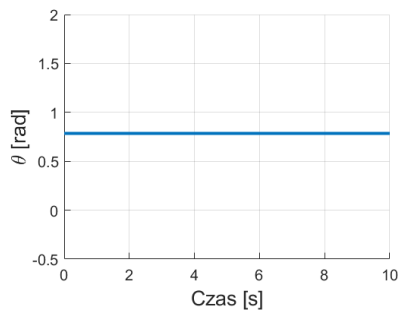
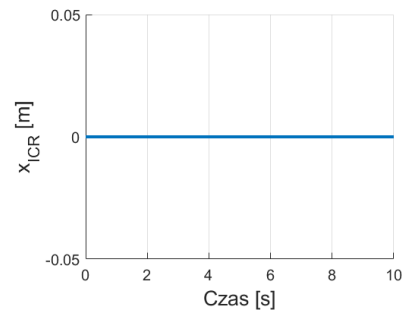
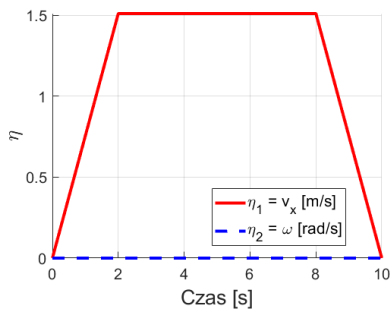
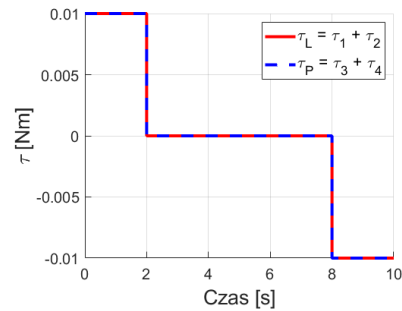


(c) Prędkości $\eta(t)$



(d) Momenty $\tau(t)$

Rysunek 2.10: Warunki początkowe W_3 (2.45): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie

(a) Wykres trajektorii robota na płaszczyźnie XY (b) Trajektoria $X(t)$ (c) Trajektoria $Y(t)$ Rysunek 2.11: Warunki początkowe W_4 (2.46): wykresy położenia robota na płaszczyźnie(a) Zmiana orientacji $\theta(t)$ (b) Wartość $x_{ICR}(t)$ (c) Prędkości $\eta(t)$ (d) Momenty $\tau(t)$ Rysunek 2.12: Warunki początkowe W_4 (2.46): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie

2.3.1.2 Prędkości w postaci szeregu Fouriera

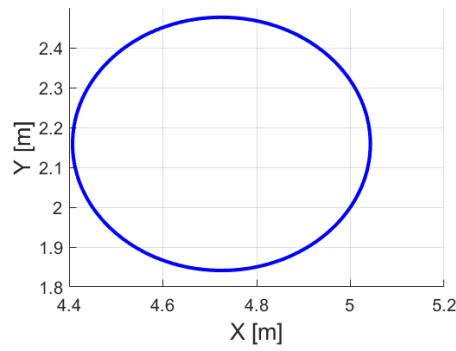
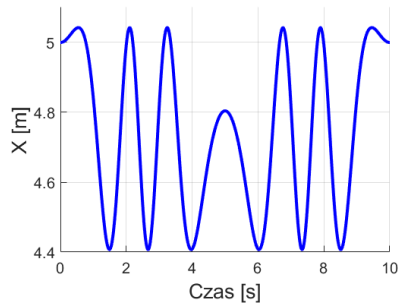
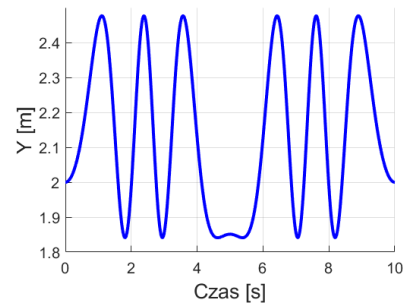
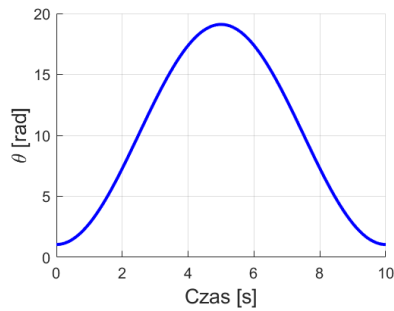
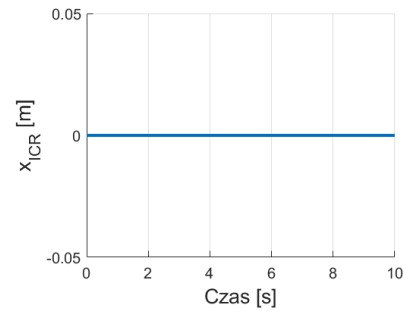
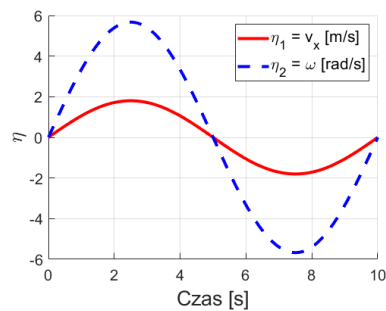
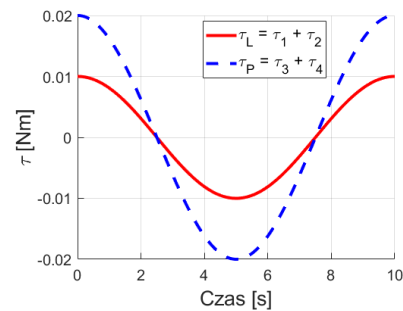
Opis generowania przebiegu prędkości w postaci szeregu Fouriera znajduje się w dodatku C, gdzie zostało wyjaśnione jakie momenty należy zadać, aby uzyskać profil prędkości o charakterze szeregu Fouriera.

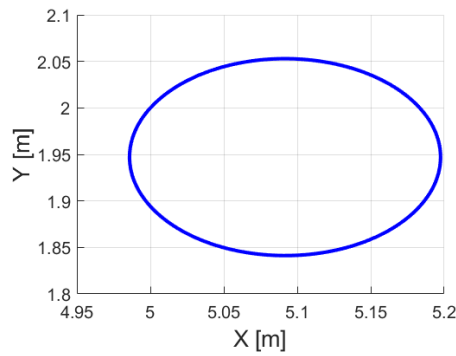
Warunki początkowe W_1 , W_2 oraz W_3 . Wykresy 2.13a, 2.15a i 2.17a pokazują, że przy większym wpływie sterowania jednej ze stron robota, obiekt porusza się po łuku, zakreślając okrąg – tak jak w przypadku prędkości o profilu trapezoidalnym. Tutaj podobnie, wielkości średnic okręgów zależą od zadanego wzmocnienia. Na podstawie wykresów $X(t)$ (2.13b, 2.15b) oraz $Y(t)$ (2.13c, 2.15c) można odczytać jak wiele okręgów robot zakreślił w czasie T – jest to 7 pełnych okręgów, przy czym trzeba wziąć pod uwagę, że zgodnie ze specyfiką sterowania, robot zatoczył 3,5 okręgu jadąc w jedną stronę, a następnie 3,5 okręgu jadąc w drugą stronę. Na tych samych wykresach widać moment zmiany kierunku jazdy robota. Wykresy $X(t)$ 2.13b – 2.17b to przebiegi cosinusoidalne, a wykresy $Y(t)$ 2.13c – 2.17c to przebiegi sinusoidalne – jak w przypadku sterowania trapezoidalnego (równanie (2.47)).

Wykresy 2.14c, 2.14d – 2.16c, 2.16d pokazują, że istotnie, prędkości zapisano w postaci szeregu Fouriera, a rysunki 2.14d, 2.16d i 2.18d pokazują, że ruch odbywał się po okręgu. Świadczy o tym występowanie składowej prędkości kątowej ω widocznej na wykresach prędkości.

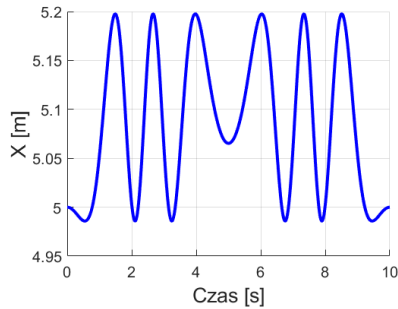
Warunek początkowy W_4 . Tym razem przebadano zachowanie modelu przy sterowaniu, które zapewni ruch po prostej (rys. 2.19a). Na wykresach 2.19b oraz 2.19c pojawia się łagodny, powolny start i powolne hamowanie modelu w strefach wzrostu prędkości η , bądź też ich spadku 2.20c. Wykres 2.20a pokazuje, że robot pozostał w stałej orientacji. Skoro ruch nie odbywa się po okręgu to nie występuje tutaj składowa prędkości kątowej ω , co jest zgodne z intuicją (rysunek 2.20c).

We wszystkich przypadkach (2.6b – 2.12b) wartość x_{ICR} wynosi zero zgodnie z założeniami przytoczonymi w 2.3.1.

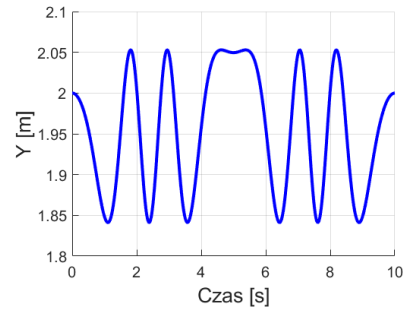
(a) Wykres trajektorii robota na płaszczyźnie XY (b) Trajektoria $X(t)$ (c) Trajektoria $Y(t)$ Rysunek 2.13: Warunki początkowe W_1 (2.43): wykresy położenia robota na płaszczyźnie(a) Zmiana orientacji $\theta(t)$ (b) Wartość $x_{ICR}(t)$ (c) Prędkości $\eta(t)$ (d) Momenty $\tau(t)$ Rysunek 2.14: Warunki początkowe W_1 (2.43): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie



(a) Wykres trajektorii robota na płaszczyźnie XY

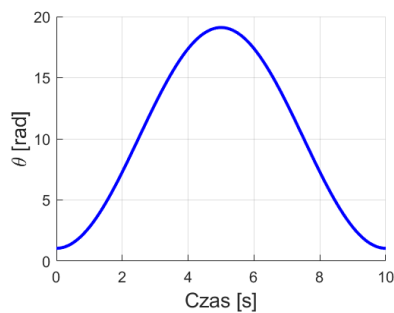


(b) Trajektoria $X(t)$

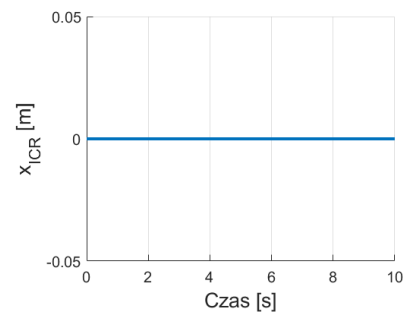


(c) Trajektoria $Y(t)$

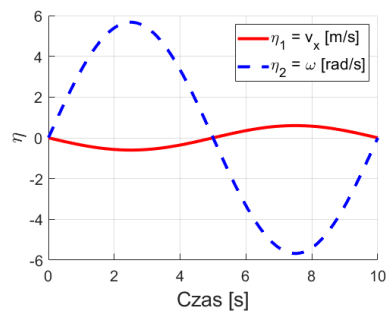
Rysunek 2.15: Warunki początkowe W_2 (2.44): wykresy położenia robota na płaszczyźnie



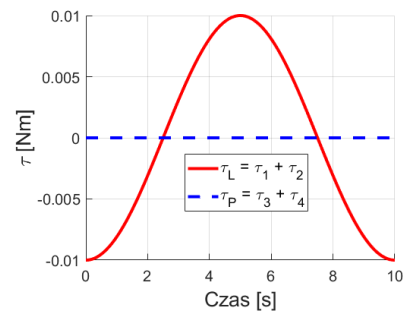
(a) Zmiana orientacji $\theta(t)$



(b) Wartość $x_{ICR}(t)$

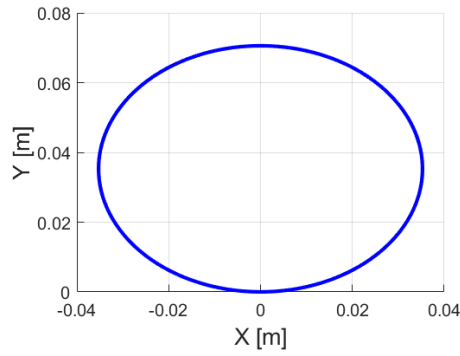
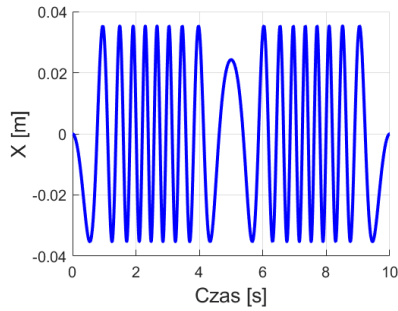
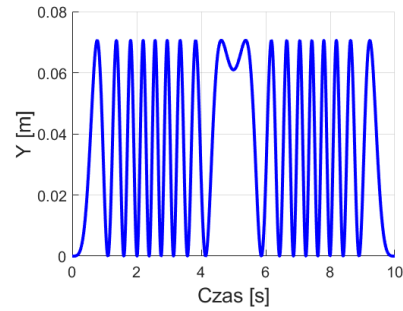
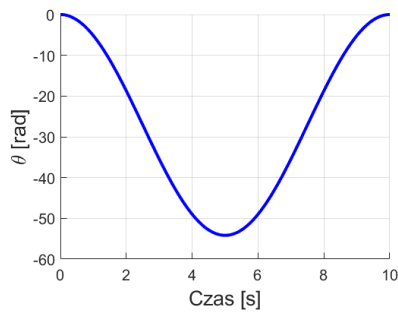
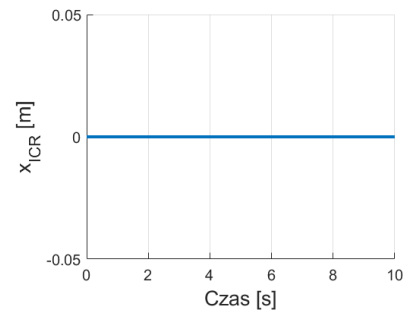
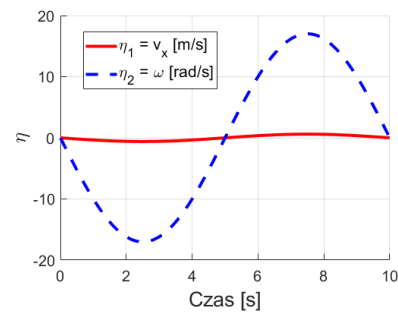
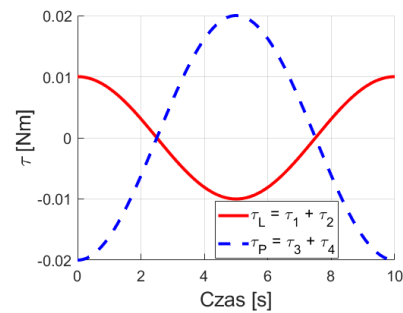


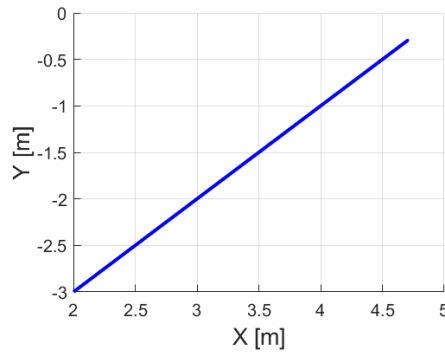
(c) Prędkości $\eta(t)$



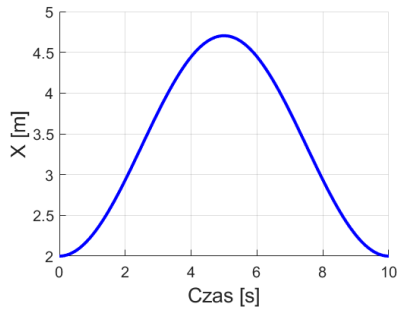
(d) Momenty $\tau(t)$

Rysunek 2.16: Warunki początkowe W_2 (2.44): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie

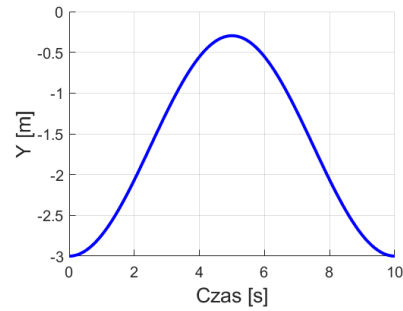
(a) Wykres trajektorii robota na płaszczyźnie XY (b) Trajektoria $X(t)$ (c) Trajektoria $Y(t)$ Rysunek 2.17: Warunki początkowe W_3 (2.45): wykresy położenia robota na płaszczyźnie(a) Zmiana orientacji $\theta(t)$ (b) Wartość $x_{ICR}(t)$ (c) Prędkości $\eta(t)$ (d) Momenty $\tau(t)$ Rysunek 2.18: Warunki początkowe W_3 (2.45): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie



(a) Wykres trajektorii robota na płaszczyźnie XY

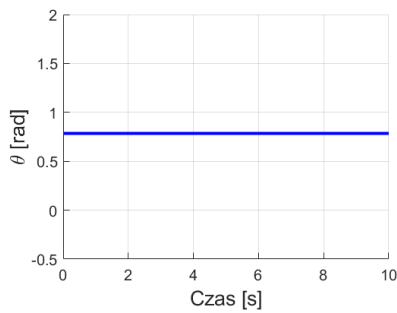


(b) Trajektoria $X(t)$

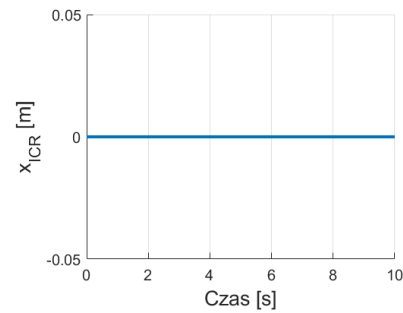


(c) Trajektoria $Y(t)$

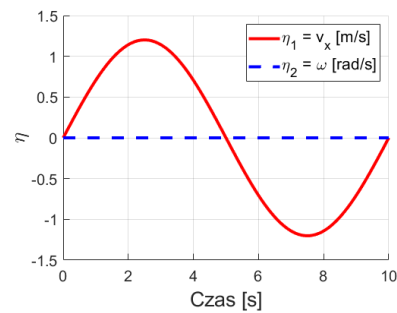
Rysunek 2.19: Warunki początkowe W_4 (2.46): wykresy położenia robota na płaszczyźnie



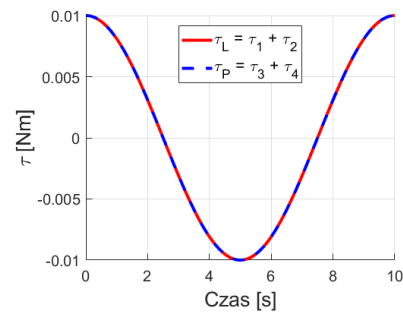
(a) Zmiana orientacji $\theta(t)$



(b) Wartość $x_{ICR}(t)$



(c) Prędkości $\eta(t)$



(d) Momenty $\tau(t)$

Rysunek 2.20: Warunki początkowe W_4 (2.46): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie

2.3.2 Model rzeczywisty (z poślizgiem)

W niniejszym podrozdziale sprawdzamy jak reaguje na wybrane funkcje sterujące robot opisany modelem dynamiki postaci (wzór (2.38))

$$\bar{M}\dot{\eta} + \bar{C}\eta + \bar{R} = \bar{B}\tau.$$

W skład modelu wchodzi macierz $R(q)$, która definiuje tarcie niezbędne do wywołania poślizgu. Natomiast w macierzy generatorów $S(q)$ pojawi się zależność od x_{ICR} , jak we wzorze (2.20)

$$S(q) = \begin{bmatrix} \cos \theta & x_{ICR} \sin \theta \\ \sin \theta & -x_{ICR} \cos \theta \\ 0 & 1 \end{bmatrix}.$$

Badając taki model należy mieć na uwadze, że istnieje możliwość

- nieintuicyjnego poruszania się robota na płaszczyźnie XY ,
- pojawienia się zniekształceń w trapezoidalnym profilu prędkości,
- gwałtownego zatrzymania się robota na skutek wystąpienia zbyt dużego tarcia.

2.3.2.1 Prędkości o profilu trapezoidalnym

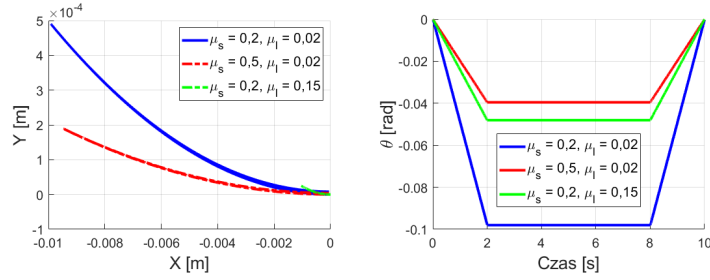
Szczegółowe informacje dotyczące formy funkcji sterujących znajdują się w dodatku B.

Warto podkreślić, że uzyskanie trapezoidalnego profilu prędkości dla tak małego robota jest trudne, gdyż należy uzyskać bardzo duże prędkości, które pozwolą na pokonanie sił tarcia. Dlatego też we wszystkich przypadkach robot zatrzymywał się na jakiś okres czasu. Wyniki dla warunków początkowych W_3 (2.45) oraz W_4 (2.46) różnią się znacząco od tych uzyskanych dla modelu nieholonomicznego.

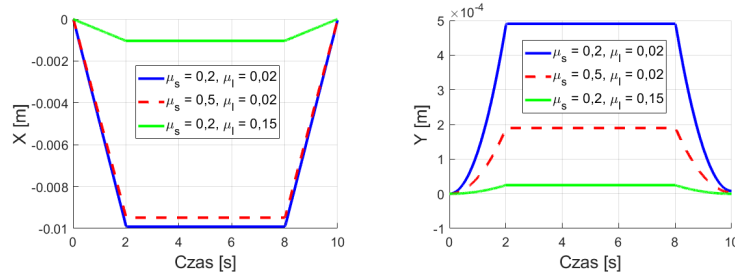
Wpływ μ_s . Wykresy 2.21a, 2.21c, 2.21d oraz 2.25a, 2.25c, 2.25d pokazują, że parametr μ_s określający poślizg wzdłużny nie wpływa znacząco na drogę zakreślaną w kierunku osi OX . Wpływ ma natomiast na poruszanie się w kierunku osi OY oraz na orientację jaką osiąga robot, ale tylko i wyłącznie wtedy, gdy robot, w warunkach idealnych, poruszałby się po łuku. W przypadku jazdy po prostej, zmienna ta nie wpływa na orientację czy kierunek poruszania się w osi OY . Z wykresów 2.23a, 2.23c, 2.23d (por. 2.21a, 2.21c, 2.21d) oraz 2.27a, 2.27c, 2.27d wynika, że zmiana x_{ICR} nie wpływa na działanie zmiennej μ_s , ani na sterowania η (por. rysunki 2.22 i 2.24 oraz 2.26 i 2.28). Sterowania różnią się między sobą ze względu na różne wartości współczynnika tarcia μ_s .

Wpływ μ_l . Wykresy 2.21a, 2.21c, 2.21d oraz 2.25a, 2.25c, 2.25d pokazują, że parametr określający poślizg poprzeczny znacząco wpływa na drogę zakreślaną w kierunku osi OX , OY oraz na orientację jaką robot osiąga. Brak wpływu na orientację obserwować można, gdy robot porusza się po linii prostej. Z wykresów 2.23a, 2.23c, 2.23d oraz 2.27a, 2.27c, 2.27d wynika, że zmiana x_{ICR} nie wpływa na działanie zmiennej μ_l , ani na sterowania η (rysunki 2.22, 2.24, 2.26, 2.28). Sterowania różnią się między sobą ze względu na różne wartości współczynnika tarcia μ_l .

Wpływ x_{ICR} . Na podstawie wykresów prędkości pomocniczych η , przy takim samym sterowaniu τ , (rysunki 2.22, 2.24, 2.26, 2.28) widać, że zmienna odpowiedzialna za poślizg x_{ICR} nie ma znaczącego wpływu na sterowanie. Zmienna ta nie ma również większego wpływu na kształty trajektorii.

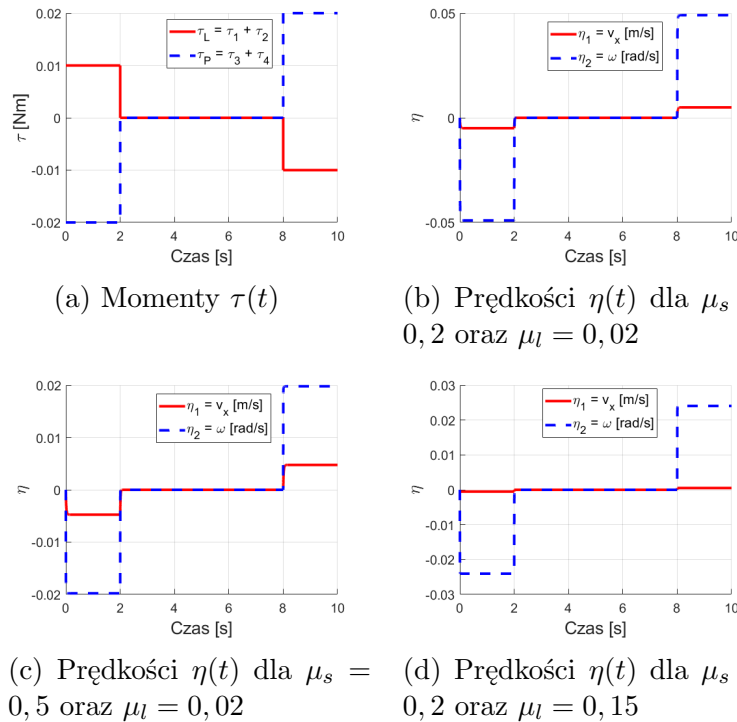


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

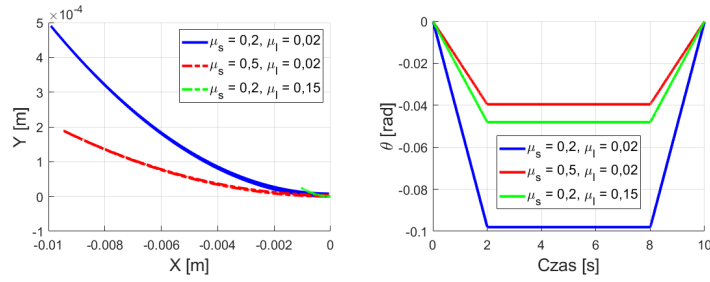
Rysunek 2.21: Warunki początkowe W_3 (2.45): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości



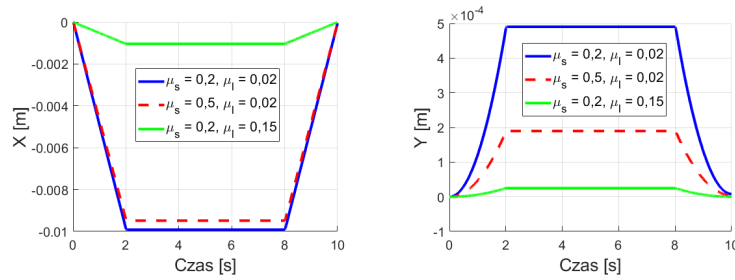
(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$

(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.22: Warunki początkowe W_3 (2.45): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości

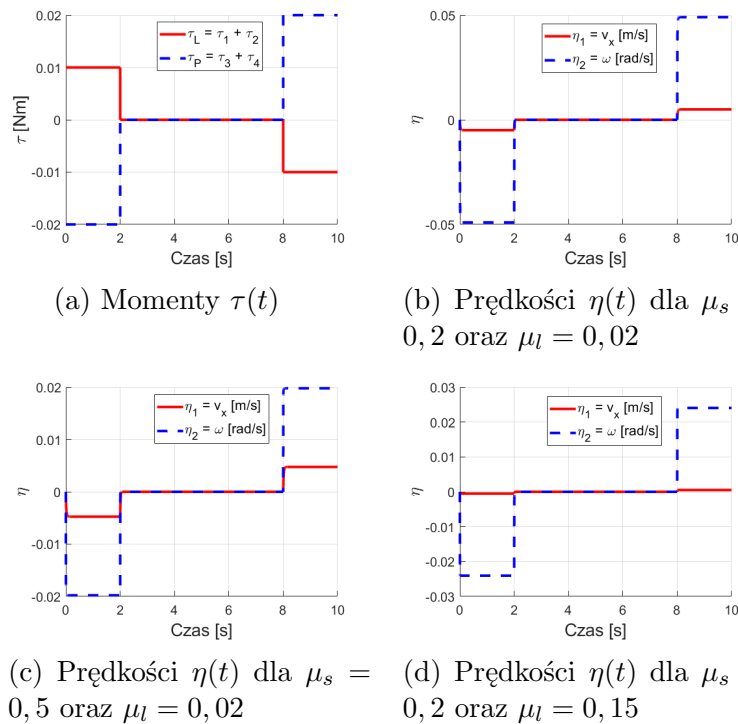


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

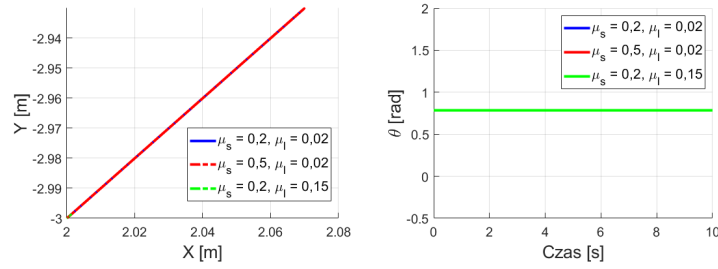
Rysunek 2.23: Warunki początkowe W_3 (2.45): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości



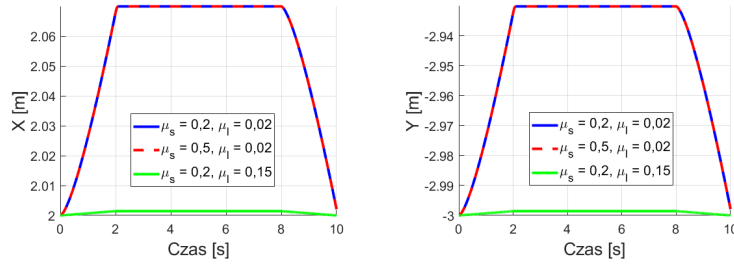
(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$

(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.24: Warunki początkowe W_3 (2.45): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości



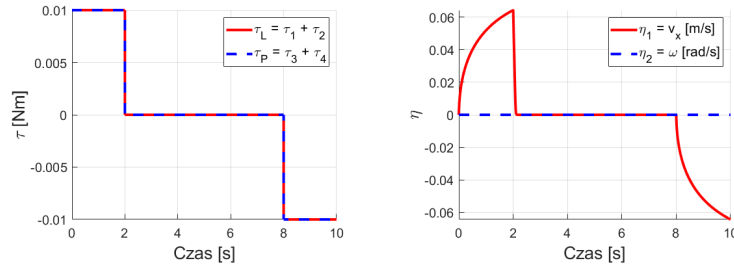
(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$

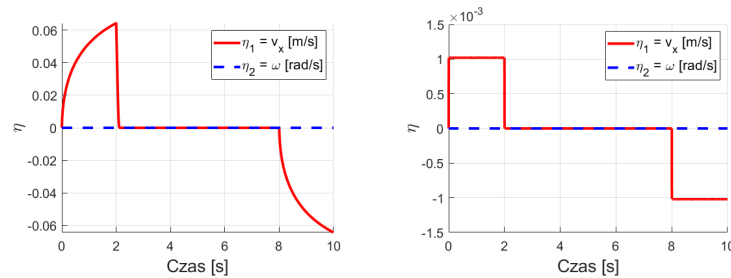
(d) Trajektoria $Y(t)$

Rysunek 2.25: Warunki początkowe W_4 (2.46): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości



(a) Momenty $\tau(t)$

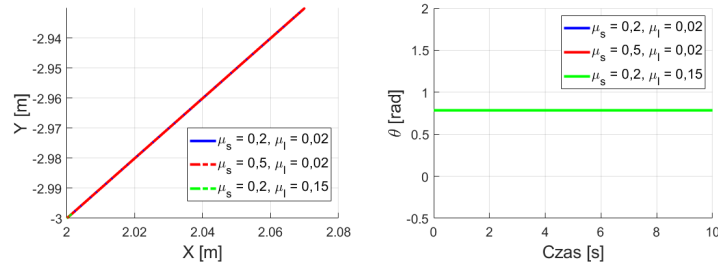
(b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$



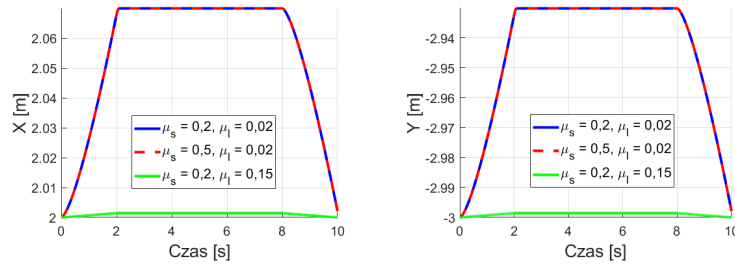
(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$

(d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.26: Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości



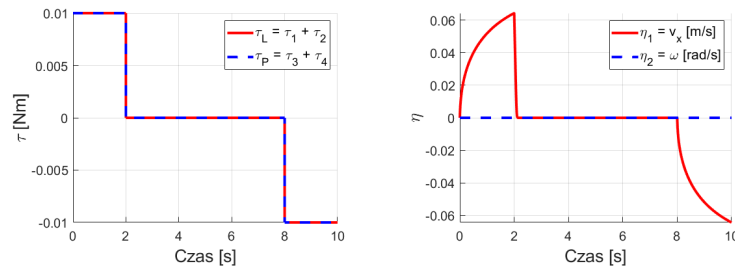
(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$

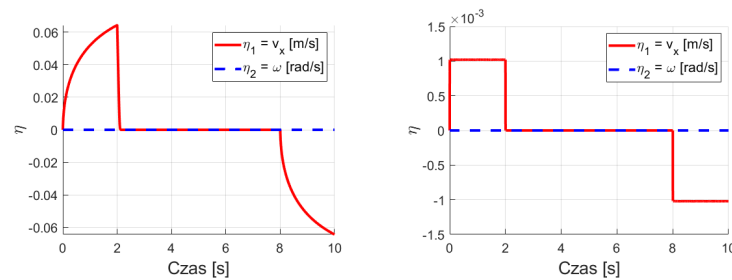
(d) Trajektoria $Y(t)$

Rysunek 2.27: Warunki początkowe W_4 (2.46): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości



(a) Momenty $\tau(t)$

(b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$



(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$

(d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.28: Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości

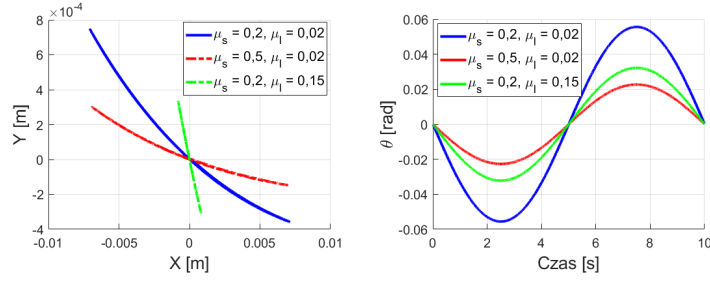
2.3.2.2 Prędkości w postaci szeregu Fouriera

Po szczegóły w kwestii matematycznej formuły sterowań odnosimy do dodatku C. Podkreślenia wymaga fakt, że wyniki różnią się znacząco od tych uzyskanych dla modelu idealnego. Spowodowane jest to uwzględnieniem tarcia w modelu robota, które znacząco ogranicza możliwość poruszania się robota. Jak w przypadku profilu trapezoidalnego badania przeprowadzono dla warunków początkowych W_3 (2.45) oraz W_4 (2.46).

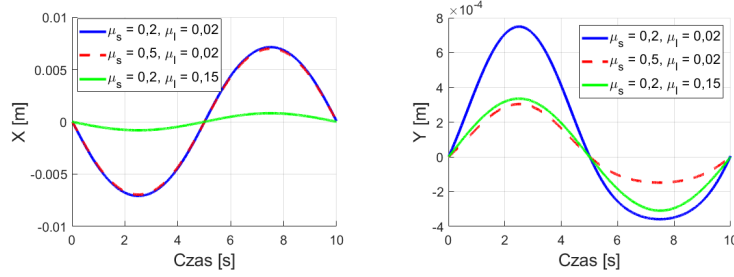
Wpływ μ_s . Wykresy 2.29a, 2.29c, 2.29d oraz 2.33a, 2.33c, 2.33d pokazują, że parametr określający poślizg wzdłużny nie wpływa znacząco na drogę określaną w kierunku osi OX . Wpływ ma natomiast na poruszanie się w kierunku osi OY oraz na orientację jaką osiąga robot, ale tylko i wyłącznie wtedy, gdy robot, w warunkach idealnych, poruszałby się po łuku, w innym przypadku, zmienna ta nie wpływa na orientację czy kierunek poruszania się w osi OY . Z porównania wykresów 2.31a, 2.31c, 2.31d i 2.29a, 2.29c, 2.29d oraz 2.35a, 2.35c, 2.35d i 2.33a, 2.33c, 2.33d wynika, że zmiana x_{ICR} nie wpływa na działanie zmiennej μ_s , ani na sterowania η (rysunki 2.30, 2.32, 2.34, 2.36). Sterowania różnią się między sobą ze względu na różne wartości współczynnika tarcia μ_s .

Wpływ μ_l . Z wykresów 2.29a, 2.29c, 2.29d oraz 2.33a, 2.33c, 2.33d widać, że parametr określający poślizg poprzeczny znacząco wpływa na drogę określaną w kierunku osi OX , OY oraz na orientację jaką robot osiąga. Brak wpływu na orientację obserwować można, gdy robot porusza się po linii prostej. Z wykresów 2.31a, 2.31c, 2.31d oraz 2.35a, 2.35c, 2.35d wynika, że zmiana x_{ICR} nie wpływa na działanie zmiennej μ_l , ani na sterowania η (rysunki 2.30, 2.32, 2.34, 2.36). Sterowania różnią się między sobą ze względu na różne wartości współczynnika tarcia μ_l .

Wpływ x_{ICR} . Na podstawie wykresów prędkości pomocniczych η , przy takim samym sterowaniu τ , (rysunki 2.30, 2.32, 2.34, 2.36) widać, że zmienna odpowiedzialna za poślizg x_{ICR} nie ma dużego wpływu na sterowanie. Zmienna ta nie ma również większego wpływu na kształty trajektorii.

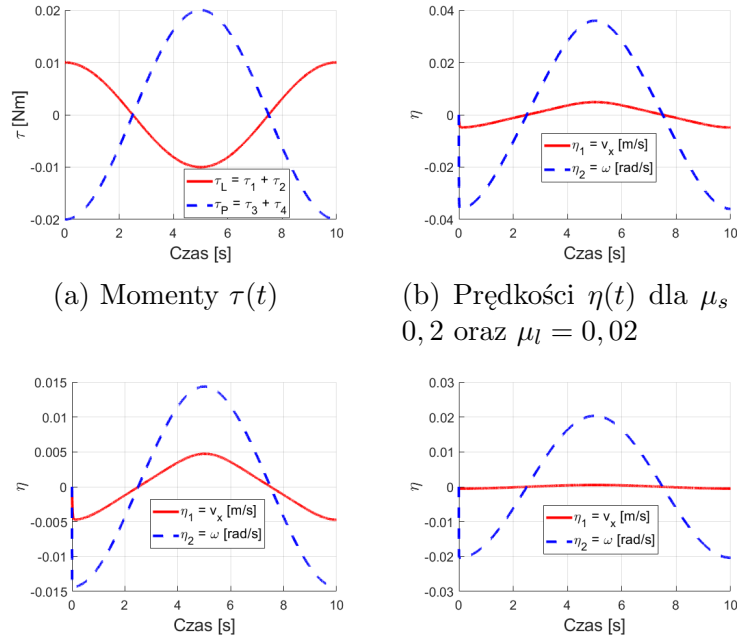


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

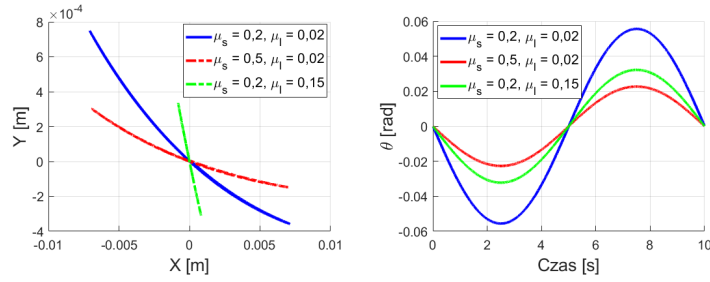
Rysunek 2.29: Warunków początkowe W_3 (2.45): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,01$ m – szereg Fouriera



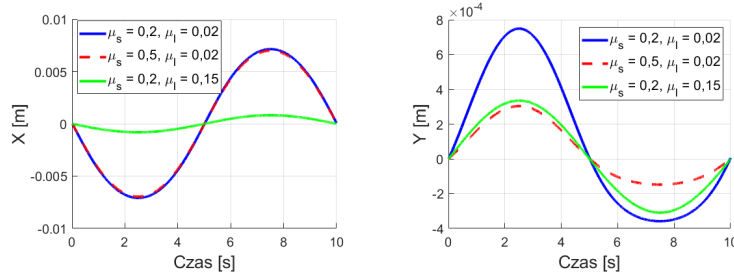
(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$

(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.30: Warunki początkowe W_3 (2.45): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – szereg Fouriera

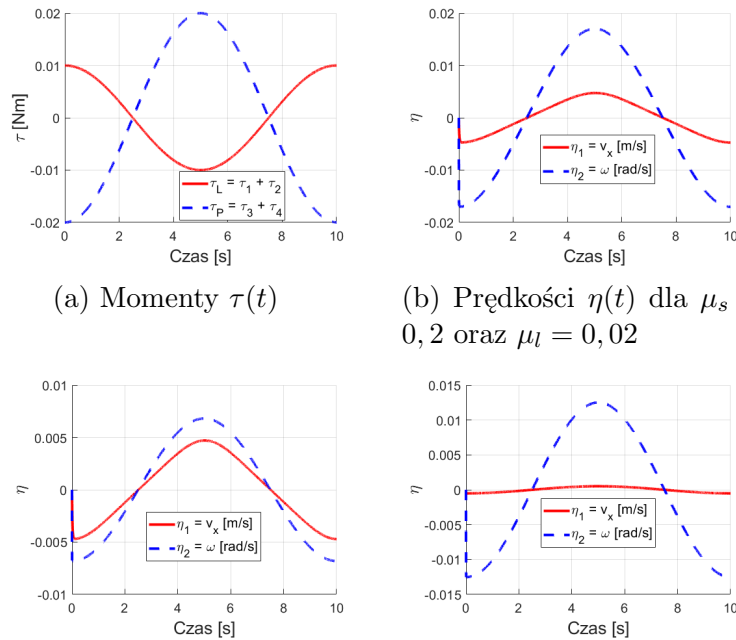


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

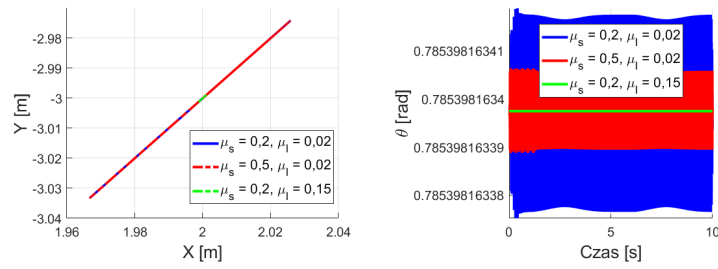
Rysunek 2.31: Warunków początkowe W_3 (2.45): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,03$ m – szereg Fouriera



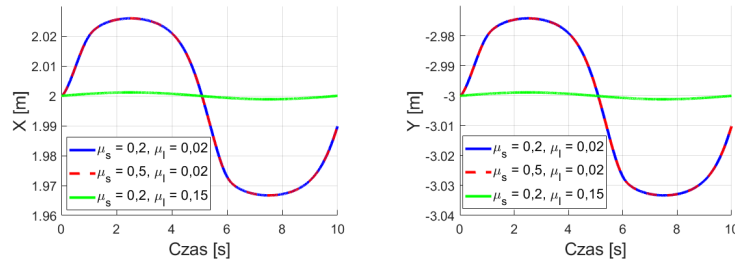
(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$

(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.32: Warunki początkowe W_3 (2.45): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – szereg Fouriera



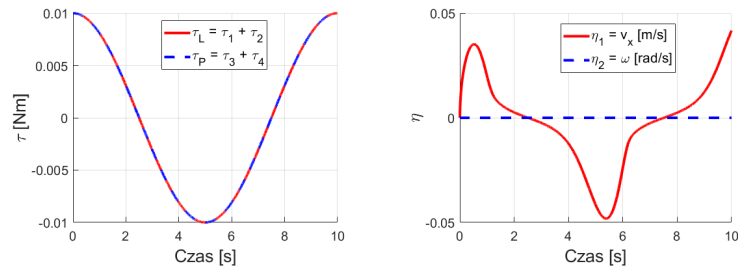
(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$

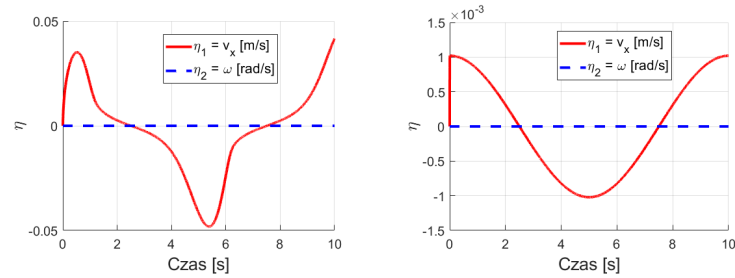
(d) Trajektoria $Y(t)$

Rysunek 2.33: Warunków początkowe W_4 (2.46): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,01$ m – szereg Fouriera



(a) Momenty $\tau(t)$

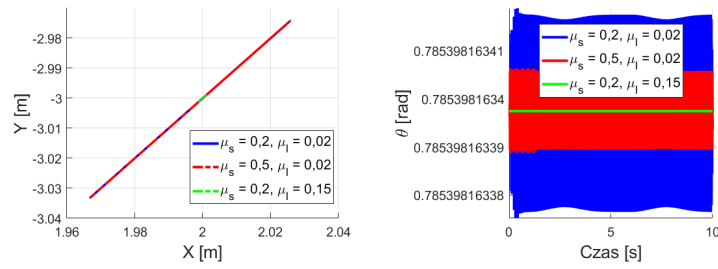
(b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$



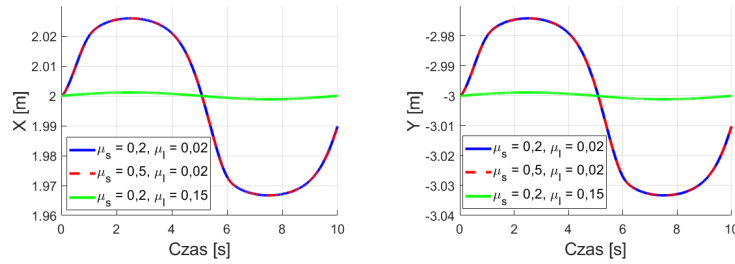
(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$

(d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.34: Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – szereg Fouriera

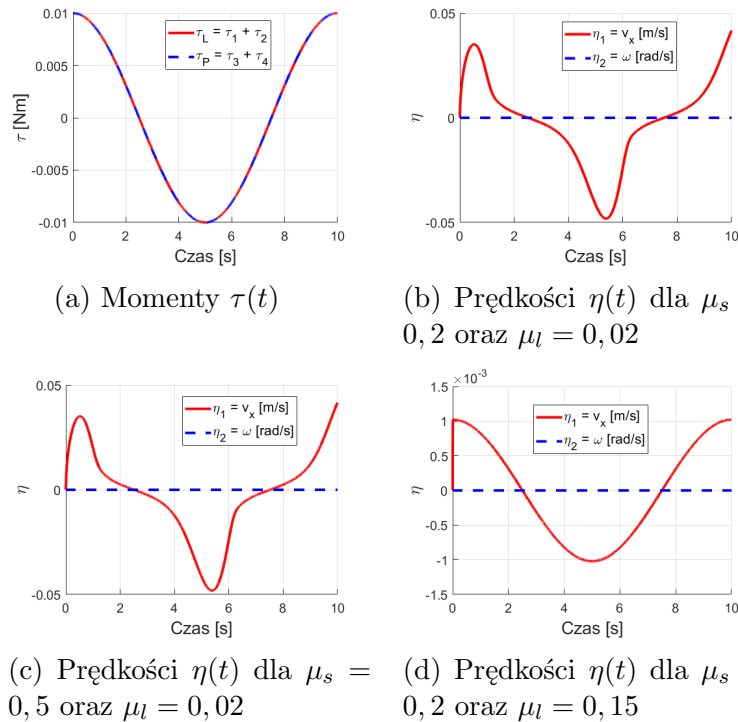


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



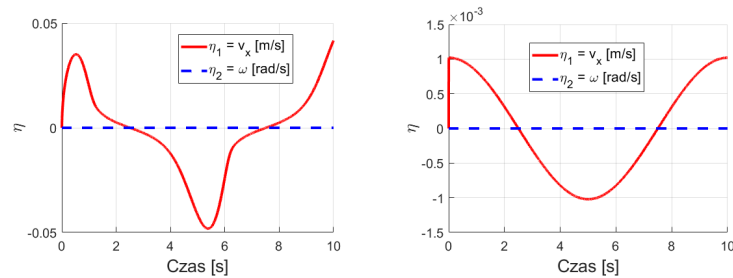
(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

Rysunek 2.35: Warunków początkowe W_4 (2.46): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,03$ m – szereg Fouriera



(a) Momenty $\tau(t)$

(b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$



(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$

(d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.36: Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – szereg Fouriera

2.3.2.3 Duże momenty dla prędkości o profilu trapezoidalnym

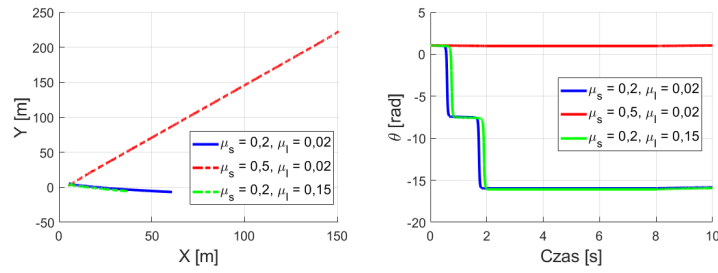
Jako, że w poprzednich rozdziałach robot często zatrzymywał się, ponieważ nie potrafił pokonać sił oporów (tarcia), postanowiono, że badania zostaną powtórzone dla ruchu po łuku (warunki W_1) dla różnych nastaw wartości modulujących przebiegi trapezoidalne. Zdecydowano przebadать dwa przypadki

$$W_{1.1}(q, \eta, \tau) = [5 \ 2 \ 60^\circ \ 0 \ 0 \ 0,3 \ 0 \ 0,1 \ 0,1]^T, \quad (2.48)$$

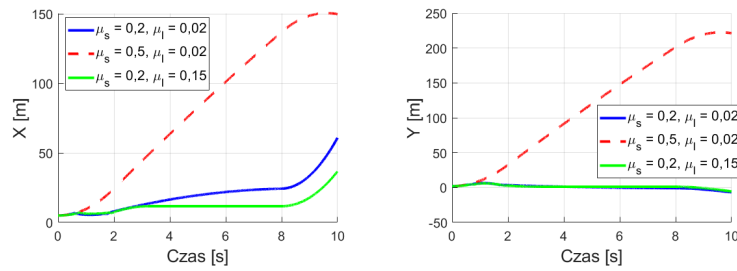
$$W_{1.2}(q, \eta, \tau) = [5 \ 2 \ 60^\circ \ 0 \ 0 \ 1 \ 1 \ 0,1 \ 0,1]^T. \quad (2.49)$$

Okazuje się, że tak dobrane momenty pozwalają na pokonanie sił tarcia, ale robot porusza się z nieosiągalnymi w rzeczywistości dla tak małego robota prędkościami. Wyniki można uznać jako dowód na to, że jest możliwe zadanie takich momentów τ , aby pokonać siłę tarcia i wprowadzić robota w ciągły ruch. Co ciekawe, widać, że tarcie jeszcze bardziej oddziałuje na model, gdyż profil prędkości jest bardzo mocno zniekształcony, wręcz nie przypomina profilu trapezoidalnego, bądź też przypomina go, ale jest lekko przesunięty gdy momenty zostają wyłączone.

Warto zauważyć (patrz rys. 2.38), że dla momentów rzędu 10^{-1} ważną rolę odgrywa jeszcze współczynnik tarcia w kierunku wzdłużnym μ_s i im jest on większy, tym bardziej profil prędkości zbliżony jest do trapezoidalnego. W początkowej fazie ruchu, gdy momenty są największe, następują gwałtowne zmiany prędkości (szarpanie robotem, gwałtowne zmiany orientacji) dla zbyt małego współczynnika tarcia μ_s (patrz rysunki 2.38b, 2.38d). Natomiast dla momentów rzędu 10^0 współczynniki tarcia przestają odgrywać jakąkolwiek rolę, a robot zdaje w dużym stopniu poruszać głównie pod wpływem poślizgu. Można stwierdzić, że im wyższa różnica między τ_L a τ_P tym lepiej robot radzi sobie z zakreślanie łuku (rysunek 2.39a). Widać też, że pokonane trasy niewiele różnią się od siebie w osi OY (rys. 2.39d), natomiast w osi OX już odchylenie jest widoczne przy wzroście tarcia poprzecznego (rys. 2.37c). Profile prędkości pomocniczych η w żadnym przypadku nie są trapezoidalne (rysunki 2.40b, 2.40c, 2.40d), co również wskazuje na fakt dużego wpływu sił związanych z poślizgiem. Tutaj również widoczne jest, że w początkowej fazie ruchu, gdy momenty są największe, następują gwałtowne zmiany prędkości robota, a co za tym idzie – szarpanie robotem oraz gwałtowne zmiany jego orientacji.

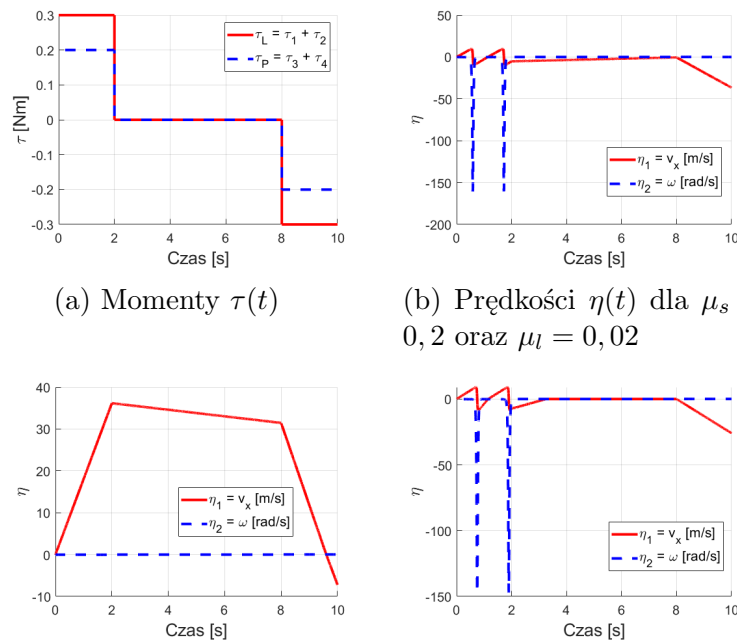


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$



(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

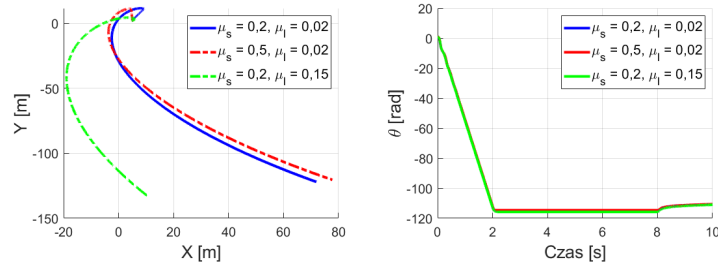
Rysunek 2.37: Warunki początkowe $W_{1.1}$ (2.48): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości



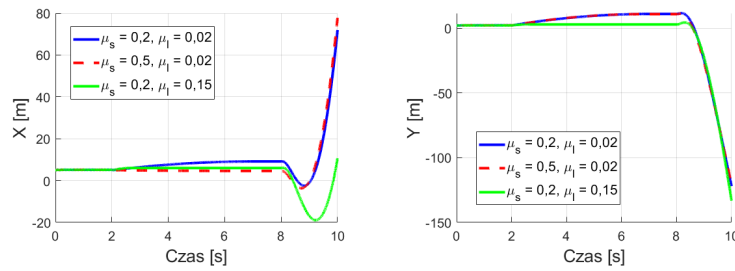
(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$

(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.38: Warunki początkowe $W_{1.1}$ (2.48): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości

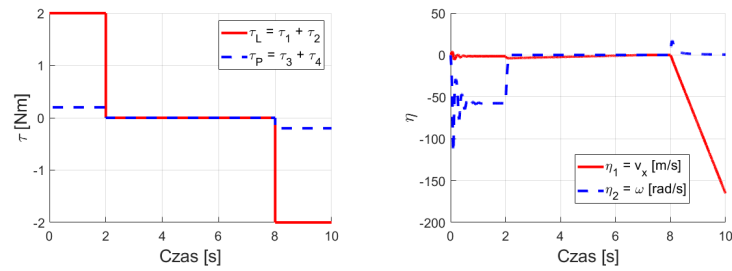


(a) Trajektoria robota na płaszczyźnie XY (b) Zmiana orientacji $\theta(t)$

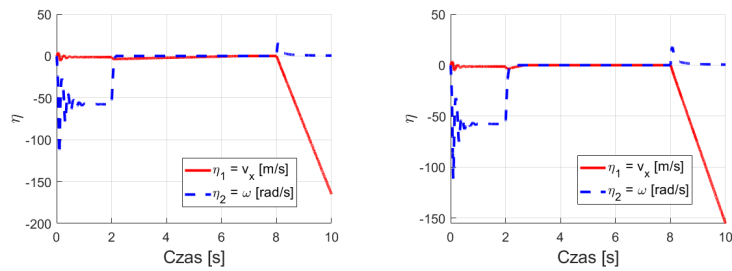


(c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

Rysunek 2.39: Warunki początkowe $W_{1.2}$ (2.49): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości



(a) Momenty $\tau(t)$ (b) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,02$



(c) Prędkości $\eta(t)$ dla $\mu_s = 0,5$ oraz $\mu_l = 0,02$ (d) Prędkości $\eta(t)$ dla $\mu_s = 0,2$ oraz $\mu_l = 0,15$

Rysunek 2.40: Warunki początkowe $W_{1.2}$ (2.49): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości

Rozdział 3

Prawo sterowania

Sterowanie jest to zadanie odpowiednich momentów obrotowych lub prędkości, w celu realizacji postawionego zadania: stabilizacji (dojazdu do ustalonej miejsca) lub przejazdu wzdłuż zadanej trajektorii lub ścieżki. Prawo sterowania natomiast pozwala na wyliczanie odpowiednich wartości prędkości lub momentów obrotowych, które realizować będą postawione zadanie. Badany robot realizuje dwa z trzech podstawowych zadań robotycznych,

- P2P – dojazd do punktu,
- śledzenie trajektorii.

W ramach realizacji tak postawionych zadań proponujemy sterowanie, które w zależności od zadanych wartości referencyjnych, będzie realizować zadanie.

Założeniem, dla którego stosuje się prawa sterowania, jest realizacja zadań przy minimalizacji pewnego błędu, który określa różnicę między obecnym stanem a stanem referencyjnym (idealnym). W idealnym przypadku błąd powinien zbiegać do zera, jednak wiadomym jest, że w rzeczywistych przypadkach jest to wręcz niemożliwe do zrealizowania.

W przypadku badanego robota przyjęto minimalizację błędu pozycji i orientacji

$$\tilde{q} = q - q_r = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} - \begin{bmatrix} X_r \\ Y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} \tilde{X} \\ \tilde{Y} \\ \tilde{\theta} \end{bmatrix}. \quad (3.1)$$

Warto mieć na uwadze, że q_r , czyli wartość referencyjna (taka, do której robot ma dojechać)

- dla sterowania P2P musi być *constant*,
- do realizacji zadania śledzenia trajektorii musi być parametryzowana czasem $q_r(t)$.

Są to warunki konieczne do realizacji proponowanego w dalszej części prawa sterowania.

Aby zrealizować zadanie śledzenia trajektorii należy proponować ścieżkę referencyjną. Można to zrobić w prosty sposób rozwiązując równanie

$$\dot{q}_r = S(q_r)\eta_r. \quad (3.2)$$

gdzie $\eta_r = [v_{rx}(t) \ \omega_r(t)]^T$.

3.1 Integrator Brockett'a i postać normalna

Znaczenie postaci normalnych w teorii sterowania wynika z dwóch powodów. Po pierwsze, postaci normalne pozwalają zrozumieć strukturę i zachowanie sterowania układu. Wg postaci kanonicznej Brunowskiego [3], system zachowuje się jak liczba niezależnych, równoległych łańcuchy integratorów. Po drugie, i najważniejsze, postaci normalne stosuje się do projektowania algorytmów sterowania. Wiemy, że zarówno postać normalna, jak i transformacja sprzężenia zwrotnego zapewniająca przejście z postaci oryginalnej układu do postaci normalnej są znane eksplicite. A zatem możemy, poprzez to znane sprzężenie zwrotne, przetransformować problem sterowania układu oryginalnego do postaci normalnej, tam skorzystać z istniejących i dobrze zdefiniowanych algorytmów sterowania, rozwiązać problem w przestrzeni postaci normalnej, a następnie przenieść rozwiązanie poprzez odwrotne sprzężenie zwrotne z powrotem do pierwotnego systemu. W celu wyprowadzenia algorytmu sterowania dla rozważanego robota należy wykonać szereg następujących transformacji kinematycznych prowadzących do zapisania naszego pierwotnego modelu w postaci normalnej po to, by móc skorzystać z dobrze zdefiniowanych dla postaci normalnych algorytmów sterowania.

Zacznijmy od zróżniczkowania równania opisującego błąd położenia (3.1)

$$\dot{\tilde{q}} = \dot{q} - \dot{q}_r, \quad (3.3)$$

dzięki czemu uzyskujemy błąd prędkości. Błąd prędkości można zapisać również jako

$$\dot{\tilde{q}} = S(q)\eta - S(q_r)\eta_r. \quad (3.4)$$

Model można podzielić na część odpowiadającą za dryf oraz części zależne od zewnętrznych sterowań v_x oraz ω

$$\dot{\tilde{q}} = -S(q_r)\eta_r + \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_x + \begin{bmatrix} x_{ICR} \sin \theta \\ -x_{ICR} \cos \theta \\ 1 \end{bmatrix} \omega.$$

Powyższy model przypomina w zapisie nieholonomiczny integrator zaproponowany przez Brockett'a. Korzystając z tego faktu spróbujemy przekształcić nasz model do postaci normalnej odpowiedniej dla nieholonomicznego integratora, a to z kolei pozwoli na skorzystanie z kinematycznego algorytmu sterowania zaproponowanego dla integratora nieholonomicznego przez Dixon'a w [7]. A zatem przekształćmy (3.4) do postaci normalnej

$$\begin{aligned} \dot{w} &= u^T J^T z + f \\ \dot{z} &= u \end{aligned}, \quad (3.5)$$

gdzie w oraz

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (3.6)$$

to zmienne pomocnicze, u to sygnał wejściowy, a f jest dryfem, definiującym pośrednio zadanie sterowania (gdy $f = 0$ – zadanie dojazdu do punktu, gdy $f \neq 0$ – zadanie śledzenia trajektorii) [2] [7]. Macierz J jest macierzą skośnie symetryczną

$$J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Jako, że system (3.4) reprezentowany jest poprzez równanie różniczkowe pierwszego rzędu, możliwe jest znalezienie globalnego dyfeomorfizmu opisującego zależność między zmiennymi pomocniczymi (w, z) oraz zmiennymi oryginalnymi \tilde{q}

$$Z = \begin{bmatrix} w & z_1 & z_2 \end{bmatrix}^T = P(\theta, \tilde{\theta})\tilde{q}. \quad (3.7)$$

Zmienne w oraz z traktowane są jako nowy wektor stanu układu (3.4).

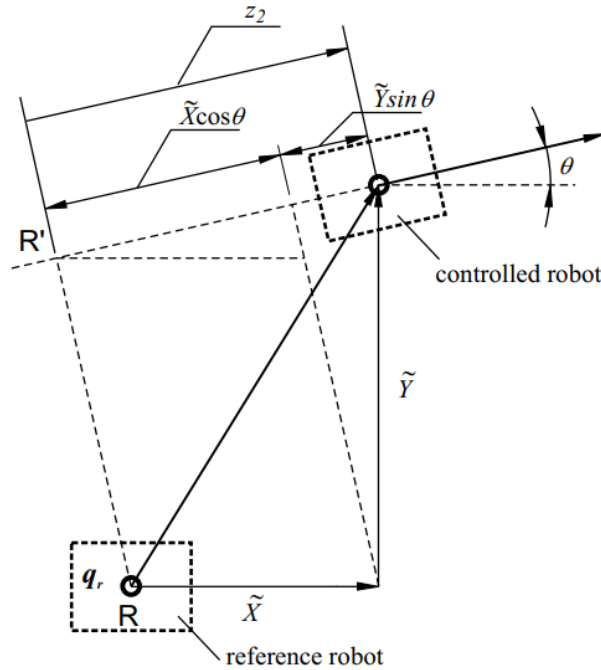
Przy założeniu, że rozważany jest problem regulacji $q_r = \text{constant}$ ($f = 0$), z równania (3.5) otrzymać można

$$\dot{w} = z_2 u_1 - z_1 u_2. \quad (3.8)$$

Bazując na rozważaniach De Luca oraz Oriolo [11] proponuje się następującą zamianę współrzędnych [11]

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \tilde{\theta} \\ \tilde{X} \cos \theta + \tilde{Y} \sin \theta \end{bmatrix}. \quad (3.9)$$

Geometryczna interpretacja nowych zmiennych jest następująca: zmienna z_1 jest błędem orientacji, natomiast $|z_2|$ określa długość rzutu wektora błędu położenia $[\tilde{X} \ \tilde{Y}]^T$ na kierunku określony przez orientację θ . Objaśnienie na rysunku 3.1.



Rysunek 3.1: Interpretacja geometryczna zmiennej z_2

Różniczkując równanie (3.9) otrzymamy

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\theta}} \\ \frac{d}{dt} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) \end{bmatrix}. \quad (3.10)$$

Podstawiając wyrażenie (3.10) do równania (3.8) dostajemy

$$\begin{aligned} \dot{w} &= \dot{\tilde{\theta}} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) - \tilde{\theta} \frac{d}{dt} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) \\ &= \frac{d}{dt} (-\tilde{\theta} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta)) + 2\dot{\tilde{\theta}} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta). \end{aligned} \quad (3.11)$$

Kolejnym krokiem, w wyliczeniu wartości w , jest całkowanie równania (3.11)

$$w = -\tilde{\theta} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) + 2 \int \dot{\tilde{\theta}} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) dt. \quad (3.12)$$

W celu wyliczenia powstałej całki wykorzystamy zapis

$$\frac{d}{dt} (\tilde{X} \sin \theta - \tilde{Y} \cos \theta) = \dot{\theta} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) + \dot{\tilde{X}} \sin \theta - \dot{\tilde{Y}} \cos \theta. \quad (3.13)$$

Na bazie błędu kinematyki (3.4), przy założeniach, że rozwiązać należy zadanie dojazdu do punktu ($\eta_r = [0 \ 0]^T$) można zapisać, że

$$\dot{\tilde{X}} \sin \theta - \dot{\tilde{Y}} \cos \theta = x_{ICR} \dot{\tilde{\theta}}. \quad (3.14)$$

Używając równania (3.14), zależność (3.13) można zapisać jako

$$\begin{aligned} \frac{d}{dt} (\tilde{X} \sin \theta - \tilde{Y} \cos \theta - x_{ICR} \tilde{\theta}) &= \dot{\tilde{X}} \sin \theta + \dot{\theta} \tilde{X} \cos \theta - \dot{\tilde{Y}} \cos \theta + \dot{\theta} \tilde{Y} \sin \theta - \dot{\tilde{\theta}} x_{ICR} \\ &= \dot{\tilde{X}} \sin \theta + \dot{\theta} \tilde{X} \cos \theta - \dot{\tilde{Y}} \cos \theta + \dot{\theta} \tilde{Y} \sin \theta - \dot{\tilde{X}} \sin \theta + \dot{\tilde{Y}} \cos \theta \\ &= \dot{\theta} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) = \dot{\tilde{\theta}} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta). \end{aligned} \quad (3.15)$$

Wówczas na bazie (3.15), wzór (3.12) przyjmie postać

$$w = -\tilde{\theta} (\tilde{X} \cos \theta + \tilde{Y} \sin \theta) + 2 (\tilde{X} \sin \theta - \tilde{Y} \cos \theta - x_{ICR} \tilde{\theta}). \quad (3.16)$$

Należy pamiętać, że wyznaczone powyżej elementy nowego wektora stanu $[w, z_1, z_2]^T$ obowiązują w przypadku zadania regulacji/stabilizacji, a zatem wtedy, gdy $q_r(t) = \text{constant}$. Kolejnym krokiem jest wyliczenie wartości f w (3.5), które określać będzie zadanie sterowania, pamiętamy bowiem, że dryf f jest niezerowy w przypadku zadania śledzenia

trajektorii. W tym celu należy zróżniczkować (3.16)

$$\begin{aligned} \dot{w} = & -\dot{\theta}(\tilde{X} \cos \theta + \tilde{Y} \sin \theta) + \tilde{\theta} \left(\dot{\theta}(\tilde{X} \sin \theta + \tilde{Y} \cos \theta) + \dot{\tilde{X}} \cos \theta - \dot{\tilde{Y}} \sin \theta \right) \\ & + 2\dot{\theta}(\tilde{X} \cos \theta + \tilde{Y} \sin \theta) + 2(\dot{\tilde{X}} \sin \theta - \dot{\tilde{Y}} \cos \theta) - 2x_{ICR}\dot{\theta} \end{aligned}$$

\parallel
używając równań (3.4) i (3.9) dostajemy

$$\begin{aligned} \dot{w} = & -\dot{\theta}z_2 - z_1\dot{z}_2 + 2\dot{\theta}z_2 + 2(x_{ICR}\dot{\theta} - v_{rx} \sin z_1 - x_{ICR}\omega_r \cos z_1) - 2x_{ICR}\dot{\theta} + 2x_{ICR}\omega_r \\ = & \dot{\theta}z_2 + \dot{\theta}_r z_2 - 2\dot{\theta}_r z_2 + 2\dot{\theta}_r z_2 - z_1\dot{z}_2 + 2(x_{ICR}\omega_r - v_{rx} \sin z_1 - x_{ICR}\omega_r \cos z_1) \\ = & \dot{\theta}z_2 - z_1\dot{z}_2 + 2\dot{\theta}_r z_2 + 2(x_{ICR}\omega_r - v_{rx} \sin z_1 - x_{ICR}\omega_r \cos z_1) \\ = & \dot{z}_1 z_2 - z_1\dot{z}_2 + 2z_2\omega_r + 2(x_{ICR}\omega_r - v_{rx} \sin z_1 - x_{ICR}\omega_r \cos z_1) \end{aligned}$$

\parallel
i dalej podstawiając (3.5) otrzymujemy

$$\begin{aligned} \dot{w} = & u_1 z_2 - z_1 u_2 + 2(z_2\omega_r + x_{ICR}\omega_r - v_{rx} \sin z_1 - x_{ICR}\omega_r \cos z_1) \\ = & \dot{z}_1 z_2 - z_1\dot{z}_2 + f, \end{aligned}$$

\parallel
gdzie

$$f = 2(-v_{rx} \sin z_1 + \omega_r(x_{ICR} + z_2 - x_{ICR} \cos z_1)).$$

Dzięki temu można określić nieosobliwą macierz $P(\theta, \tilde{\theta})$ w równaniu 3.7 transformującym oryginalny wektor stanu $\tilde{q} = [\tilde{x}, \tilde{y}, \tilde{\theta}]$ do nowego wektora stanu $[w, z_1, z_2]^T$

$$P(\theta, \dot{\theta}) = \begin{bmatrix} -\dot{\theta} \cos \theta + 2 \sin \theta & -\dot{\theta} \sin \theta - 2 \cos \theta & -2x_{ICR} \\ 0 & 0 & 1 \\ \cos \theta & \sin \theta & 0 \end{bmatrix}.$$

Na bazie równań (3.5) oraz (3.10) można skonstruować równanie, które określać będzie zależność między sterowaniem (mającym sens prędkości) z układu oryginalnego η i pomocniczym u określonym dla nowych współrzędnych

$$u = T^{-1}\eta - \begin{bmatrix} \omega_r \\ v_{rx} \cos z_1 - x_{ICR}\omega_r \sin z_1 \end{bmatrix}, \quad (3.17)$$

gdzie

$$T^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & -l \end{bmatrix} \quad \vee \quad T = \begin{bmatrix} l & 1 \\ 1 & 0 \end{bmatrix}, \quad (3.18)$$

$$l = \tilde{X} \sin \theta - \tilde{Y} \cos \theta \quad (3.19)$$

mającym sens odległości. Przekształcając równanie (3.17) otrzymano formułę odwrotną, pozwalającą wyliczyć prędkości η , które bezpośrednio oddziałują na model oryginalny q

$$\eta = Tu + \Pi, \quad (3.20)$$

gdzie

$$\Pi = \begin{bmatrix} v_{rx} \cos z_1 + \omega_r(-x_{ICR} \sin z_1 + l) \\ \omega_r \end{bmatrix}. \quad (3.21)$$

Wzór (3.20) jest sterownikiem kinematycznym, który odpowiadać będzie za prędkości podawane do modelu oryginalnego. Te prędkości mają być takie, aby zapewnić realizację obu postawionych zadań sterowania, tzn. zadania regulacji i śledzenia. W kolejnym podrozdziale proponujemy sterownik, który zapewni realizację obu zadań.

3.2 Kinematyczne prawo sterowania w zapisie postaci normalnej

Poniższy rozdział prezentuje kinematyczne prawo sterowania zdefiniowane przez Dixon'a w [7]. Celem jest dobranie algorytmu sterowania, który na podstawie prędkości referencyjnych η_{ref} robota wyliczać będzie sterowanie η w 2.14, które z kolei realizować będzie postawione zadanie (stabilizacja/śledzenie). Bazując na [7] przyjęto prawo sterowania w postaci

$$u = u_a - k_2 z, \quad (3.22)$$

gdzie u_a wprowadza zmienne w czasie sprzężenie zwrotne i wyraża się wzorem [6, 2]

$$u_a = \frac{k_1 w + f}{\delta_d^2} J z_d + \Omega_1 z_d. \quad (3.23)$$

Z równania (3.23) wynika, że sygnał ten jest modulowany poprzez pomocniczy sygnał modulujący z_d generowany przez oscylator, a ten opisany jest poprzez liniowe niestacjonarne równanie różniczkowe

$$\dot{z}_d = \frac{\dot{\delta}_d}{\delta_d} z_d + \left(\frac{k_1 w + f}{\delta_d^2} + w \Omega_1 \right) J z_d, \quad (3.24)$$

gdzie $z_d = [z_{d1} \ z_{d2}]^T$, a δ_d oraz Ω_1 wyrażone są jako

$$\Omega_1 = k_2 + \frac{\dot{\delta}_d}{\delta_d} + w \frac{k_1 w + f}{\delta_d^2}, \quad (3.25)$$

$$\delta_d = \alpha_0 e^{-\alpha_1 t} + \epsilon_1, \quad (3.26)$$

a zmienne α_0 , α_1 , k_1 , k_2 oraz ϵ_1 są stałymi wartościami podanymi w tabeli 2.1.

Łatwo pokazać, że $\|z_d(t)\|$ jest zależna od wartości funkcji δ_d . W celu udowodnienia tego można zróżniczkować normę kwadraturową

$$\frac{d}{dt} (z_d^T z_d) = 2 \frac{\dot{\delta}_d}{\delta_d} z_d^T z_d,$$

$$z_d^T(0) z_d(0) = \delta_d^2(0).$$

W takim przypadku można zapisać, że

$$\forall t \geq 0 \quad \|z_d(t)\| = \delta_d(t). \quad (3.27)$$

Twierdzenie 1 *Jeśli trajektoria referencyjna jest ograniczona dla wszystkich czasów $t > 0$ i model kinematyczny spełnia ograniczenie nieholonomiczne (2.17), kinematyczne prawo sterowania dane we wzorach 3.22–3.26 zapewnia, że błędy położenia i orientacji są globalnie jednolicie ograniczone w następujący sposób*

$$\|\tilde{q}\| \geq \kappa_0 e^{-\kappa_1 t} + \kappa_2 \epsilon_1,$$

gdzie ϵ_1 może być dowolnie małe, a κ_0 , κ_1 i κ_2 są dodatnimi stałymi.

Szczegółowy dowód stabilności wykładniczej w małym otoczeniu punktu równowagi, opiera się na teorii stabilności Lyapunowa (metoda opisana w dodatku D) [16]. Kandydat na funkcję Lyapunowa, zaproponowany przez Dixon'a, przyjmuje postać [7]

$$V_1(w, \tilde{z}) = \frac{1}{2}w^2 + \frac{1}{2}\tilde{z}^T\tilde{z}, \quad (3.28)$$

gdzie $\tilde{z} = z_d - z$ to pomocniczy błąd.

Zróżniczkowane równanie (3.28) dla zamkniętej pętli sprzężenia zwrotnego przyjmuje postać

$$\dot{V}_1(w, \tilde{z}) = -k_1w^2 - k_2\tilde{z}^T\tilde{z} \quad (3.29)$$

Patrząc na sygnały w sterowniku kinematycznym i opierając się na pracy Dixon'a, prostą obserwacją jest, że błędy w oraz \tilde{z} zbiegają eksponencjalnie do zera. Oznacza to, że kontroler jest stabilny [7].

Wniosek 1 Parametry α_0 , α_1 oraz ϵ_1 odgrywają bardzo ważną rolę w sterowniku podczas procesu sterowania. Jako, że równanie (3.27) jest spełnione oraz $\lim_{t \rightarrow \infty} z(t) = z_d(t)$, funkcja skalarna $\delta_d(t)$ pozwala wpływać na wszystkie stany przejściowe. Dzięki temu może być ona postrzegana jako funkcja planowania ruchu. Jest to niezbędne, gdy błędy początkowe są duże, a dopuszczalne prędkości robota są ograniczone. Podobnie można rozważać relację między sygnałami oscylatora z_{d1} oraz z_{d2} , które określają warunki fazowe. Z uwagi na (3.27) warto wprowadzić następującą parametryzację

$$z_d(0) = \delta_d(0) \begin{bmatrix} \cos \phi \\ \sin \phi \end{bmatrix}, \quad (3.30)$$

gdzie $\phi \in (-\pi; \pi]$.

3.3 Model dynamiki w zapisie postaci normalnej

Opracowując prawo sterowania dobrze jest rozpatrywać oba modele (kinematyczny i dynamiczny) osobno, jak w przypadku modelowania samego robota. Dynamiczny kontroler ma za zadanie wyliczać odpowiednie sterowania, aby osiągać zadane prędkości referencyjne. Jako, że model jest kaskadowy, a wzory oparte są na zmiennych pomocniczych, należy przekształcić model dynamiki robota (2.38) tak, aby opisywał on relację między najwyższym stopniem kaskady a jednym poziomem kaskady niżej. W celu wyprowadzenia równań dynamiki w ramach wprowadzonego w podrozdziale 3.1 zapisu postaci normalnych wykorzystamy przekształcenie (3.20) do zastąpienia sterowań η w modelu dynamiki (2.38) za pomocą sterowań u z zapisu postaci normalnych

$$\bar{M} \frac{d}{dt} (Tu + \Pi) + \bar{C} (Tu + \Pi) + \bar{R} = \bar{B}\tau,$$

$$\bar{M} (\dot{T}u + T\dot{u} + \dot{\Pi}) + \bar{C} (Tu + \Pi) + \bar{R} = \bar{B}\tau,$$

$$\bar{M}\dot{T}u + \bar{M}T\dot{u} + \bar{M}\dot{\Pi} + \bar{C}Tu + \bar{C}\Pi + \bar{R} = \bar{B}\tau.$$

Dalej mnożąc całe równanie prawostronnie przez T^T

$$T^T \bar{M} \dot{T}u + T^T \bar{M} T \dot{u} + T^T \bar{M} \dot{\Pi} + T^T \bar{C} Tu + T^T \bar{C} \Pi + T^T \bar{R} = T^T \bar{B} \tau$$

oraz wyciągając wspólne elementy przed nawias otrzymujemy

$$T^T \bar{M} T \dot{u} + T^T (\bar{C} T + \bar{M} \dot{T}) + T^T (\bar{C} \Pi + \bar{M} \dot{\Pi} + \bar{R}) = T^T \bar{B} \tau.$$

Po powyższych przekształceniach można zapisać, że model dynamiki w zapisie postaci normalnej przyjmuje formę

$$\bar{\bar{M}} \dot{u} + \bar{\bar{C}} u + \bar{\bar{R}} = \bar{\bar{B}} \tau, \quad (3.31)$$

gdzie

$$\bar{\bar{M}} = T^T \bar{M} T, \quad (3.32)$$

$$\bar{\bar{C}} = T^T (\bar{C} T + \bar{M} \dot{T}), \quad (3.33)$$

$$\bar{\bar{R}} = T^T (\bar{C} \Pi + \bar{M} \dot{\Pi} + \bar{R}), \quad (3.34)$$

$$\bar{\bar{B}} = T^T \bar{B}. \quad (3.35)$$

Macierz $\bar{\bar{M}}$ jest skośnie symetryczna i dodatnio określona.

3.4 Dynamiczne prawo sterowania w zapisie postaci normalnej

Od tego momentu sygnał $u(t)$ nie jest już wejściem sterującym. W oparciu o (3.22) można zapisać, że pożądany sygnał prędkości $u_d(t)$ przyjmuje postać

$$u_d = u_a - k_2 z. \quad (3.36)$$

Model (3.31) może być określony przez u_d , co spowoduje, że model zostanie liniowo sparymetryzowany. Wcześniej model ten był skomplikowany, a dodatkowo nieliniowy. Zastosowanie linearyzacji spowoduje, iż model stanie się globalnie liniowy przez co prosty do rozwiązania. W celu linearyzacji obiektu wybrano kilka parametrów (wektor ϑ (3.38)), które są istotne dla modelu oraz pomagają wytworzyć macierz regresji liniowej Y_d (wzór (3.39))

$$\bar{\bar{M}} \dot{u}_d + \bar{\bar{C}} u_d + \bar{\bar{R}} = Y_d (u_d, \dot{u}_d, \tilde{q}, \theta, \eta_r) \vartheta, \quad (3.37)$$

$$\vartheta = \begin{bmatrix} m \\ I \\ \mu_s m \\ \mu_l m \end{bmatrix}. \quad (3.38)$$

W celu znalezienia Y_d należy rozpisać dynamikę lewej strony równania (3.37) i wybrać odpowiednie wyrażenia, zgodnie z parametrami wektora ϑ . Postać macierzy Y_d wstępnie można zapisać jako macierz ośmiu zmiennych, gdyż u_d definiuje rozmiar lewej strony równania i ma wymiar 2×1 , a wektor ϑ – 4×1 , więc aby równanie było spełnione należy zapewnić odpowiedni wymiar macierzy Y_d (2×4)

$$Y_d(u_d, \dot{u}_d, \tilde{q}, \theta, \eta_r) = {}_{2 \times 4} \begin{bmatrix} {}_{2 \times 1} \Upsilon_1 & {}_{2 \times 1} \Upsilon_2 & {}_{2 \times 1} \Upsilon_3 & {}_{2 \times 1} \Upsilon_4 \end{bmatrix}, \quad (3.39)$$

gdzie

$$\begin{aligned}
{}_{2x1}\Upsilon_1 &= T^T \left(\begin{bmatrix} 1 & 0 \\ 0 & x_{ICR}^2 \end{bmatrix} (T\dot{u}_d + \dot{T}u_d + \dot{\Pi}) + x_{ICR}A(Tu_d + \Pi) \right) \\
{}_{2x1}\Upsilon_2 &= T^T \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} (T\dot{u}_d + \dot{T}u_d + \dot{\Pi}) \\
{}_{2x1}\Upsilon_3 &= T^T \begin{bmatrix} \cos \theta \sum_{i=1}^4 g\widehat{sgn}(v_{xi}) \\ x_{ICR} \sin \theta \sum_{i=1}^4 g\widehat{sgn}(v_{xi}) + c \left[-\sum_{i=1,2} g\widehat{sgn}(v_{xi}) + \sum_{i=3,4} g\widehat{sgn}(v_{xi}) \right] \end{bmatrix} \\
{}_{2x1}\Upsilon_4 &= T^T \begin{bmatrix} -\sin \theta \sum_{i=1}^4 g\widehat{sgn}(v_{yi}) \\ x_{ICR} \cos \theta \sum_{i=1}^4 g\widehat{sgn}(v_{yi}) - a \sum_{i=1,4} g\widehat{sgn}(v_{yi}) + b \sum_{i=2,3} g\widehat{sgn}(v_{yi}) \end{bmatrix} \\
A &= \begin{bmatrix} 0 & \dot{\theta} \\ -\dot{\theta} & \dot{x}_{ICR} \end{bmatrix}.
\end{aligned}$$

Wybrane parametry wektora ϑ są niepewne ze względu na brak dokładnego określenia parametrów dynamicznych robota. Można zatem zaproponować błąd wartości oszacowanych $\tilde{\vartheta}$, który będzie definiowany jako różnica dopasowanie parametrów ϑ_0 z rzeczywistymi ϑ

$$\tilde{\vartheta} = \vartheta_0 - \vartheta,$$

gdzie

$$\vartheta_0 = \begin{bmatrix} m_0 \\ I_0 \\ \mu_{s0}m_0 \\ \mu_{l0}m_0 \end{bmatrix}.$$

Opracowując prawo sterowania dla dynamiki obiektu należy zdefiniować nowy błąd prędkości \tilde{u}

$$\tilde{u} = u_d - u, \quad (3.40)$$

a równania różniczkowe 3.5 uwzględniające błędy \tilde{z} oraz \tilde{u} przekształcone zostaną do postaci (Dixon [7])

$$\begin{aligned}
\dot{w} &= -k_1 w + u_a^T J \tilde{z} + \tilde{u}^T J z \\
\dot{\tilde{z}} &= -k_2 \tilde{z} + w J u_a + \tilde{u}.
\end{aligned} \quad (3.41)$$

Badając stabilność prawa sterowania uwzględniającego dynamikę robota ponownie wykorzystano drugą metodę Lapunowa (opis w dodatku D) [17]. Autorzy [7] zaproponowali, aby rachunek energetyczny został zapisany w formie

$$V_2(w, \tilde{z}, \tilde{u}) = \frac{1}{2}w^2 + \frac{1}{2}\tilde{z}^T \tilde{z} + \frac{1}{2}\tilde{u}^T \bar{M} \tilde{u}.$$

Rozważając zachowanie pochodnej można zapisać, że

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = w\dot{w} + \frac{1}{2}\tilde{z}^T \dot{\tilde{z}} + \tilde{u}^T \bar{M} \dot{\tilde{u}} + \frac{1}{2}\tilde{u}^T \dot{\bar{M}} \tilde{u}.$$

Wyliczając postać wyrażenia $\dot{\bar{M}}\tilde{u}$ z równań (3.31), (3.41), a następnie (3.29), otrzymano

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = \dot{V}_1(w, \tilde{z}) + \tilde{u}^T \left(wJz + \tilde{z} + \bar{M}\dot{u}_d + \bar{C}u + \bar{R} - \bar{B}\tau + \frac{1}{2}\dot{\bar{M}}\tilde{u} \right).$$

Jasnym jest, że $\dot{\bar{M}} = \bar{C}^T + \bar{C}$

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} + \tilde{u}^T \left(wJz + \tilde{z} + \bar{M}\dot{u}_d + \bar{C}u + \bar{R} - \bar{B}\tau \right) + \frac{1}{2} \tilde{u}^T (\bar{C}^T + \bar{C}) \tilde{u}.$$

Można więc do prawej strony równania dodać oraz odjąć wartość $\frac{1}{2} \tilde{u}^T \bar{C} \tilde{u}$, co spowoduje, że równanie przyjmie postać

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} + \tilde{u}^T \left(wJz + \tilde{z} + \bar{M}\dot{u}_d + \bar{C}u_d + \bar{R} - \bar{B}\tau \right) + \frac{1}{2} \tilde{u}^T (\bar{C}^T - \bar{C}) \tilde{u}.$$

Macierz $\bar{C}^T - \bar{C}$ jest skośnie symetryczna, bo

$$\tilde{u}^T (\bar{C}^T - \bar{C}) \tilde{u} = 0,$$

więc równanie przyjmie postać

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} + \tilde{u}^T \left(wJz + \tilde{z} + \bar{M}\dot{u}_d + \bar{C}u_d + \bar{R} - \bar{B}\tau \right). \quad (3.42)$$

Stosując wzory (3.37) oraz (3.42) równanie można uprościć do postaci

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} + \tilde{u}^T \left(wJz + \tilde{z} + Y_d \vartheta - \bar{B}\tau \right). \quad (3.43)$$

Aby uodpornić prawo sterowania na niepewność parametryczną $\tilde{\vartheta}$ wprowadza się dodatkowe sterowanie wejściowe τ_a . Na podstawie (3.42) proponowane sterowanie to

$$\tau = \bar{B}^{-1} (wJz + \tilde{z} + Y_d \vartheta_0 + \tau_a + k_3 \tilde{u}), \quad (3.44)$$

a powołując się na pracę Spong'a oraz Dixon'a [18, 7] τ_a można wyrazić jako

$$\tau_a = Y_d \frac{\rho^2 Y_d^T \tilde{u}}{\|Y_d^T \tilde{u}\| \rho + \epsilon_2}, \quad (3.45)$$

gdzie ρ spełnia nierówność $\|\tilde{\vartheta}\| \leq \rho$, a ϵ_2 jest stałą, dodatnią liczbą, która może być ustawiona jako arbitralnie mała [18, 7].

Wniosek 2 *Jasnym jest, że funkcja τ_a jest zawsze różniczkowalna w czasie, więc $\|Y_d^T \tilde{u}\|$. Ta zależność jest inna od typowego sterowania odpornego zaproponowanego u Spong'a [18]. Dzięki temu ograniczono drgania sygnału sterującego. Właściwość ta jest ważna, gdyż oferuje możliwość zastosowania klasycznego podejścia całkowania wstecznego. Można zapisać, że*

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} - \tilde{u}^T k_3 \tilde{u} + (Y_d^T \tilde{u})^T \left(\tilde{\vartheta} - \frac{\rho^2 Y_d^T \tilde{u}}{\|Y_d^T \tilde{u}\| \rho + \epsilon_2} \right).$$

Można udowodnić, że

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) \leq -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} - \tilde{u}^T k_3 \tilde{u} + \frac{\|Y_d^T \tilde{u}\| \rho \epsilon_2}{\|Y_d^T \tilde{u}\| \rho + \epsilon_2},$$

a wtedy doprowadzić do nierówności

$$0 \leq \frac{\|Y_d^T \tilde{u}\| \rho \epsilon_2}{\|Y_d^T \tilde{u}\| \rho + \epsilon_2} \leq \epsilon_2,$$

aby finalnie zapisać, że

$$\dot{V}_2(w, \tilde{z}, \tilde{u}) = -k_1 w^2 - k_2 \tilde{z}^T \tilde{z} - \tilde{u}^T k_3 \tilde{u} + \epsilon_2.$$

3.5 Testy symulacyjne

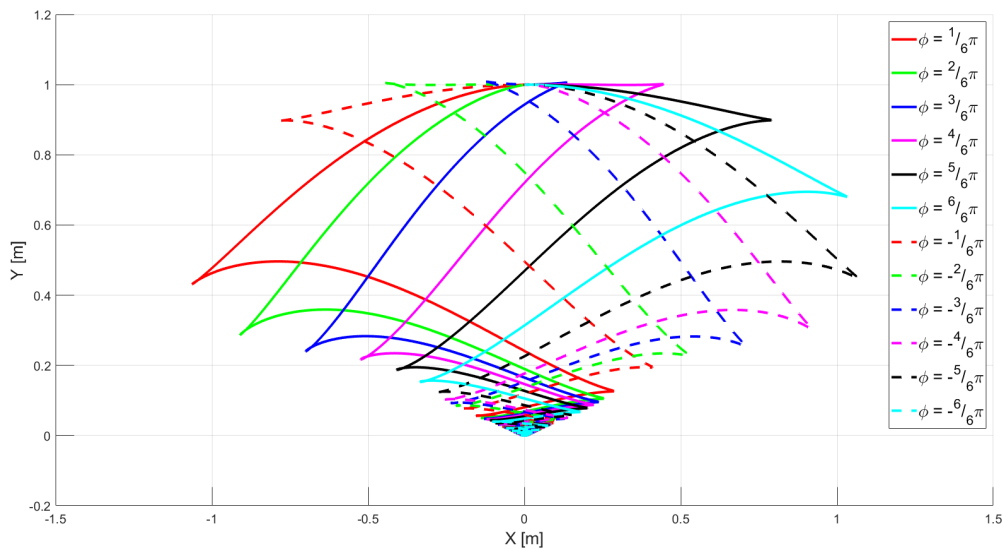
Testy symulacyjne przeprowadzone zostały w środowisku **MATLAB**. Zrealizowano dwa podstawowe zadania:

- stabilizacji – dojazd do punktu,
- śledzenia trajektorii.

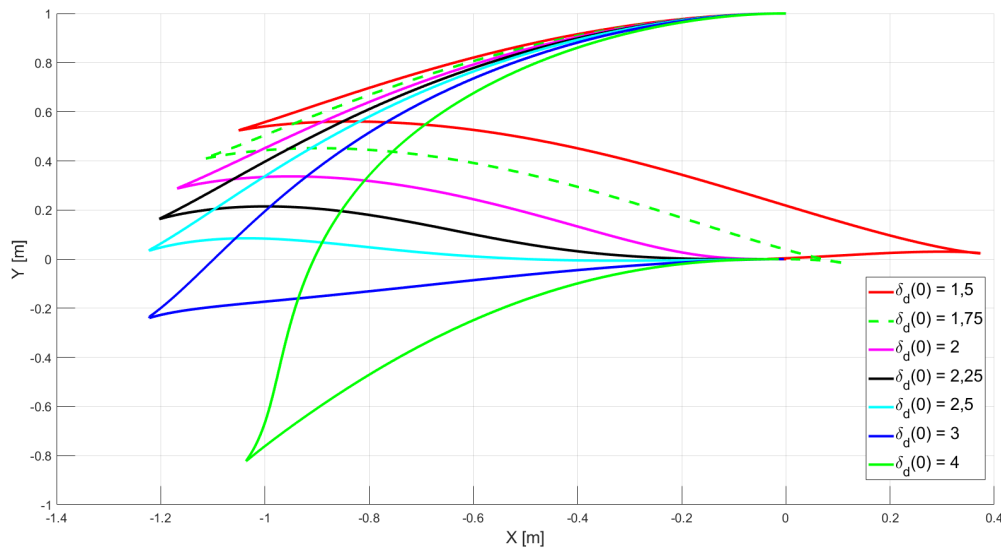
Model jest tak skonstruowany, że potrafi realizować oba zadania jednocześnie.

3.5.1 Wpływ ϕ oraz δ_d na realizację ruchu

Wiadomym jest, że niektóre współczynniki wpływają bardziej, inne mniej, na zachowanie modelu. Parametry ϕ oraz δ_d oscylatora (wzór (3.30)) znacząco wpływają na kształt drogi pokonanej przez robota, a tym samym czas realizacji zadania. Zmienna ϕ definiuje fazę oscylatora, natomiast δ_d definiuje tłumienie dryfu robota, spowodowanego ślizgiem. Badania symulacyjne wykazały, że δ_d ma bardzo duży wpływ na czas obliczeń, a jej wartość określa dokładność obliczeń, a co za tym idzie - dokładność realizacji zadania – gdy parametr ten jest mały, dokładność obliczeń jest większa, natomiast obliczenia trwają znacznie dłużej i zupełnie odwrotnie jest dla dużej wartości tej zmiennej. Rysunki 3.2 oraz 3.3 pokazują jak bardzo oba parametry wpływają na ruch robota na płaszczyźnie XY . Badanie wpływu ϕ oraz δ_d realizowane zostało dla zadania stabilizacji w punkcie $q_r = [0 \ 0 \ 0]^T$ przy założeniu, że robot startuje z punktu $q_0 = [0 \ 1 \ 0]^T$.



Rysunek 3.2: Wpływ ϕ na trajektorię pokonaną przez robota



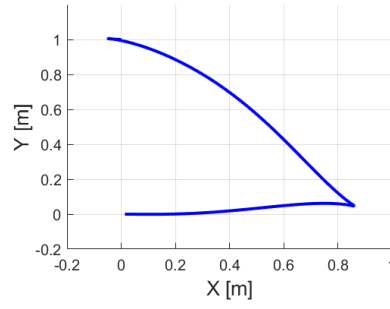
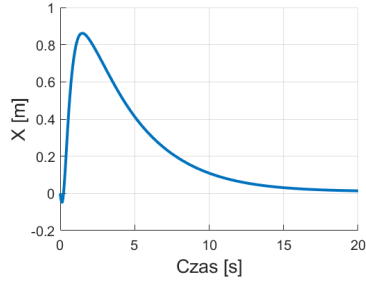
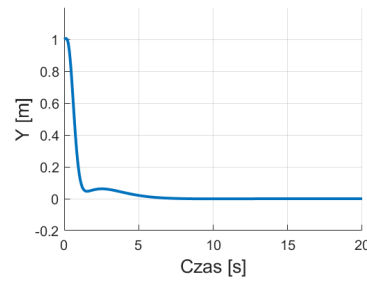
Rysunek 3.3: Wpływ δ_d na trajektorię pokonaną przez robota

3.5.2 Zadanie stabilizacji

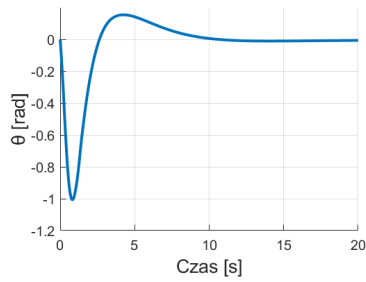
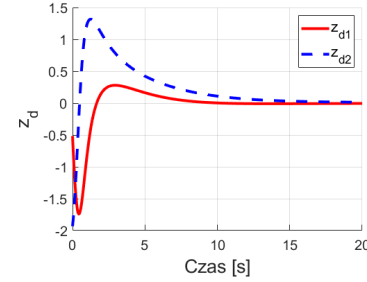
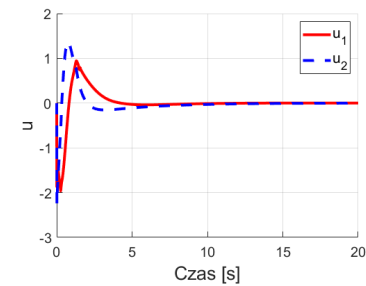
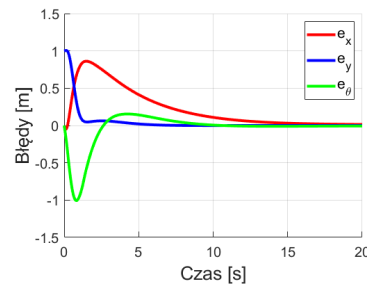
Prawo sterowania jest tak zdefiniowane, że algorytm stabilizuje robota w punkcie referencyjnym $q_r(x_r, y_r, \theta_r) = [0 \ 0 \ 0]^T$. W badaniach przyjęto, że $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ \theta_0]^T$, przy czym $(\theta_0 = \{0, -\pi\})$, czyli jedno z najtrudniejszych zadań, gdyż robot nie potrafi poruszać się w kierunku poprzecznym, a żądamy, by przemieścił się wzdłuż osi Y . Zgodnie z założeniami algorytmu, podczas rozwiązywania tego zadania części sterownika, w których skład wchodzi prędkości referencyjne $\eta_{ref} = [v_{rx} \ \omega_r]^T$, nie były używane, gdyż zadanie stabilizacji charakteryzuje się tym, że prędkości referencyjne $\eta_{ref} = 0$ (bo $q_{ref} = const$).

Pierwsze zadanie stabilizacji w punkcie zrealizowano dla parametrów $k_1 = 0,5$, $\alpha_1 = 0,3$, $\delta_d(0) = 2$, $\phi = -\frac{7}{12}\pi$, $x_{ICR} = -0,015\text{ m}$, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ 0]^T$. Wykres (3.4a) pokazuje, że robot pokonał trasę zbliżoną do trajektorii z rysunku (3.2), gdy $\phi = -\frac{3}{6}\pi$. Ponadto, wygenerowano wykresy zmiany położenia w osi OX (rys. 3.4b) oraz OY (rys. 3.4c). Badano również wartości błędów 3.5d, które świadczyć mają o poprawności wykonania zadania, jeśli ich wartości asymptotycznie zbiegają do zera.

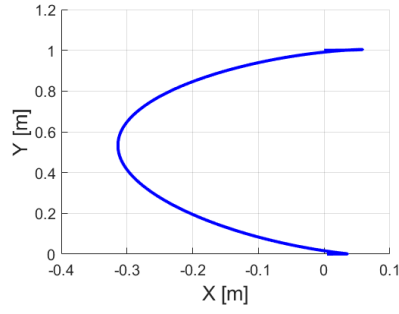
Drugie zadanie stabilizacji w punkcie zrealizowane dla parametrów $k_1 = 1$, $\alpha_1 = 0,5$, $\delta_d(0) = 3,2$, $\phi = \pi$, $x_{ICR} = -0,015\text{ m}$, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ -\pi]^T$. Tym razem zmieniono jedynie orientację początkową robota oraz przyjęto nieco inne parametry - bardziej optymalne dla tak postawionego zadania. Według danych z rysunku (3.6a) robot pokonał trasę wzdłuż łuku. Ponadto, wygenerowano wykresy zmiany położenia w osi OX (rys. 3.6b) oraz OY (rys. 3.6c). Badano również wartości błędów 3.7d, które świadczyć mają o poprawności wykonania zadania.

(a) Trajektoria robota na płaszczyźnie XY (b) Trajektoria $X(t)$ (c) Trajektoria $Y(t)$

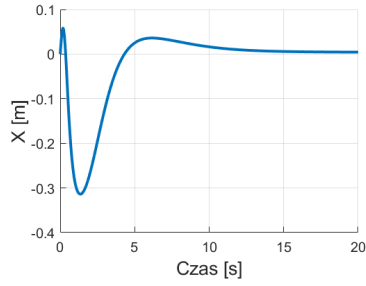
Rysunek 3.4: Stabilizacja robota na płaszczyźnie; $k_1 = 0,5$, $\alpha_1 = 0,3$, $\delta_d(0) = 2$, $\phi = -\frac{7}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ 0]^T$

(a) Zmiana orientacji $\theta(t)$ (b) Sterowanie $z_d(t)$ - postać normalna(c) Sterowanie $u(t)$ - postać normalna(d) Błędy stabilizacji $e(t)$

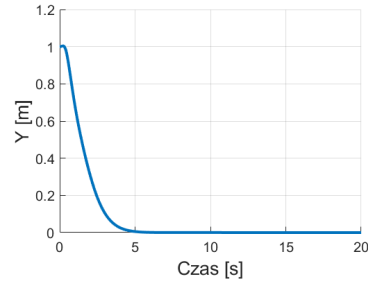
Rysunek 3.5: Wykresy zmiany orientacji, sterowań i błędów w zadaniu stabilizacji; $k_1 = 0,5$, $\alpha_1 = 0,3$, $\delta_d(0) = 2$, $\phi = -\frac{7}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ 0]^T$



(a) Trajektoria robota na płaszczyźnie XY

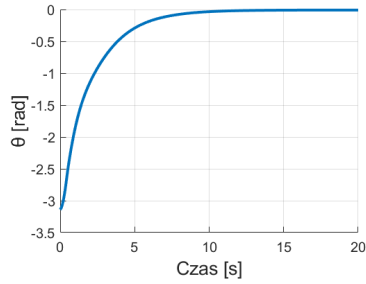


(b) Trajektoria $X(t)$

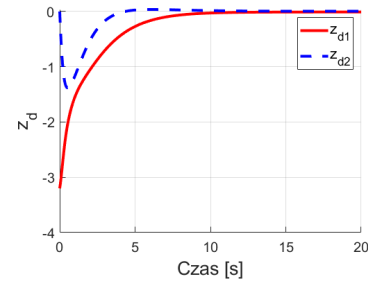


(c) Trajektoria $Y(t)$

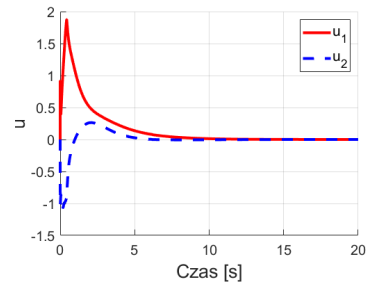
Rysunek 3.6: Stabilizacja robota na płaszczyźnie XY ; $k_1 = 1$, $\alpha_1 = 0,5$, $\delta_d(0) = 3,2$, $\phi = \pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ -\pi]^T$



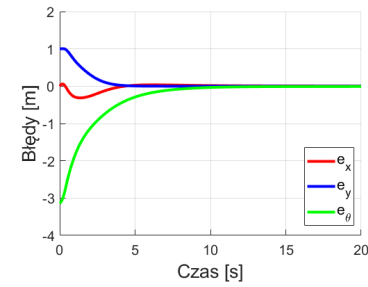
(a) Zmiana orientacji $\theta(t)$



(b) Sterowanie $z_d(t)$ - postać normalna



(c) Sterowanie $u(t)$ - postać normalna



(d) Błędy stabilizacji $e(t)$

Rysunek 3.7: Wykresy zmiany orientacji, sterowań i błędów w zadaniu stabilizacji; $k_1 = 1$, $\alpha_1 = 0,5$, $\delta_d(0) = 3,2$, $\phi = \pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ -\pi]^T$

3.5.3 Zadanie śledzenia trajektorii

W badaniach symulacyjnych rozważano śledzenie trajektorii opisanych zależnościami

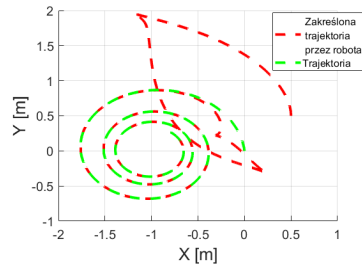
$$\begin{cases} x_r(t) = \left(\frac{1}{1+0,02t} \cos 0,2t - 1 \right) \text{ [m]} \\ y_r(t) = \frac{1}{1+0,02t} \sin 0,2t - 1 \text{ [m]} \end{cases}, \quad (3.46)$$

$$\begin{cases} x_r(t) = ((1 + 0,2 \sin 0,9t) \cos 0,15t - 1) \text{ [m]} \\ y_r(t) = (1 + 0,2 \sin 0,9t) \sin 0,15t \text{ [m]} \end{cases}. \quad (3.47)$$

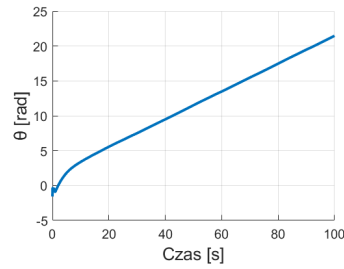
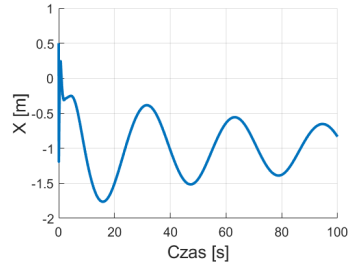
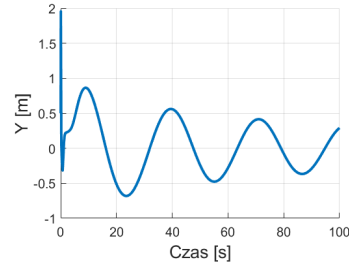
W przypadku tego zadania znaczącą rolę odgrywały części prawa sterowania (np. 3.45 we wzorze 3.44) zawierające w swym składzie prędkości referencyjne η_{ref} . Miały one za zadanie tłumić poślizg robota. Co ciekawe zmienna odpowiedzialna za tłumienie τ_a podczas realizacji testów została wyzerowana, gdyż wprowadzała do modelu dodatkowe częstotliwości, które zamiast tłumić sygnał, wzbudzały go. Nadal obserwuje się chwilowe wzbudzenie się robota równych odstępach czasu, co powoduje, że błędy śledzenia e_x , e_y oraz e_θ mogą chwilowo wzrastać. Nie jest to jednak duża uciążliwość, gdyż końcowe błędy są bliskie 0.001 m, a co za tym idzie można uznać, że robot poprawnie śledzi trajektorię.

Zadanie śledzenia dla parametrów $k_1 = 2$, $\alpha_1 = 0,3$, $\delta_d(0) = 3,2$, $\phi = -\frac{1}{12}\pi$, $x_{ICR} = -0,015 \text{ m}$, $q_0(x_0, y_0, \theta_0) = [0,5 \ 0,5 \ -\frac{\pi}{2}]^T$. Trajektoria zadana przypomina spiralę (rysunek 3.8a). Na rysunkach 3.8c oraz 3.8d widać, że robot na chwilę uciekał z trajektorii, ale szybko na nią wracał. Widać to też na wykresach 3.9b, 3.9c i 3.9d, gdzie w równych odstępach czasu następuje odchylenie od trajektorii (błędy śledzenia rosną – rys 3.9d), a sterowniki próbują korygować ruch robota (sterowanie z_d oraz u (rysunki 3.9b i 3.9c)). Dodatkowo widać, że momenty τ (rysunek 3.9a) na początku ruchu przyjmują bardzo dużą wartość, jednak po dotarciu robota na trajektorię, momenty spadają do bardzo niewielkich wartości (około 0,02 Nm).

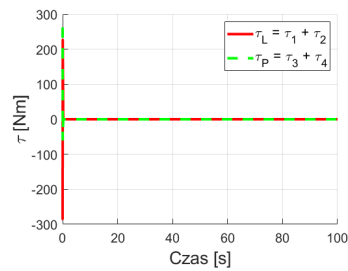
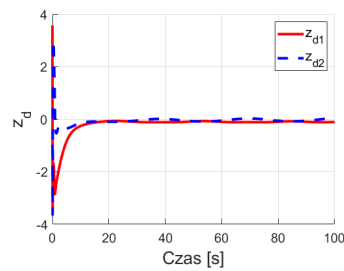
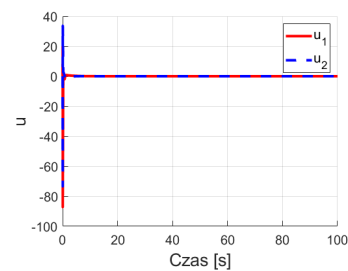
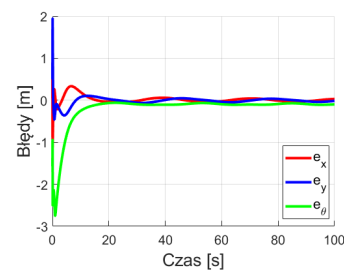
Zadanie śledzenia dla parametrów $k_3 = 80$, $k_4 = 50$, $\alpha_1 = 0,3$, $\delta_d(0) = 1,5$, $\phi = -\frac{2}{3}\pi$, $x_{ICR} = -0,015 \text{ m}$, $q_0(x_0, y_0, \theta_0) = [-1 \ 0 \ -\frac{\pi}{4}]^T$. Trajektoria zadana przypomina gwiazdę (rysunek 3.10a). Widać, że w przypadku tego zadania poślizg odgrywa duży wpływ na ruch robota (patrz wykresy błędów 3.11d), gdzie pomimo tego, że są one rzędu 0.01 m, to widać fluktuacje ruchu. Robot trzyma się zadanej trajektorii i śledzi ją, jednak z małymi błędami. Sterowania z_d oraz u (rysunki 3.11b oraz 3.11c) oscylują wokół zera, aby wysterować pojazd, ale nie osiągają wartości zera. Wykres 3.10b pokazuje, że następują cykliczne zmiany orientacji robota, przez co błąd tego stanu fluktuuje najbardziej. Momenty τ (rysunek 3.11a) na początku ruchu mają bardzo dużą wartość, jednak po dotarciu robota na trajektorię, spadają do niewielkich wartości (około 0,02 Nm).



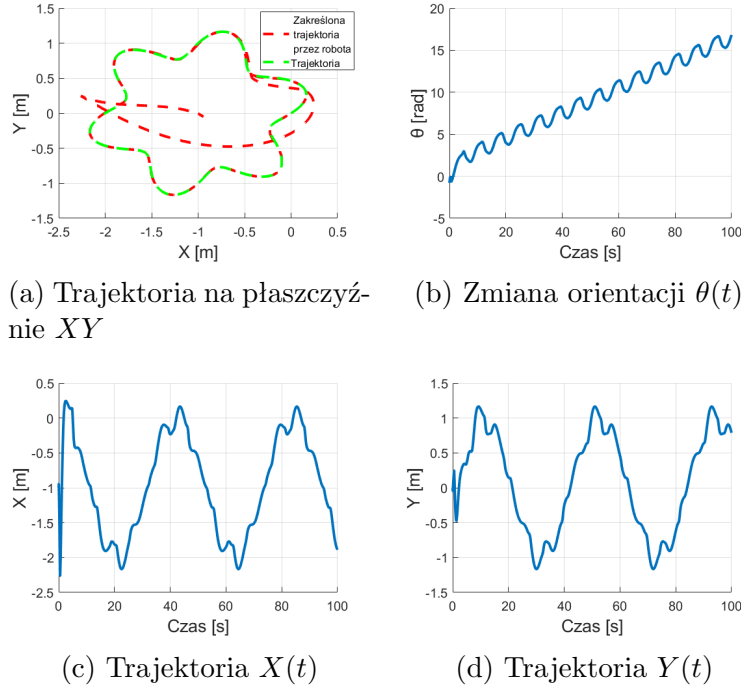
(a) Trajektoria na płaszczyźnie XY

(b) Zmiana orientacji $\theta(t)$ (c) Trajektoria $X(t)$ (d) Trajektoria $Y(t)$

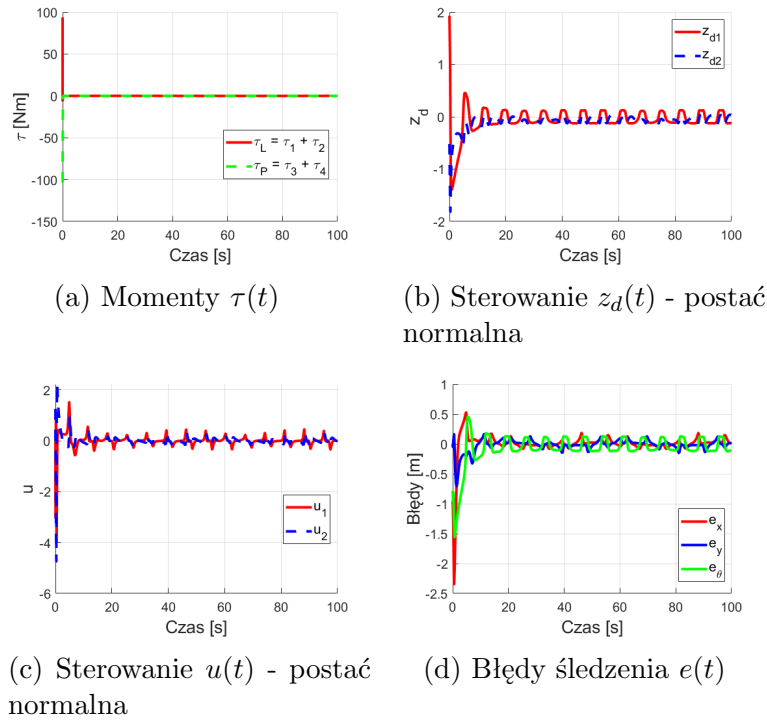
Rysunek 3.8: Śledzenie trajektorii (3.46) na płaszczyźnie XY; $k_1 = 2$, $\alpha_1 = 0, 3$, $\delta_d(0) = 3, 2$, $\phi = -\frac{1}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0.5 \ 0.5 \ -\frac{\pi}{2}]^T$

(a) Momenty $\tau(t)$ (b) Sterowanie $z_d(t)$ - postać normalna(c) Sterowanie $u(t)$ - postać normalna(d) Błędy śledzenia $e(t)$

Rysunek 3.9: Śledzenie trajektorii (3.46): zmiana orientacji, sterowanie, błędy; $k_1 = 2$, $\alpha_1 = 0, 3$, $\delta_d(0) = 3, 2$, $\phi = -\frac{1}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0.5 \ 0.5 \ -\frac{\pi}{2}]^T$



Rysunek 3.10: Śledzenie trajektorii (3.47) na płaszczyźnie XY ; $k_3 = 80$, $k_4 = 50$, $\alpha_1 = 0, 3$, $\delta_d(0) = 1, 5$, $\phi = -\frac{2}{3}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [-1 \ 0 \ -\frac{\pi}{4}]^T$



Rysunek 3.11: Śledzenie trajektorii (3.47): zmiana orientacji, sterowanie, błędy; $k_3 = 80$, $k_4 = 50$, $\alpha_1 = 0, 3$, $\delta_d(0) = 1, 5$, $\phi = -\frac{2}{3}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [-1 \ 0 \ -\frac{\pi}{4}]^T$

Rozdział 4

Podsumowanie

Zadaniem niniejszej pracy było zamodelowanie ruchu robotów poruszających się z poślizgiem. Modelowanie robotów samo w sobie jest zadaniem wymagającym zarówno matematycznie jak i implementacyjnie. Dodatkowo, uwzględnienie poślizgów w ruchu robota, czyli zniesienie pewnych ograniczeń nieholonomicznych, ma swoje konsekwencje w postaci dodatkowych trudności zarówno w procesie modelowania, jak i projektowania algorytmów sterowania dla takiego robota.

O ile wyprowadzenie modelu kinematyki czy dynamiki robota jest już samo w sobie zadaniem skomplikowanym z uwagi na konieczność opracowania i uwzględnienia w modelu konkretnych prędkości/sił reakcji od podłoża, to najtrudniejszym zadaniem okazuje się być opracowanie algorytmu sterowania dla takiego modelu. Powszechnym podejściem, z którego skorzystano w niniejszej pracy, jest znalezienie transformacji do postaci normalnej, rozwiązanie zadania sterowania w ramach postaci normalnej oraz zastosowania transformacji odwrotnej w celu powrotu do postaci pierwotnej modelu. Należy jednak pamiętać, że opracowanie i przeprowadzenie tej procedury wymaga talentu, intuicji i zaawansowanej wiedzy robotycznej. Badanie stabilności algorytmu sterowania również stanowi poważne wyzwanie. W tym przypadku często wykorzystuje się metodę Lapunowa, co z kolei wymaga doświadczenia w definiowaniu odpowiednich funkcji Lapunowa. To wszystko potęguje skalę trudności i podkreśla, jak istotne jest połączenie teorii z praktyką w obszarze sterowania robotami.

W procesie implementacji modelu ruchu robota oraz algorytmu sterowania wykorzystano platformę **MATLAB**, co okazało się być trafnym wyborem, gdyż program ten wspiera obliczenia na macierzach co znacznie uprościło zadanie implementacyjne. Ponadto, **MATLAB** świetnie sprawdził się podczas sporządzania wykresów/rysunków/animacji, co dodatkowo ułatwiło analizę i prezentację wyników.

Dla wyprowadzonego modelu kinematyki i dynamiki robota podlegającego działaniu poślizgów przeprowadzono szereg testów w warunkach idealnych (bez poślizgu) oraz w warunkach rzeczywistych (z poślizgiem). Zbadano wpływ różnych postaci funkcji sterujących na działanie robota. Badania wykazały, że tarcie ma kluczowy wpływ na ruch robota, często powodując jego zatrzymanie, gdy moment napędowy był niewystarczający do pokonania sił tarcia. W dalszej części, zaproponowano i zaimplementowano algorytm sterowania dzięki któremu policzono zadanie stabilizacji robota w punkcie oraz zadanie śledzenia trajektorii. Algorytm okazał się być bardzo wrażliwy na wartości wielu współczynników wchodzących w jego skład, co również wykazano w testach symulacyjnych. Pomimo wraź-

liwości algorytmu, udało się uzyskać zadowalające rezultaty w procesie realizacji zadania stabilizacji oraz śledzenia trajektorii.

W ramach przyszłych badań, w pierwszej kolejności należy przyrzeć się prawu sterowania, gdyż możliwe jest jego zoptymalizowanie. Warto również przebadąć wpływ każdego z parametrów algorytmu na zachowanie robota w zadaniu stabilizacji oraz śledzenia trajektorii.

Ponadto, obiecującym kierunkiem badań byłoby uzależnienie parametru x_{ICR} od czasu z uwagi na fakt, że podczas ruchu zmienna ta przemieszcza się wzdłuż osi poprzecznej robota. Wartość tego parametru jest kluczowa, gdyż zmienia się w zależności od odległości od centralnego punktu obrotu ICR oraz od prędkości robota. Ta dynamiczna zmienność może wpływać na zachowanie modelu, sprawiając, że reakcje robota będą się różnić w zależności od aktualnych warunków, w odróżnieniu od sytuacji, w której wartość tego parametru byłaby stała. To podkreśla, jak istotne jest uwzględnienie tej dynamicznej zmienności w analizach i projektowaniu sterowania. Proponowanym rozwiązaniem jest wyprowadzenie wzoru na $x_{ICR}(t)$ z zależności (2.11). Otrzymamy wówczas

$$x_{ICR}(t) = \frac{-v_y}{\omega}.$$

Problemem tego rozwiązania jest przypadek poruszania się po prostej, gdyż wtedy prędkość kątowna ω wynosi zero. Warto więc zaimplementować wersję odporną tego wzoru w postaci

$$x_{ICR}(t) = \frac{-v_y \omega}{\omega^2 + \varepsilon_{ICR}},$$

gdzie współczynnik ε_{ICR} musi mieć na tyle małą wartość, żeby nie wprowadzał zaburzenia. Proponowana wartość to $\varepsilon_{ICR} = 0,001$. Dla tego rozwiązania dobrze jest ponownie sprawdzić jak zmiany w czasie parametru x_{ICR} wpływają na ruch robota.

Podsumowując, modelowanie robotów poruszających się z poślizgiem jest zadaniem ciekawym, trudnym, ale i potrzebnym w celu jak najwierniejszego odwzorowania rzeczywistości. Człowiek ma problem z predykcją takiego układu, gdyż o ile perfekcyjnie potrafi przewidzieć zachowanie ruchu robota w warunkach idealnych, o tyle w warunkach rzeczywistych nie potrafi tego dobrze zrobić.

Cel pracy został osiągnięty: zaproponowano oraz zaimplementowano i przetestowano model ruchu robota poruszającego się pod wpływem poślizgów. Dodatkowo, zaproponowano i zaimplementowano algorytm sterowania oraz przetestowano jego działanie w procesie realizacji zadania stabilizacji oraz śledzenia trajektorii.

Dodatek A

Modelowanie tarcia

W celu jak najwierniejszego oddania rzeczywistości modelu należy uwzględnić w nim siły oporu. Najbardziej podstawowymi są siły tarcia. Pomimo tego, że są one jednymi z podstawowych sił, to ich modelowanie jest stosunkowo trudne, gdyż siły te są skomplikowane ze względu na

- zależność od wielu zmiennych,
- nieliniowy charakter.

Niech wektor F_{si} wynika z momentu oporu toczenia τ_{ri} , a wektor F_{li} odpowiada za boczną siłę bierną. Korzystając z obu siły reaktywne można uznać za siły tarcia. Ze względu na trudność w modelowaniu, rozważa się jedynie przybliżone tarcie F_f opisujące siły jako superpozycję tarcia Coulomb'a oraz tarcia lepkiego [4]

$$F_f(\sigma) = \mu_c N \operatorname{sgn}(\sigma) + \mu_v \sigma, \quad (\text{A.1})$$

gdzie:

- σ – prędkość liniowa,
- N – siła prostopadła do powierzchni,
- μ_c – współczynnik Coulomb'a,
- μ_v – współczynnik tarcia lepkiego.

Z powyższego wzoru wynika, że podstawową różnicą pomiędzy tarcie Coulomb'a, a tarcie lepkiem jest zależność od prędkości. W tym pierwszym prędkość określa jedynie zwrot działania siły w stosunku do kierunku poruszania się ciała, natomiast tarcie lepkie wprost zależy od prędkości, co oznacza, że im szybciej porusza się pojazd, tym większy opór jest do pokonania. Co więcej tarcie Coulomb'owskie opisuje opór napotykaną przez obiekt na powierzchni suchej, podczas gdy tarcie lepkie nakreśla opór jaki ciało musi pokonać w ośrodku, który jest cieczą lub gazem. Ciekawym jest, że tarcie lepkie zależy również od powierzchni styku obiektu z ośrodkiem lepkiem. Im jest ono większe, tym większy opór występuje. Tarcie Coulomb'a można podzielić na tarcie

- statyczne – opór jaki należy pokonać, aby wprowadzić ciało w ruch,
- kinetyczny – opór jaki musi pokonywać ciało, aby pozostać w ruchu (zawsze mniejsze od tarcia statycznego),

podczas gdy tarcie lepkie określa opór na styku dwóch powierzchni ze sobą, z czego jedna musi być cieczą bądź gazem. Przez fakt iż w badanym modelu zakłada się, że powierzchnia styku koła z podłożem jest punktowa, tarcie lepkie nie występuje, co znacznie upraszcza modelowanie obiektu.

Prędkość liniowa modelu σ podczas poślizgu bocznego jest stosunkowo niska, więc obowiązuje zależność $\mu_c N \gg |\mu_v \sigma|$, co pozwala na zaniechanie drugiego członu nierówności w celu uproszczenia modelu. Ponadto funkcja opisana wzorem (A.1) nie jest gładka, gdy prędkość σ jest równa zero, ze względu na charakter funkcji $sgn(\sigma)$. Przy prędkości σ równej zero oczywistym jest, że funkcja jest nieróżniczkowalna w czasie. Przyjęto więc przybliżenie funkcji $sgn(\sigma)$ funkcją

$$\widehat{sgn}(\sigma) = \frac{2}{\pi} \arctan(k_s \sigma), \quad (\text{A.2})$$

gdzie $k_s \gg 1$ w celu uzyskania funkcji jak najbardziej zbliżonej do oryginalnej funkcji signum tak, że

$$\lim_{k_s \rightarrow \infty} \frac{2}{\pi} \arctan(k_s \sigma) = sgn(x).$$

Opierając się na powyższych danych siły tarcia dla jednego koła można przedstawić następująco

$$F_{li} = \mu_{lci} m g \widehat{sgn}(v_{yi}), \quad (\text{A.3})$$

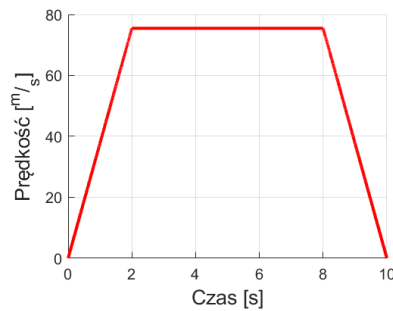
$$F_{si} = \mu_{sci} m g \widehat{sgn}(v_{xi}), \quad (\text{A.4})$$

gdzie μ_{lci} oraz μ_{sci} są współczynnikami tarcia odpowiednio w kierunku poprzecznym i wzdłużnym.

Dodatek B

Trapezoidalny profil prędkości

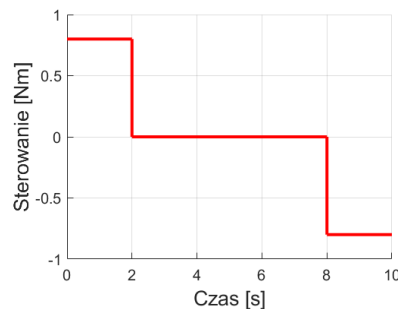
Prostym badaniem, jakie można przeprowadzić na obiekcie, jest zadanie trapezoidalnego profilu prędkości robota. Taki profil prędkości charakteryzuje się liniowo narastającymi, w próbie czasu, prędkościami modelu η (wzór (2.16)). Kolejno ważne jest, aby utrzymać stałe wartości prędkości w kolejnej chwili czasowej, a następnie liniowo zmniejszać wartości prędkości aż do momentu kiedy $\eta = [0 \ 0]^T$.



Rysunek B.1: Trapezoidalny profil prędkości

Dzięki takiemu zabiegowi, nie występuje gwałtowne podanie prędkości na obiekt, co lepiej odzwierciedla początkową i końcową fazę ruchu fizycznego robota. Gwałtowne podanie napięcia na robota spowodować mogłoby buksowanie jego kół lub też charakterystyczne szarpnięcie podczas fazy rozruchu czy też gwałtowne hamowanie w końcowej fazie ruchu.

W celu uzyskania takiego profilu prędkości badanego robota (rysunek B.1) należy zadać sterowanie τ (wzór (2.32)) takie, aby otrzymać całkowny profil przyspieszeń (przedstawiony na rysunku B.2), który pozwoli na uzyskanie trapezoidalnego profilu dla η .



Rysunek B.2: Wykres przyspieszeń τ generujący trapezoidalny profil prędkości

Dodatek C

Szereg Fouriera

W prowadzeniu badań nad zachowaniem modeli, często stosuje się sterowania w postaci trygonometrycznego szeregu Fouriera. Jest to jedna z najpowszechniej stosowanych ortogonalnych funkcji sterujących, tuż obok wielomianów Czebyszewa, Legendre’a, Hemite’a, czy funkcji Haara. Stosowanie funkcji ortogonalnych pozwala na przedstawienie nieskończonej wymiarowej endogenicznej przestrzeni konfiguracyjnej w jej skończonej wymiarowej ortogonalnej reprezentacji [9]. Szeregi Fouriera są dosyć proste w interpretacji i zachowują się zgodnie z intuicją.

Szereg Fouriera jest to suma wartości stałej oraz sinusów bądź cosinusów przemnożonych przez odpowiednie mnożniki (amplitudę) określające ich wpływ na przebieg finalny. Każdy z kolejnych mnożników ma coraz mniejszy wpływ na sumę. Szereg wyraża się wzorem

$$F(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(n\omega t) + b_n \sin(n\omega t)).$$

Dla badanego modelu wykorzystano najprostszy z możliwych szeregów. Celem jest uzyskanie przebiegu sinusoidalnego dla prędkości pomocniczych η robota. W tej sytuacji momenty należy zadać sterowanie w postaci różniczki prędkości. Należy zatem przyjąć, że wartości momentów, będą miały charakter

$$\tau(t) = \cos \omega t.$$

W celu uzyskania prędkości należy scałkować funkcję $\tau(t)$ i wówczas przebiegi prędkości pomocniczych przyjmą postać

$$\eta(t) = \int \cos \omega t dt = \frac{1}{\omega} \sin \omega t$$

Dodatek D

Metody Lapunowa

Istnieje wiele metod badania stabilności układów robotycznych. Aleksandr Lapunow opracował dwie metody, które umożliwiają badanie stabilności układów nieliniowych.

D.1 I metoda Lapunowa

Dotyczy tych układów, które mogą być zlinearyzowane wokół punktu równowagi. Bada się je następnie tak jak układy liniowe. Mówi ona o interpretacji rozmieszczenia wartości własnych równania charakterystycznego na płaszczyźnie zespolonej. Rozwiązania jakie można otrzymać stosując tą metodę to

- układ stabilny asymptotycznie – wszystkie wartości własne leżą w lewej części płaszczyzny zespolonej. Ich część rzeczywista jest ujemna,
- układ niestabilny – co najmniej jedna wartość własna równania charakterystycznego ma dodatnią część rzeczywistą,
- układ stabilny – jeśli wartości własne leżą w lewej części płaszczyzny zespolonej lub leżą na osi urojonej. W takim przypadku metoda nie rozstrzyga stabilności asymptotycznej, na której nam najbardziej zależy.

D.2 II metoda Lapunowa

Jak widać, I metoda jest stosunkowo prosta, ale nie zawsze daje efekt zadowalający, dlatego zaproponowana została druga metoda, która zawsze rozstrzyga o stabilności układu. Ma ona charakter energetyczny, gdzie dąży się do osiągnięcia punktu równowagi w sensie energetycznym w punkcie równowagi q_e . Dla potrzeb tej metody tworzona jest funkcji $V(\cdot)$, zwana funkcją Lapunowa, a dla niej formułuje się warunki

1. $V(q_e) = 0$ – wartość funkcji w punkcie równowagi w sensie energetycznym musi być równa 0,
2. $V(q) > 0 \quad \forall q \neq q_e$ – funkcja jest dodatnia dla wszystkich punktów różnych od punktu równowagi,
3. $\dot{V}(q) \leq 0$ – określenie zachowania pochodnej funkcji Lapunowa. Ten warunek określa w pełni stabilność układu.

Dodatkowo warto podkreślić, że samo zdefiniowanie funkcji Lapunowa implikuje stabilność układu. Jeżeli w trzecim warunku jest nierówność ostra dla $q \neq q_e$, to układ jest asymptotycznie stabilny.

Literatura

- [1] M. M. Ahmed, S. A. Moiz, i in. Steering algorithm for drift free control systems. *Communications in Applied Analysis*, 11(4), 2007.
- [2] R. W. Brockett, i in. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1):181–191, 1983.
- [3] P. Brunovský. A classification of linear controllable systems. *Kybernetika*, 6(3):173–188, 1970.
- [4] L. Caracciolo, A. De Luca, S. Iannitti. Trajectory tracking control of a four-wheel differentially driven mobile robot. *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, wolumen 4, strony 2632–2638. IEEE, 1999.
- [5] B. d’Andréa Novel, G. Campion, G. Bastin. Control of nonholonomic wheeled mobile robots by state feedback linearization. *The International journal of robotics research*, 14(6):543–559, 1995.
- [6] W. E. Dixon, A. Behal, D. M. Dawson, S. P. Nagarkatti. *Nonlinear control of engineering systems: a Lyapunov-based approach*. Springer Science & Business Media, 2003.
- [7] W. E. Dixon, D. M. Dawson, E. Zergeroglu, A. Behal. *Nonlinear control of wheeled mobile robots*, wolumen 175. Springer London, 2001.
- [8] T. Fukao, H. Nakagawa, N. Adachi. Adaptive tracking control of a nonholonomic mobile robot. *IEEE transactions on Robotics and Automation*, 16(5):609–615, 2000.
- [9] D. Gidel. Praca dyplomowa inżynierska – różne parametryzacje funkcji sterujących platformy rex: nieliniowy model poślizgów poprzecznych kół. 2017.
- [10] K. Kozłowski, D. Pazderski. Modeling and control of a 4-wheel skid-steering mobile robot. *International journal of applied mathematics and computer science*, 14(4):477–496, 2004.
- [11] A. D. Luca, G. Oriolo. Modelling and control of nonholonomic mechanical systems. *Kinematics and dynamics of multi-body systems*, strony 277–342. Springer, 1995.
- [12] Murray, R. M, Sastry, S. Shankar. Nonholonomic motion planning: Steering using sinusoids. *IEEE transactions on Automatic Control*, 38(5):700–716, 1993.
- [13] T. P. Nascimento, C. E. Dórea, L. M. G. Gonçalves. Nonholonomic mobile robots’ trajectory tracking model predictive control: a survey. *Robotica*, 36(5):676–696, 2018.

- [14] H. Pacejka. Tyre and vehicle dynamics analysis. 2002.
- [15] M. Roozegar, M. J. Mahjoob. Modelling and control of a non-holonomic pendulum-driven spherical robot moving on an inclined plane: simulation and experimental results. *IET Control Theory & Applications*, 11(4):541–549, 2017.
- [16] L. A. Şandru, M.-F. Crainic, S. Preitl, V. Dolga. Path calculation of 4 dof remote vehicle for educational purpose. *2015 IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMi)*, strony 89–92. IEEE, 2015.
- [17] Shankar, S. Sastry. Lyapunov stability theory. *Nonlinear Systems: Analysis, Stability, and Control*, strony 182–234, 1999.
- [18] M. W. Spong. On the robust control of robot manipulators. *IEEE Transactions on automatic control*, 37(11):1782–1786, 1992.
- [19] E. Velenis, D. Katzourakis, E. Frazzoli, P. Tsiotras, R. Happee. Steady-state drifting stabilization of rwd vehicles. *Control Engineering Practice*, 19(11):1363–1376, 2011.
- [20] H. Wang, J. Zhang, J. Yi, D. Song, S. Jayasuriya, J. Liu. Modeling and motion stability analysis of skid-steered mobile robots. *2009 IEEE International Conference on Robotics and Automation*, strony 4112–4117. IEEE, 2009.
- [21] H. Wang, L. Zhao, Y. Yang, Y. Yan, J. Liu. Modeling and analysis of wheel mobile robot for high-speed precise drift. *Proceedings of the 31st Chinese Control Conference*, strony 5064–5069. IEEE, 2012.
- [22] J. Yi, D. Song, J. Zhang, Z. Goodwin. Adaptive trajectory tracking control of skid-steered mobile robots. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, strony 2605–2610. IEEE, 2007.
- [23] K. Zadarnowska, K. Tchoń. Modeling and motion planning of wheeled mobile robots subject to slipping. *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, strony 78–83. IEEE, 2015.
- [24] I. Zubov, I. Afanasyev, A. Gabdullin, R. Mustafin, I. Shimchik. Autonomous drifting control in 3d car racing simulator. *2018 International Conference on Intelligent Systems (IS)*, strony 235–241. IEEE, 2018.

Spis rysunków

2.1	Schemat robota ([10, str. 2])	6
2.2	Rozkład prędkości dla i-tego koła (Kozłowski, Pazderski [10, str. 3])	7
2.3	Rozkład prędkości w robocie (Kozłowski [10, str. 3])	8
2.4	Rozkład sił dla i-tego koła (Kozłowski, Pazderski [10, str. 4])	11
2.5	Warunki początkowe W_1 (2.43): wykresy położenia robota na płaszczyźnie	18
2.6	Warunki początkowe W_1 (2.43): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	18
2.7	Warunki początkowe W_2 (2.44): wykresy położenia robota na płaszczyźnie	19
2.8	Warunki początkowe W_2 (2.44): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	19
2.9	Warunki początkowe W_3 (2.45): wykresy położenia robota na płaszczyźnie	20
2.10	Warunki początkowe W_3 (2.45): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	20
2.11	Warunki początkowe W_4 (2.46): wykresy położenia robota na płaszczyźnie	21
2.12	Warunki początkowe W_4 (2.46): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	21
2.13	Warunki początkowe W_1 (2.43): wykresy położenia robota na płaszczyźnie	23
2.14	Warunki początkowe W_1 (2.43): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	23
2.15	Warunki początkowe W_2 (2.44): wykresy położenia robota na płaszczyźnie	24
2.16	Warunki początkowe W_2 (2.44): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	24
2.17	Warunki początkowe W_3 (2.45): wykresy położenia robota na płaszczyźnie	25
2.18	Warunki początkowe W_3 (2.45): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	25
2.19	Warunki początkowe W_4 (2.46): wykresy położenia robota na płaszczyźnie	26
2.20	Warunki początkowe W_4 (2.46): wykresy zmian parametrów θ , x_{ICR} , η oraz τ w czasie	26
2.21	Warunki początkowe W_3 (2.45): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości	29
2.22	Warunki początkowe W_3 (2.45): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości	29
2.23	Warunki początkowe W_3 (2.45): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości	30
2.24	Warunki początkowe W_3 (2.45): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości	30
2.25	Warunki początkowe W_4 (2.46): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości	31

2.26	Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – trapezoidalny profil prędkości . . .	31
2.27	Warunki początkowe W_4 (2.46): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości	32
2.28	Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – trapezoidalny profil prędkości . . .	32
2.29	Warunków początkowe W_3 (2.45): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,01$ m – szereg Fouriera	34
2.30	Warunki początkowe W_3 (2.45): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – szereg Fouriera	34
2.31	Warunków początkowe W_3 (2.45): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,03$ m – szereg Fouriera	35
2.32	Warunki początkowe W_3 (2.45): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – szereg Fouriera	35
2.33	Warunków początkowe W_4 (2.46): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,01$ m – szereg Fouriera	36
2.34	Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,01$ m – szereg Fouriera	36
2.35	Warunków początkowe W_4 (2.46): wykresy położenia robota rzeczywistego na płaszczyźnie; $x_{ICR} = 0,03$ m – szereg Fouriera	37
2.36	Warunki początkowe W_4 (2.46): wykresy prędkości pomocniczych η oraz zadanego momentu τ ; $x_{ICR} = 0,03$ m – szereg Fouriera	37
2.37	Warunki początkowe $W_{1.1}$ (2.48): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości	39
2.38	Warunki początkowe $W_{1.1}$ (2.48): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości	39
2.39	Warunki początkowe $W_{1.2}$ (2.49): wykresy położenia rzeczywistego robota na płaszczyźnie; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości	40
2.40	Warunki początkowe $W_{1.2}$ (2.49): prędkości pomocnicze η oraz zadany moment τ ; $x_{ICR} = 0,025$ m – trapezoidalny profil prędkości	40
3.1	Interpretacja geometryczna zmiennej z_2	43
3.2	Wpływ ϕ na trajektorię pokonaną przez robota	51
3.3	Wpływ δ_d na trajektorię pokonaną przez robota	52
3.4	Stabilizacja robota na płaszczyźnie; $k_1 = 0,5$, $\alpha_1 = 0,3$, $\delta_d(0) = 2$, $\phi = -\frac{7}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ 0]^T$	53
3.5	Wykresy zmiany orientacji, sterowań i błędów w zadaniu stabilizacji; $k_1 = 0,5$, $\alpha_1 = 0,3$, $\delta_d(0) = 2$, $\phi = -\frac{7}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ 0]^T$	53
3.6	Stabilizacja robota na płaszczyźnie XY ; $k_1 = 1$, $\alpha_1 = 0,5$, $\delta_d(0) = 3,2$, $\phi = \pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ -\pi]^T$	54
3.7	Wykresy zmiany orientacji, sterowań i błędów w zadaniu stabilizacji; $k_1 = 1$, $\alpha_1 = 0,5$, $\delta_d(0) = 3,2$, $\phi = \pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0 \ 1 \ -\pi]^T$	54
3.8	Śledzenie trajektorii (3.46) na płaszczyźnie XY ; $k_1 = 2$, $\alpha_1 = 0,3$, $\delta_d(0) = 3,2$, $\phi = -\frac{1}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0.5 \ 0.5 \ -\frac{\pi}{2}]^T$	56
3.9	Śledzenie trajektorii (3.46): zmiana orientacji, sterowanie, błędy; $k_1 = 2$, $\alpha_1 = 0,3$, $\delta_d(0) = 3,2$, $\phi = -\frac{1}{12}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [0.5 \ 0.5 \ -\frac{\pi}{2}]^T$	56

3.10	Śledzenie trajektorii (3.47) na płaszczyźnie XY; $k_3 = 80$, $k_4 = 50$, $\alpha_1 = 0,3$, $\delta_d(0) = 1,5$, $\phi = -\frac{2}{3}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [-1 \ 0 \ -\frac{\pi}{4}]^T$	57
3.11	Śledzenie trajektorii (3.47): zmiana orientacji, sterowanie, błędy; $k_3 =$ 80 , $k_4 = 50$, $\alpha_1 = 0,3$, $\delta_d(0) = 1,5$, $\phi = -\frac{2}{3}\pi$, $x_{ICR} = -0,015$ m, $q_0(x_0, y_0, \theta_0) = [-1 \ 0 \ -\frac{\pi}{4}]^T$	57
B.1	Trapezoidalny profil prędkości	63
B.2	Wykres przyspieszeń τ generujący trapezoidalny profil prędkości	63

Spis tabel

2.1	Tabela zbiorcza wartości parametrów	15
-----	---	----