

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

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

**PRACA DYPLOMOWA
INŻYNIERSKA**

Sterowanie monocyklem realizujące różne
zadania

Control of unicycle for different tasks

AUTOR:
Aleksandra Nycz

PROWADZĄCY PRACĘ:
dr hab. inż. Alicja Mazur, prof. PWr

OCENA PRACY:

*Moim Rodzicom oraz panom
B. i S.; za wiarę we mnie i wspie-
ranie każdego kolejnego dnia.
Dziękuję.*

Spis treści

1	Wprowadzenie	3
1.1	Cel pracy	4
1.2	Przegląd zawartości pracy	4
2	Modelowanie monocykla	5
2.1	Opis układu	5
2.2	Ograniczenia nieholonomiczne	6
2.2.1	Współrzędne uogólnione	6
2.2.2	Współrzędne pomocnicze	7
2.3	Dynamika monocykla	8
2.3.1	Współrzędne uogólnione	8
2.3.2	Współrzędne pomocnicze	10
2.4	Układ kaskadowy	10
3	Sformułowanie problemu sterowania	11
3.1	Sterownik dynamiczny	12
3.2	Zadania sterowania	13
4	Algorytm śledzenia trajektorii dla monocykla	15
4.1	Model obiektu	15
4.2	Model generatora	16
4.3	Błędy śledzenia	16
4.4	Sterownik kinematyczny	16
4.5	Algorytm Samsona i Ait-Abderrahima	16
4.6	Badania symulacyjne	17
4.6.1	Nastawy sterownika kinematycznego	18
4.6.2	Wpływ wartości wzmocnienia k_m	21
4.6.3	Podsumowanie	23
5	Algorytm stabilizacji w punkcie dla monocykla	25
5.1	Postać typowa układu	25
5.2	Algorytm Astolfiego	25
5.3	Zastosowanie algorytmu do monocykla nieholonomicznego	27
5.4	Badania symulacyjne	28
5.4.1	Wpływ wartości parametru k na zbieżność algorytmu	28
5.4.2	Wpływ wartości wzmocnienia k_m	30
5.4.3	Podsumowanie	32

6	Algorytm śledzenia ścieżki dla monocykla	33
6.1	Opis układu w układzie Serreta-Freneta	33
6.2	Algorytm Samsona	35
6.3	Badania symulacyjne	35
6.3.1	Nastawy sterownika kinematycznego	36
6.3.2	Wpływ sterownika dynamicznego na zbieżność algorytmu	40
6.3.3	Podsumowanie	42
7	Podsumowanie	43
	Bibilografia	45

Rozdział 1

Wprowadzenie

Początki robotyki sięgają drugiej połowy XX wieku – w roku 1961 firma General Motors uruchomiła pierwszy manipulator przemysłowy. Robotyka jest dziedziną wiedzy łączącą w sobie inne obszary naukowe, m. in. mechanikę, elektronikę, automatykę, informatykę. W jej ramach można zajmować się rozważaniami nad modelami teoretycznymi, symulacjami działania zaprojektowanych maszyn, ich budową i programowaniem, a także rozwojem sztucznej inteligencji i zagadnieniami związanymi z komunikacją pomiędzy człowiekiem a robotem.

Nie ma jednego, ścisłego podziału robotów, poszczególne kategorie przenikają się wzajemnie, można jednak wyróżnić kilka rodzajów maszyn; roboty i manipulatory przemysłowe, roboty medyczne, chirurgiczne oraz roboty mobilne. Te ostatnie można podzielić na kroczące, kołowe, podwodne, latające oraz kosmiczne.

W zależności od ograniczeń nałożonych na robota, maszyny mobilne można podzielić na nieholonomiczne i holonomiczne. W przypadku pierwszych zakłada się, że robot nie może się ślizgać. Ułatwia to określanie stanu robota – aby wyznaczyć jego położenie wystarczy zliczać obroty kół, np. za pomocą odpowiednio dobranych enkoderów. Ponieważ roboty holonomiczne mogą się pośliznąć, do autolokalizacji potrzebują sensorów takich jak czujniki ultradźwiękowe czy kamery.

Holonomiczność jest cechą określającą zależność między kontrolowaną a dostępną liczbą stopni swobody robota. Jeśli te liczby są sobie równe, robota określa się jako holonomicznego. Oznacza to, że może się poruszać w dowolnym kierunku niezależnie od swojej orientacji. Takie zachowanie można uzyskać używając w robocie kół szwedzkich (omni-kierunkowych). Gdy liczba stopni swobody, które można kontrolować, jest mniejsza niż liczba dostępnych stopni swobody, robot jest nieholonomiczny. Przykładem może być samochód, który ma trzy stopnie swobody (ruch po osi X, po osi Y i prędkość), ale tylko dwa z nich da się kontrolować (obrot kierownicy i prędkość).

Każdego konwencjonalnego robota mobilnego można również przyporządkować do określonej klasy [3]. Klasa robota mobilnego jest opisana przy pomocy dwóch liczb (δ_m, δ_s). Pierwsza z nich to stopień mobilności robota – liczba stopni swobody, na które można wpływać za pomocą zmiany prędkości kół. Druga liczba określa stopień sterowalności, czyli liczbę kół niezależnie orientowanych. Aby ruch był możliwy, liczby te muszą spełniać następującą zależność

$$\begin{aligned}1 &\leq \delta_m \leq 3, \\0 &\leq \delta_s \leq 2, \\2 &\leq \delta_m + \delta_s \leq 3.\end{aligned}$$

Z pięciu możliwych klas tylko jedna z nich – (3, 0), opisuje robota holonomicznego.

1.1 Cel pracy

Celem pracy jest zapoznanie się z modelem nieholonomicznej platformy kołowej klasy $(2, 0)$, tzw. monocyklem, oraz podstawowymi zadaniami, które realizuje.

1.2 Przegląd zawartości pracy

Pierwszy rozdział to wstęp określający problematykę pracy. Drugi rozdział zawiera opis modelu kinematyki i dynamiki robota oraz wyprowadzenie równań potrzebnych przy projektowaniu sterowników. Ogólny problem sterowania oraz wyszczególnienie zadań stawianych przed monocyklem przedstawiono w rozdziale trzecim. W czwartym rozdziale opisano zadanie śledzenia trajektorii dla monocykla oraz zaprojektowano sterownik w oparciu o algorytm Samsona. Piąty rozdział to wyprowadzenie równań potrzebnych w algorytmie Astolfiego, który stosuje się do stabilizacji robota w punkcie. W szóstym rozdziale przedstawiono algorytm Samsona realizujący śledzenie ścieżki. Każdy z rozdziałów 5-7 zawiera również symulacje pozwalające określić wpływ wartości nastaw każdego ze sterowników na zbieżność algorytmów. Siódmy rozdział zawiera podsumowanie zrealizowanej pracy.

Rozdział 2

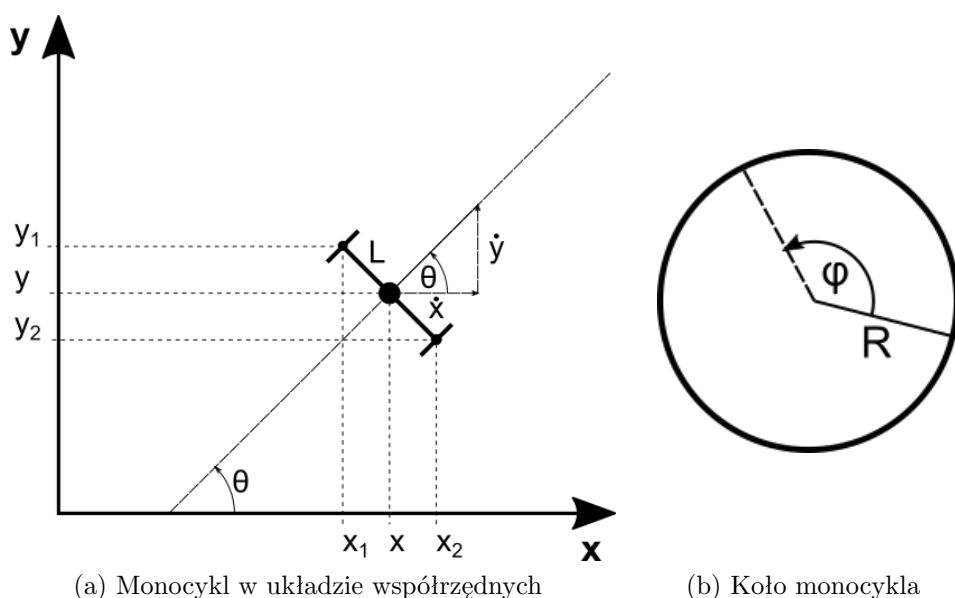
Modelowanie monocykla

Pełny opis robota nieholonomicznego przedstawiają równania ograniczeń ruchu oraz równania dynamiki. Pierwsze z nich, wynikające z założenia braku poślizgu, są ograniczeniami prędkościowymi. Równania dynamiki opisują natomiast zależność pomiędzy strukturą obiektu a jego położeniem, prędkością i przyspieszeniem.

W poniższym rozdziale został opisany model nieholonomicznego monocykla. Podano opis wielkości fizycznych w układzie współrzędnych oraz wyprowadzono równania kinematyki i dynamiki we współrzędnych uogólnionych i pomocniczych.

2.1 Opis układu

Konwencjonalny monocykl nieholonomiczny jest robotem klasy $(2, 0)$. Oznacza to, że posiada dwa niezależnie sterowane koła umieszczone na wspólnej osi oraz nie posiada kół skrętnych.



Rysunek 2.1: Układ robotyczny monocykla nieholonomicznego

Monocykl opisuje się za pomocą następujących wartości:

- R – promień koła,
- L – długość połowy osi,

- (x, y) – położenie środka masy układu,
- $(x_1, y_1), (x_2, y_2)$ – położenie środka kół (punktów styczności kół z podłożem),
- θ – orientacja układu,
- φ_1, φ_2 – kąty obrotu kół platformy,
- $\mathbf{q} = \begin{pmatrix} x \\ y \\ \theta \\ \varphi_1 \\ \varphi_2 \end{pmatrix} \in \mathbb{R}^N$ – wektor współrzędnych uogólnionych,
- $\dot{\mathbf{q}} \in \mathbb{R}^N$ – wektor prędkości uogólnionych.

Równania kinematyki i dynamiki robota można przedstawić na kilka sposobów. W tej pracy wykorzystano dwa – współrzędne uogólnione oraz współrzędne pomocnicze.

2.2 Ograniczenia nieholonomiczne

Założono następujące ograniczenia ruchu robota – układ przemieszcza się po płaszczyźnie poziomej, szkielet i koła robota są sztywne, układ porusza się bez poślizgu kół.

2.2.1 Współrzędne uogólnione

Ograniczenia nieholonomiczne nałożone na układ wynikają z warunków na brak poślizgu bocznego i wzdłużnego obu kół. Zależności wyliczone na podstawie rysunku 2.1 mają postać [5]:

- brak poślizgu wzdłużnego

$$\dot{x} \cos \theta + \dot{y} \sin \theta - R\dot{\varphi} = 0, \quad (2.1)$$

- brak poślizgu bocznego

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0. \quad (2.2)$$

Parametr φ jest inny dla każdego z kół, dlatego równanie (2.1) rozbija się na dwa

$$\begin{cases} \dot{x}_1 \cos \theta + \dot{y}_1 \sin \theta - R\dot{\varphi}_1 = 0, \\ \dot{x}_2 \cos \theta + \dot{y}_2 \sin \theta - R\dot{\varphi}_2 = 0. \end{cases}$$

Aby te równania wyrazić we współrzędnych uogólnionych, korzysta się z poniższych zależności

$$\begin{aligned} x_1 = x - L \sin \theta &\Rightarrow \dot{x}_1 = \dot{x} - L \cos \theta \dot{\theta}, \\ y_1 = y + L \cos \theta &\Rightarrow \dot{y}_1 = \dot{y} - L \sin \theta \dot{\theta}, \\ x_2 = x + L \sin \theta &\Rightarrow \dot{x}_2 = \dot{x} + L \cos \theta \dot{\theta}, \\ y_2 = y - L \cos \theta &\Rightarrow \dot{y}_2 = \dot{y} + L \sin \theta \dot{\theta}. \end{aligned}$$

Po podstawieniu i uporządkowaniu równań otrzymuje się

$$\begin{cases} \dot{x} \cos \theta + \dot{y} \sin \theta - L\dot{\theta} - R\dot{\varphi}_1 = 0, \\ \dot{x} \cos \theta + \dot{y} \sin \theta + L\dot{\theta} - R\dot{\varphi}_2 = 0. \end{cases}$$

Wszystkie zależności na brak poślizgu opisuje więc układ równań

$$\begin{cases} \dot{x} \sin \theta - \dot{y} \cos \theta = 0, \\ \dot{x} \cos \theta + \dot{y} \sin \theta - L\dot{\theta} - R\dot{\varphi}_1 = 0, \\ \dot{x} \cos \theta + \dot{y} \sin \theta + L\dot{\theta} - R\dot{\varphi}_2 = 0. \end{cases} \quad (2.3)$$

Powyższe warunki można przedstawić w postaci fazowej, czyli tzw. postaci Pfaffa

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}, \quad (2.4)$$

gdzie $\mathbf{A}(\mathbf{q})$ to macierz ograniczeń Pfaffa.

Równanie (2.4) można wyrazić również w postaci macierzowej jako

$$\begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & -L & -R & 0 \\ \cos \theta & \sin \theta & L & 0 & -R \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (2.5)$$

2.2.2 Współrzędne pomocnicze

Ponieważ wszystkie ograniczenia są nieholonomiczne, dopuszczalne prędkości układu zawierają się w jądrze macierzy Pfaffa

$$\dot{\mathbf{q}} \in \text{Ker } \mathbf{A}(\mathbf{q}), \quad \text{rank}(\text{Ker } \mathbf{A}(\mathbf{q})) = n - l = m,$$

gdzie:

n – liczba niezależnych zmiennych uogólnionych,

l – liczba niezależnych ograniczeń.

Z tego wynika, że prędkości te można wyrazić przy pomocy wektorów $\boldsymbol{\eta}$ rozpinających jądro macierzy Pfaffa

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\boldsymbol{\eta}, \quad (2.6)$$

gdzie $\boldsymbol{\eta} \in \mathbb{R}^m$ to prędkości pomocnicze pełniące rolę sterowań.

Jest to postać bezdryfowego układu sterowania. W przypadku monocykla macierz $\mathbf{G}(\mathbf{q})$ będzie miała wymiar $n \times m$, czyli 5×2 . Generatory macierzy $\mathbf{G}(\mathbf{q})$ wylicza się z zależności $\mathbf{A}\mathbf{G} = \mathbf{0}$, np. w następujący sposób

$$\begin{bmatrix} \sin \theta & -\cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & -L & -R & 0 \\ \cos \theta & \sin \theta & L & 0 & -R \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \\ a_5 & b_5 \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{cases} \sin \theta a_1 - \cos \theta a_2 = 0 & \Rightarrow & a_1 = \cos \theta, & a_2 = \sin \theta, \\ \cos \theta a_1 + \sin \theta a_2 - La_3 - Ra_4 = 0, \\ \cos \theta a_1 + \sin \theta a_2 + La_3 - Ra_5 = 0, \end{cases}$$

$$a_4 + a_5 = \frac{2}{R},$$

$$a_4 = \frac{2}{R}, \quad a_5 = 0 \quad \Rightarrow \quad a_3 = \frac{1}{L} \quad \vee \quad a_4 = 0, \quad a_5 = \frac{2}{R} \quad \Rightarrow \quad a_3 = -\frac{1}{L}.$$

Z powyższych wyliczeń wynika, że generatory mogą być równe odpowiednio

$$g_1(\mathbf{q}) = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \frac{1}{L} \\ 0 \\ \frac{2}{R} \end{pmatrix}, \quad g_2(\mathbf{q}) = \begin{pmatrix} \cos \theta \\ \sin \theta \\ -\frac{1}{L} \\ \frac{2}{R} \\ 0 \end{pmatrix}.$$

Ostatecznie układ sterowania (2.6) w postaci macierzowej ma postać

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{pmatrix} = \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ \frac{1}{L} & -\frac{1}{L} \\ 0 & \frac{2}{R} \\ \frac{2}{R} & 0 \end{bmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}. \quad (2.7)$$

2.3 Dynamika monocykla

Wyrażenie równań dynamiki we współrzędnych uogólnionych ma istotne wady – układ jest niedosterowany (występuje więcej wielkości sterowanych niż wejść), a w równaniach obecne są mnożniki Lagrange'a, które są trudne obliczeniowo. Z tego powodu w zadaniach sterowania używa się najczęściej drugiego opisu, czyli modelu wyrażonego w tzw. prędkościach pomocniczych. Wadą współrzędnych pomocniczych jest natomiast brak łatwo wyliczalnych interpretacji fizycznych pomiędzy prędkościami pomocniczymi a konfiguracją.

2.3.1 Współrzędne uogólnione

W układach nieholonomicznych obowiązuje zasada, iż każde niezależne ograniczenie nieholonomiczne nałożone na ruch układu blokuje jedno sterowanie. Dynamikę robota uzyskuje się więc wyznaczając najpierw lagranżian dla układu bez ograniczeń fazowych (układu swobodnego)

$$\mathbf{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}_k(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{E}_p(\mathbf{q}), \quad (2.8)$$

gdzie::

$\mathbf{E}_k(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q}) \dot{\mathbf{q}}$ – energia kinetyczna układu swobodnego,

$\mathbf{E}_p(\mathbf{q})$ – energia potencjalna układu swobodnego.

Zgodnie z *Zasadą Najmniejszego Działania Hamiltona* równania dynamiki można zapisać w postaci równań Eulera-Lagrange'a

$$\frac{d}{dt} \frac{\partial \mathbf{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \frac{\partial^2 \mathbf{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}^2} \ddot{\mathbf{q}} + \frac{\partial^2 \mathbf{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q} \partial \dot{\mathbf{q}}} \dot{\mathbf{q}} - \frac{\partial \mathbf{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \mathbf{F}, \quad (2.9)$$

gdzie \mathbf{F} to uogólnione siły działające na układ takie jak tarcie, opory ruchu, sterowanie.

Aby połączyć model dynamiki (2.9) z postacią bezdryfowego układu sterowania (2.6) wykorzystuje się zasadę d'Alemberta, zgodnie z którą siły niepotencjalne, zapewniające spełnienie ograniczeń fazowych (siły więzów), nie wykonują pracy na dopuszczalnych przemieszczeniach układu

$$\mathbf{F}^T \dot{\mathbf{q}} = 0.$$

Powyższa równość, wobec założenia niezależności ograniczeń fazowych, prowadzi do wniosku, że istnieje wektor mnożników Lagrange'a $\boldsymbol{\lambda}$ taki, że [12]

$$\mathbf{F}^T = \boldsymbol{\lambda}^T \mathbf{A}(\mathbf{q}) \quad \Rightarrow \quad \mathbf{F} = \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda}.$$

Równanie (2.6) można więc zapisać w postaci

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{L}}{\partial \mathbf{q}} = \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda}.$$

Po wzięciu pod uwagę, że sterowanie wpływa na m stopni swobody układu poprzez macierz wejściową $\mathbf{B}(\mathbf{q})$, równanie dynamiki wygląda następująco

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathbf{L}}{\partial \mathbf{q}} = \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda} + \mathbf{B}(\mathbf{q})\mathbf{u}. \quad (2.10)$$

Korzystając z równania (2.8) oraz (2.10) można określić dynamikę robota we współrzędnych uogólnionych jako

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}(\mathbf{q}) = \mathbf{B}\mathbf{u} + \mathbf{A}^T\boldsymbol{\lambda}, \quad (2.11)$$

gdzie:

$\mathbf{M}(\mathbf{q})$ – symetryczna, dodatnio określona macierz inercji,

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ – macierz sił Coriolisa i sił odśrodkowych,

$\mathbf{D}(\mathbf{q})$ – wektor sił grawitacji,

$\mathbf{B} \in \mathbb{R}^{n \times m}$ – macierz wejściowa,

$\mathbf{u} \in \mathbb{R}^m$ – wektor sterowań,

$\boldsymbol{\lambda} \in \mathbb{R}^l$ – wektor mnożników Lagrange'a,

n – wymiar stanu,

l – liczba ograniczeń nieholonomicznych,

$m = n - l$.

Macierz inercji, określona na podstawie energii kinetycznej całego układu, oraz macierz wejściowa wynoszą odpowiednio

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_c & 0 & 0 & 0 & 0 \\ 0 & M_c & 0 & 0 & 0 \\ 0 & 0 & I_p & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 \\ 0 & 0 & 0 & 0 & I_{xx} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

gdzie:

$M_c = M_p + 2M_k$ – masa całego układu równa masie platformy i masie obu kół,

I_p – moment bezwładności platformy wraz z kołami,

I_{xx} – moment bezwładności koła.

Zakłada się, że oba koła mają taką samą masę oraz taki sam moment bezwładności.

Macierz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ uzyskuje się z macierzy bezwładności licząc tzw. symbole Christoffela I rodzaju [12]

$$C_{ij}(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{k=1}^n c_{kj}^i(\mathbf{q})\dot{q}_k,$$

gdzie:

$$c_{kj}^i(\mathbf{q}) = \frac{1}{2} \left(\frac{\partial M_{ij}(\mathbf{q})}{\partial q_k} + \frac{\partial M_{ik}(\mathbf{q})}{\partial q_j} + \frac{\partial M_{jk}(\mathbf{q})}{\partial q_i} \right).$$

Ponieważ elementy macierzy $\mathbf{M}(\mathbf{q})$ są stałe, macierz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ jest równa 0.

Wektor $\mathbf{D}(\mathbf{q})$ również się zeruje, ponieważ układ nie zmienia wysokości nad poziomem podłoża.

2.3.2 Współrzędne pomocnicze

Aby wyrazić dynamikę monocykla we współrzędnych pomocniczych korzysta się z równania (2.6). Wyprowadzenie wygląda następująco

$$\begin{aligned}
M\ddot{q} + C\dot{q} + D &= Bu + A^T\lambda, \\
M(\dot{G}\eta + G\dot{\eta}) + C(G\eta) + D &= Bu + A^T\lambda, \\
G^T / MG\dot{\eta} + (M\dot{G} + CG)\eta + D &= Bu + A^T\lambda, \\
\underbrace{G^T MG\dot{\eta}}_{M^*} + \underbrace{G^T(M\dot{G} + CG)\eta}_{C^*} + \underbrace{G^T D}_{D^*} &= \underbrace{G^T B}_{B^*}u + \underbrace{G^T A^T\lambda}_{=0}, \\
M^*\dot{\eta} + C^*\eta + D^* &= B^*u.
\end{aligned} \tag{2.12}$$

Nowa macierz bezwładności wynosi

$$M^* = \begin{bmatrix} M_c + 4\frac{I_{xx}}{R^2} + \frac{I_p}{L^2} & M_c - \frac{I_p}{L^2} \\ M_c - \frac{I_p}{L^2} & M_c + 4\frac{I_{xx}}{R^2} + \frac{I_p}{L^2} \end{bmatrix}.$$

Zgodnie z zależnościami przedstawionymi w sekcji 2.3.1 macierze $C^*(q, \dot{q})$ oraz D^* ulegają wyzerowaniu.

Nowa macierz wejściowa wygląda następująco

$$B = \begin{bmatrix} \frac{2}{R} & 0 \\ 0 & \frac{2}{R} \end{bmatrix}.$$

2.4 Układ kaskadowy

Kompletny opis monocykla nieholonomicznego przedstawia układ równań

$$\begin{cases} \dot{q} = G(q)\eta & \Rightarrow \text{I stopień kaskady,} \\ M^*\dot{\eta} + C^*\eta + D^* = B^*u & \Rightarrow \text{II stopień kaskady.} \end{cases}$$

Jest to układ kaskadowy charakteryzujący się tym, że każdą grupę równań można potraktować oddzielnie. To oznacza, że w i -tym podsystemie funkcje nie mogą zależeć od zmiennych o wyższych niż i numerach. Te zmienne mogą natomiast pełnić rolę sterowań.

Jak widać, zmienne stanu równań dynamiki η (II stopień kaskady), pełnią rolę sterowań w równaniach kinematyki (I stopień kaskady).

Rozdział 3

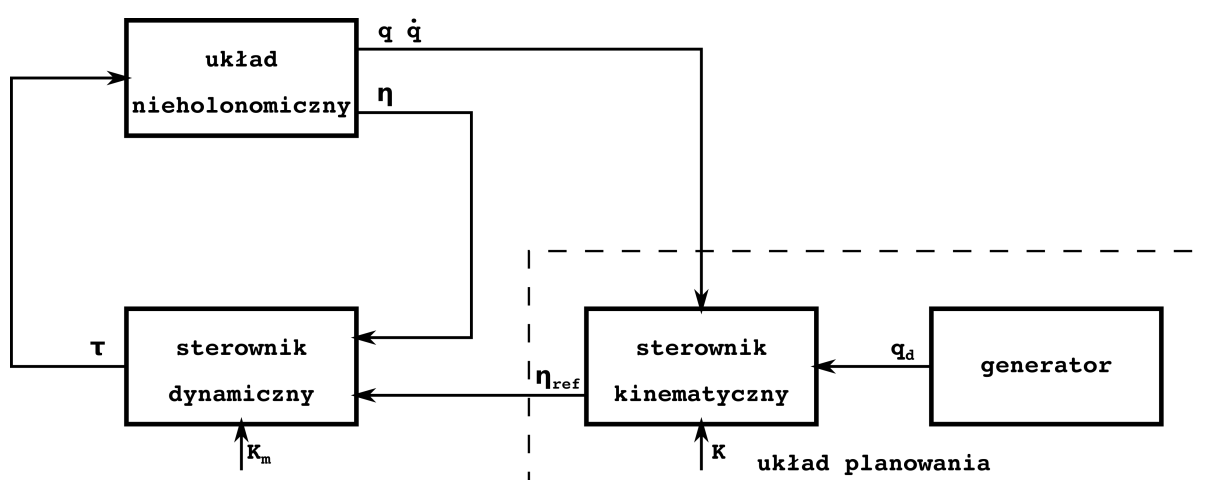
Sformułowanie problemu sterowania

Zadanie postawione w pracy polega na zaprojektowaniu algorytmów sterowania dla rzeczywistego obiektu – uwzględniając kinematykę i dynamikę. Rzeczywisty monocykl ma strukturę kaskadową (podrozdział 2.4). W pracy założono, że

- kinematyka obiektu jest w pełni znana,
- dynamika układu również jest w pełni znana.

Głównym problemem w sterowaniu układem o strukturze kaskadowej jest niemożność bezpośredniego wpływania na obiekt. Dla tego typu układów wykorzystuje się algorytm całkowania wstecznego (*backstepping integrator*) [5]. Polega on na stworzeniu sterownika działającego w sposób rekurencyjny, który traktuje niektóre zmienne jako wejścia kolejnego podsystemu.

Układ sterowania monocykla nieholonomicznego zawiera trzy elementy – właściwy obiekt, układ planowania oraz sterownik dynamiczny. Cała struktura została przedstawiona na rysunku 3.1. Widoczne na schemacie wielkości K oraz K_m , to nastawy regulatorów zadawane z zewnątrz.



Rysunek 3.1: Schemat układu sterowania

Główną rolę w układzie odgrywa sterownik kinematyczny. To on rozwiązuje zadania postawione przed monocyklem – zawiera implementację wybranego algorytmu sterowania. Generator jest elementem, który nie zawsze występuje w modelu w sposób jawny.

W dalszej pracy sterownik dynamiczny będzie miał identyczną strukturę niezależnie od rozwiązywanego zadania. Jego model matematyczny zostanie przedstawiony poniżej.

3.1 Sterownik dynamiczny

W projektowaniu tego sterownika należy wziąć pod uwagę fakt, że w układzie kaskadowym nie można bezpośrednio sterować prędkościami. Z tego powodu pojawiają się różnice pomiędzy prędkościami zadanymi $\boldsymbol{\eta}_{ref}$ a prędkościami rzeczywistymi $\boldsymbol{\eta}$. Zadaniem sterownika dynamicznego jest sprawienie, by prędkości rzeczywiste $\boldsymbol{\eta}$ dążyły do prędkości wygenerowanych przez sterownik kinematyczny.

Sterowanie uzyskuje się na podstawie równania (2.12). Zgodnie z tym co omówiono w rozdziale 2.3.2 macierze \mathbf{C}^* oraz \mathbf{D}^* zerują się. Równanie dynamiki można więc określić jako

$$\mathbf{M}^* \dot{\boldsymbol{\eta}} = \mathbf{B} \mathbf{u}.$$

W tej pracy do sterowania dynamiką obiektu wykorzystano algorytm dokładnej linearyzacji

$$\mathbf{u} = \mathbf{B}^{-1} (\mathbf{M}^* \dot{\boldsymbol{\eta}}_r - \mathbf{K}_m \mathbf{e}_\eta), \quad (3.1)$$

gdzie:

$$\mathbf{K}_m = \begin{bmatrix} k_m & 0 \\ 0 & k_m \end{bmatrix}, \quad k_m > 0 \text{ – diagonalna macierz wzmocnienia,}$$

$$\mathbf{e}_\eta = \boldsymbol{\eta} - \boldsymbol{\eta}_r.$$

Dowód zbieżności powyższego algorytmu można przedstawić w następujący sposób

- równanie układu z zamkniętą pętlą sprzężenia

$$\mathbf{M} \dot{\boldsymbol{\eta}} = \mathbf{B} \mathbf{B}^{-1} (\mathbf{M} \dot{\boldsymbol{\eta}}_r - \mathbf{K}_m \mathbf{e}_\eta),$$

$$\mathbf{M} \dot{\boldsymbol{\eta}} = \mathbf{M} \dot{\boldsymbol{\eta}}_r - \mathbf{K}_m \mathbf{e}_\eta,$$

$$\mathbf{M} (\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_r) + \mathbf{K}_m \mathbf{e}_\eta = 0,$$

$$\mathbf{M} \dot{\mathbf{e}}_\eta + \mathbf{K}_m \mathbf{e}_\eta = 0,$$

- funkcja Lapunowa

$$\mathbf{V}(\mathbf{e}_\eta) = \frac{1}{2} \mathbf{e}_\eta^T \mathbf{M} \mathbf{e}_\eta,$$

- pochodna funkcji Lapunowa

$$\dot{\mathbf{V}} = \mathbf{e}_\eta^T \mathbf{M} \dot{\mathbf{e}}_\eta + \underbrace{\frac{1}{2} \mathbf{e}_\eta^T \dot{\mathbf{M}} \mathbf{e}_\eta}_{=0},$$

- pochodna liczona wzdłuż trajektorii układu z zamkniętą pętlą

$$\dot{\mathbf{V}} = \mathbf{e}_\eta^T (-\mathbf{K}_m \mathbf{e}_\eta),$$

$$\dot{\mathbf{V}} = -\mathbf{e}_\eta^T \mathbf{M}^* \mathbf{e}_\eta \leq 0,$$

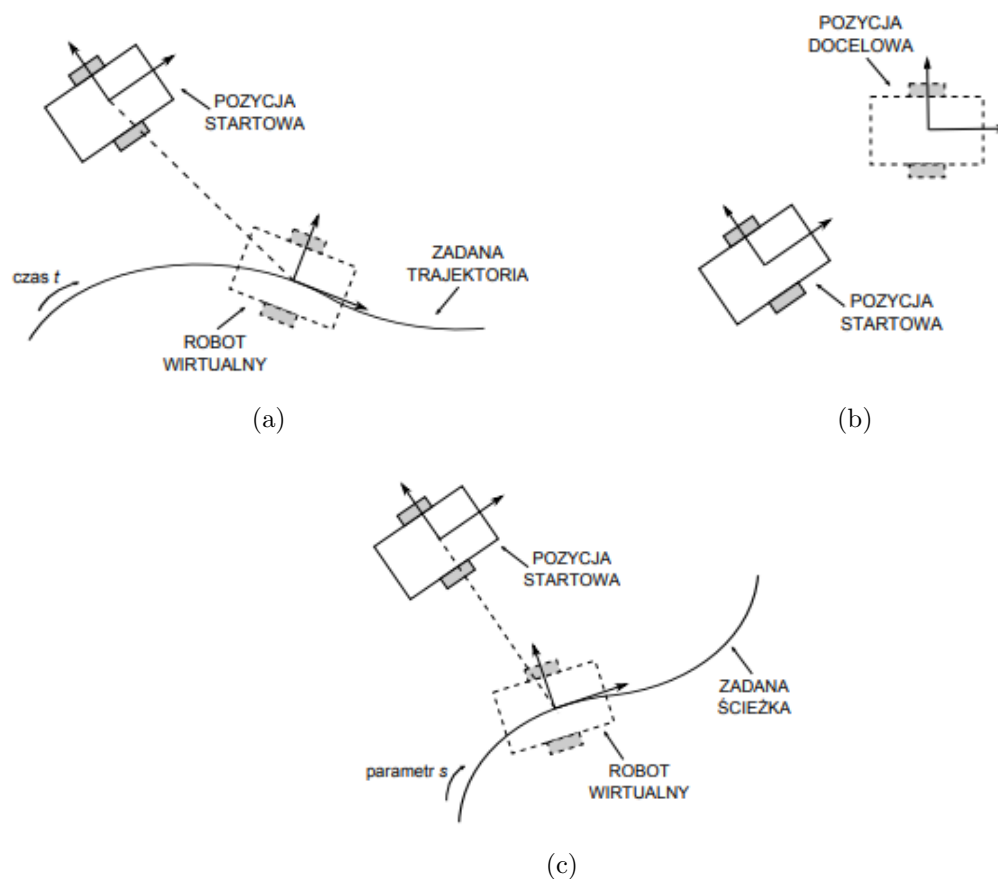
- \mathbf{e}_η jest punktem równowagi asymptotycznie stabilnym na mocy zasady niezmienności LaSalle'a.

3.2 Zadania sterowania

Każde z zaproponowanych zadań sterowania robotem mobilnym musi spełniać określone kryteria:

- zadanie musi się dać zrealizować – trajektorie muszą być dopuszczalne,
- ścieżki muszą być odpowiednio gładkie.

Podstawowe typy zadań wyróżnionych dla monocykla nieholonomicznego zostały przedstawione na rys. 3.2.



Rysunek 3.2: Typy zadań wyróżnionych dla platformy mobilnej; (a) śledzenie trajektorii, (b) sterowanie do punktu, (c) śledzenie ścieżki [9]

Każde z wymienionych zadań można rozwiązać przy pomocy różnych algorytmów. W tej pracy przedstawiono następujące:

- dla śledzenia trajektorii – algorytm Samsona i Ait-Abderrahima,
- dla sterowania do punktu – algorytm Astolfiego,
- dla śledzenia ścieżki – algorytm Samsona.

Rozdział 4

Algorytm śledzenia trajektorii dla monocykla

W robotyce trajektoria to krzywa $\mathbf{q}_d(t)$ parametryzowana zmienną czasu t . Zakłada się, że jest ona klasy c^2 . Zadanie śledzenia trajektorii jest realizowane przez I stopień kaskady i polega na asymptotycznym stabilizowaniu konfiguracji $\mathbf{q}(t)$ robota do zadanej krzywej.

W poniższym rozdziale zostanie opisany algorytm Samsona, który jest głównym sposobem rozwiązywania tego typu zadań dla monocykla. Przedstawione zostaną również wyniki przeprowadzonych badań.

4.1 Model obiektu

Równania potrzebne do zaprojektowania modelu wyznaczają się z zależności (2.7)

$$\begin{cases} \dot{x} = \cos \theta (\eta_1 + \eta_2), \\ \dot{y} = \sin \theta (\eta_1 + \eta_2), \\ \dot{\theta} = \frac{1}{L} (\eta_1 - \eta_2), \\ \dot{\varphi}_1 = \frac{2}{R} \eta_1 \Rightarrow \eta_1 = \frac{R}{2} \dot{\varphi}_1, \\ \dot{\varphi}_2 = \frac{2}{R} \eta_2 \Rightarrow \eta_2 = \frac{R}{2} \dot{\varphi}_2. \end{cases}$$

Wejściem obiektu jest wektor sterowań \mathbf{u} składający się z prędkości liniowej v oraz prędkości kątowej ω

$$\begin{cases} v = \eta_1 + \eta_2, \\ \omega = \frac{1}{L} (\eta_1 - \eta_2), \end{cases}$$

natomiast wyjściem są wartości x , y , θ wyliczane na podstawie zależności geometrycznych przedstawionych na rys. 2.1

$$\begin{cases} \dot{x} = v \cos \theta, \\ \dot{y} = v \sin \theta, \\ \dot{\theta} = \omega. \end{cases} \quad (4.1)$$

Parametry własne badanego w tej pracy monocykla nieholonomicznego zaczerpnięto z literatury [13]. Macierz inercji \mathbf{M}^* ma postać

$$\mathbf{M}^* = \begin{bmatrix} 107.9 & 0 \\ 0 & 83.6 \end{bmatrix}.$$

4.2 Model generatora

W zadaniu śledzenia trajektorii, w układzie sterowania przedstawionym na rysunku 3.1, blok generatora występuje jawnie i ma za zadanie tworzyć żądaną trajektorię. Z góry zadany jest wektor sterowań \mathbf{u}_d .

4.3 Błędy śledzenia

Podczas ruchu robota pojawiają się różnice pomiędzy rzeczywistym położeniem robota a punktem, w którym powinien się znajdować. Różnice te nazywane są podstawowymi błędami śledzenia i wylicza się je następująco

$$\begin{cases} e_x = x - x_d, \\ e_y = y - y_d, \\ e_\theta = \theta - \theta_d. \end{cases} \quad (4.2)$$

4.4 Sterownik kinematyczny

Jest to implementacja algorytmu, który ma za zadanie rozwiązać równania ograniczeń robota nie biorąc pod uwagę dynamiki obiektu. Innymi słowy, jest to układ planowania ruchu. Sterownik generuje pewne sterowanie referencyjne $\boldsymbol{\eta}_{ref}$, które zapewnia zmianę konfiguracji z aktualnej na żądaną, spełniając jednocześnie ograniczenia nałożone na układ w każdej chwili czasu.

4.5 Algorytm Samsona i Ait-Abderrahima

Algorytm Samsona i Ait-Abderrahima jest jednym ze sposobów rozwiązania zadania śledzenia trajektorii [11]. Samson zaproponował, by wektor sterowań referencyjnych $\boldsymbol{\eta}_{ref}$ wyliczać w następujący sposób

$$\begin{cases} v_{ref} = v_d \cos \theta_e - K_1 x_e, \\ \omega_{ref} = \omega_d - K_2 \theta_e - v_d y_e \frac{\sin \theta_e}{\theta_e}, \end{cases} \quad (4.3)$$

gdzie $K_1, K_2 > 0$ to parametry regulacji.

W powyższych równaniach występują błędy referencyjne \mathbf{q}_e , które wyznacza się z podstawowych błędów śledzenia poprzez rotację wokół osi z

$$\begin{aligned} \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} &= \text{Rot}(z, -\theta) \begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix}, \\ \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} &= \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix}, \\ \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} &= \begin{pmatrix} \cos \theta e_x + \sin \theta e_y \\ -\sin \theta e_x + \cos \theta e_y \\ e_\theta \end{pmatrix}. \end{aligned} \quad (4.4)$$

Ze względu na to, że macierz rotacji \mathbf{R} jest zawsze nieosobliwa (wyznacznik macierzy jest zawsze równy $\det \mathbf{R} = 1$), to zbieżność błędów referencyjnych do zera powoduje równoczesną zbieżność podstawowych błędów śledzenia do zera.

4.6 Badania symulacyjne

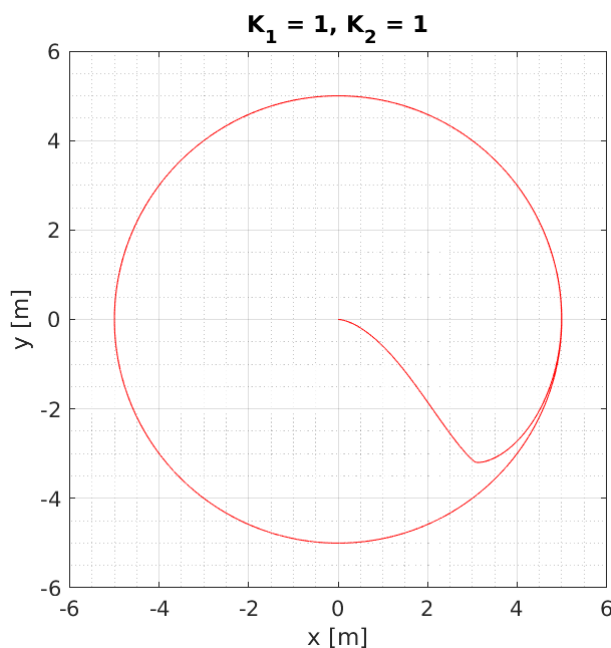
Wszystkie symulacje przeprowadzono dla trajektorii zadanej w kształcie okręgu. Parametry obiektu i trajektorii wynosiły

$$\begin{cases} r_d = 5, \\ \omega_d = 0.1, \\ v_d = r_d \omega_d = 0.5. \end{cases}$$

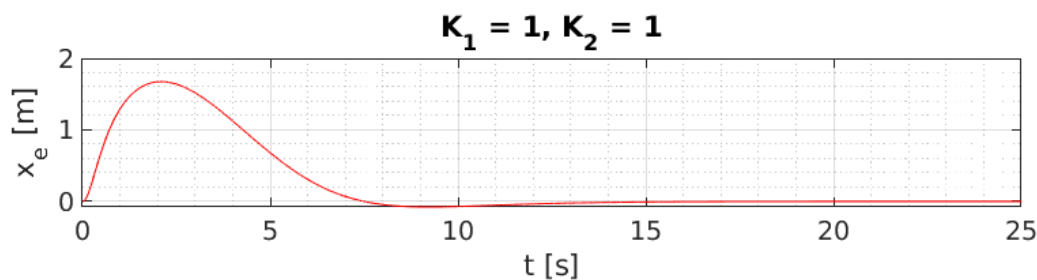
Warunki początkowe obiektu (początkowa konfiguracja robota) wyglądała następująco

$$\begin{cases} x(0) = 0, \\ y(0) = 0, \\ \theta(0) = 0.01. \end{cases}$$

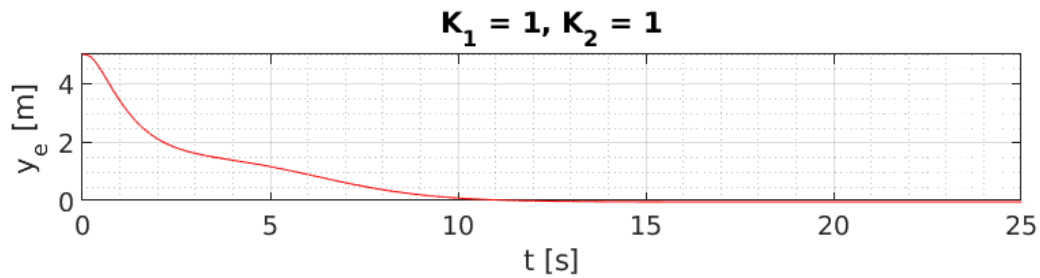
Podczas symulacji starano się sprawdzić, jaki wpływ na zachowanie monocykła mają wartości poszczególnych nastaw regulatorów. Jako rozwiązanie odniesienia przyjęto sterowanie referencyjne wygenerowane przy nastawach równych $K_1 = 1$, $K_2 = 1$, gdy do obiektu podłączony był tylko sterownik kinematyczny (rys. 4.1).



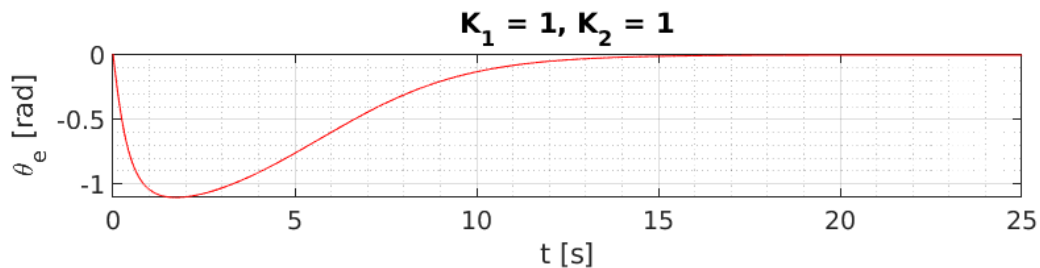
Rysunek 4.1: Trajektoria referencyjna układu ze sterownikiem kinematycznym (rozwiązanie odniesienia)



Rysunek 4.2: Błąd referencyjny x_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia)



Rysunek 4.3: Błąd referencyjny y_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia)



Rysunek 4.4: Błąd referencyjny θ_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia)

We wszystkich poniższych wykresach kolorem czerwonym oznaczono rozwiązanie odniesienia.

4.6.1 Nastawy sterownika kinematycznego

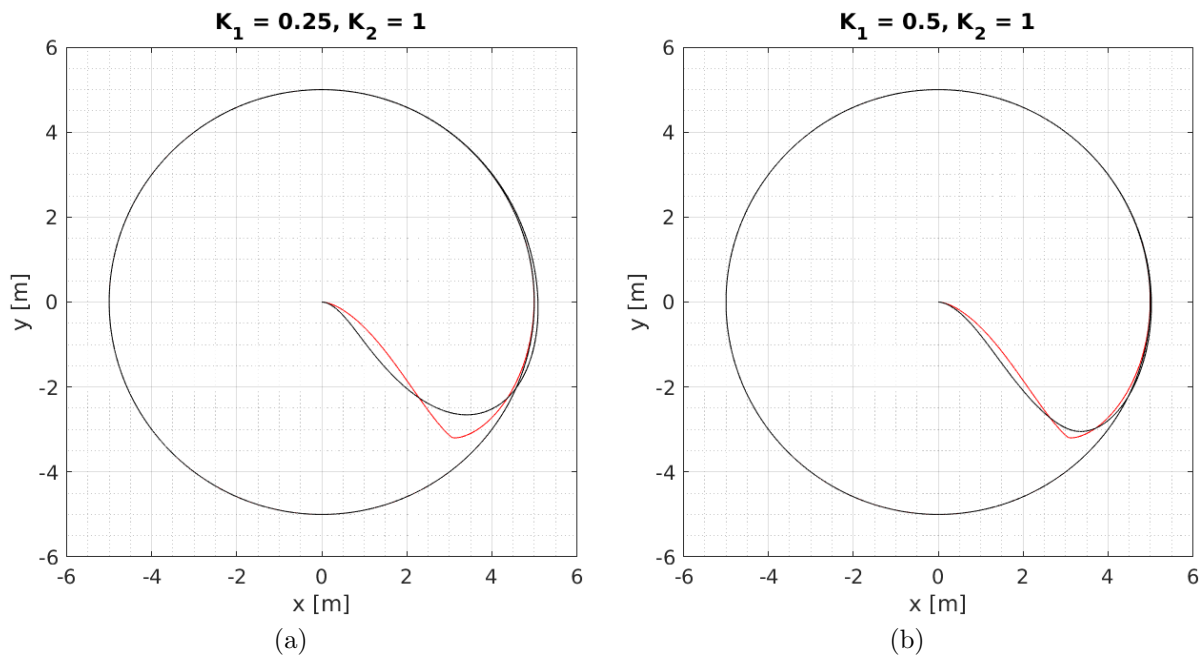
Po przeprowadzeniu badań dla każdej nastawy z osobna sprawdzono, jak zachowuje się obiekt, gdy oba wzmocnienia są sobie równe $K_1 = K_2$. Okazało się, że reakcja jest złożeniem zachowań pojawiających się przy odpowiednich badaniach dotyczących poszczególnych nastaw. Przykładowo, dla $K_1 = K_2 = 2.5$ pojawia się ostra zmiana kierunku ruchu widoczna na wykresie 4.6a oraz podążanie ruchem spiralnym jak na wykresie 4.8a. Zdecydowano się więc pokazać tylko wyniki symulacji dla poszczególnych nastaw.

Zmienna wartość K_1

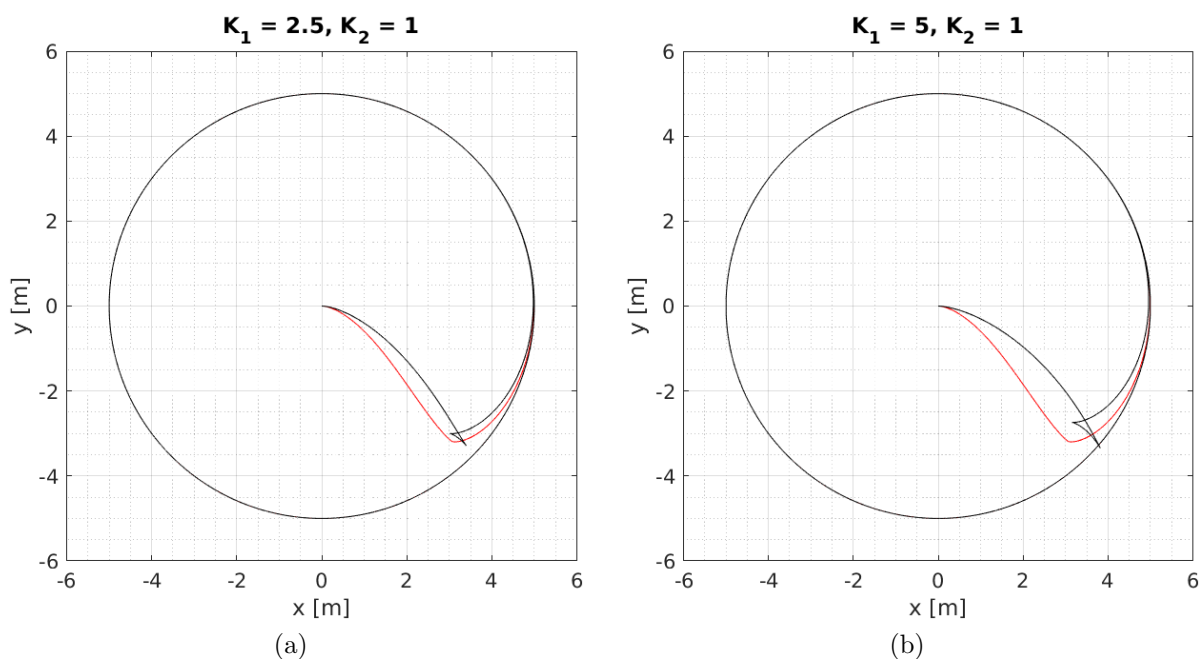
Z równań algorytmu Samsona (4.3) wynika, że wielkość K_1 wpływa na błąd położenia, a tym samym na prędkość liniową v . Wartości te są odwrotnie proporcjonalne. Gdy obiekt porusza się nieco szybciej $K_1 \approx 0.5$ rozwiązanie okazuje się lepsze niż otrzymane sterowanie odniesienia. Przy większych wartościach v układ staje się przeregulowany.

Duże wartości K_1 sprawiają, że układ porusza się wolniej i jest bardziej wrażliwy na zmiany prędkości kątowej. Prowadzi to do gwałtownych zmian kierunku jazdy.

Wyniki badań przedstawiają wykresy 4.5–4.6



Rysunek 4.5: Rozwiązanie odniesienia oraz badana trajektoria, (a) $K_1 = 0.25$, $K_2 = 1$, (b) $K_1 = 0.5$, $K_2 = 1$

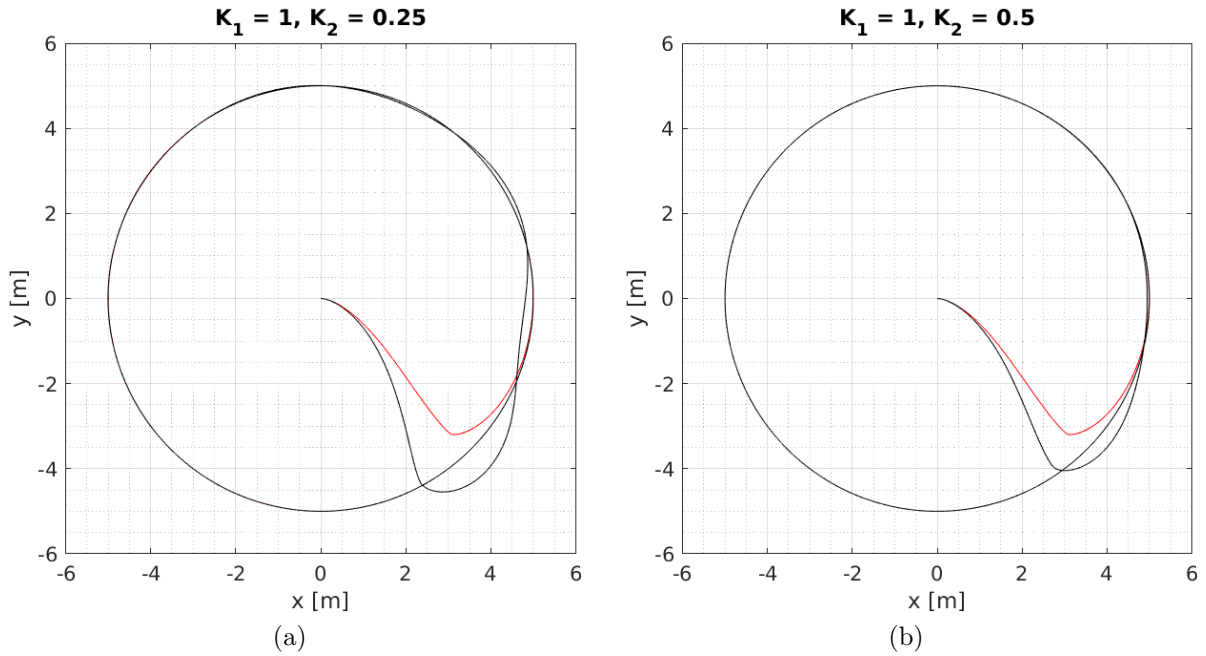


Rysunek 4.6: Rozwiązanie odniesienia oraz badana trajektoria, (a) $K_1 = 2.5$, $K_2 = 1$, (b) $K_1 = 5$, $K_2 = 1$

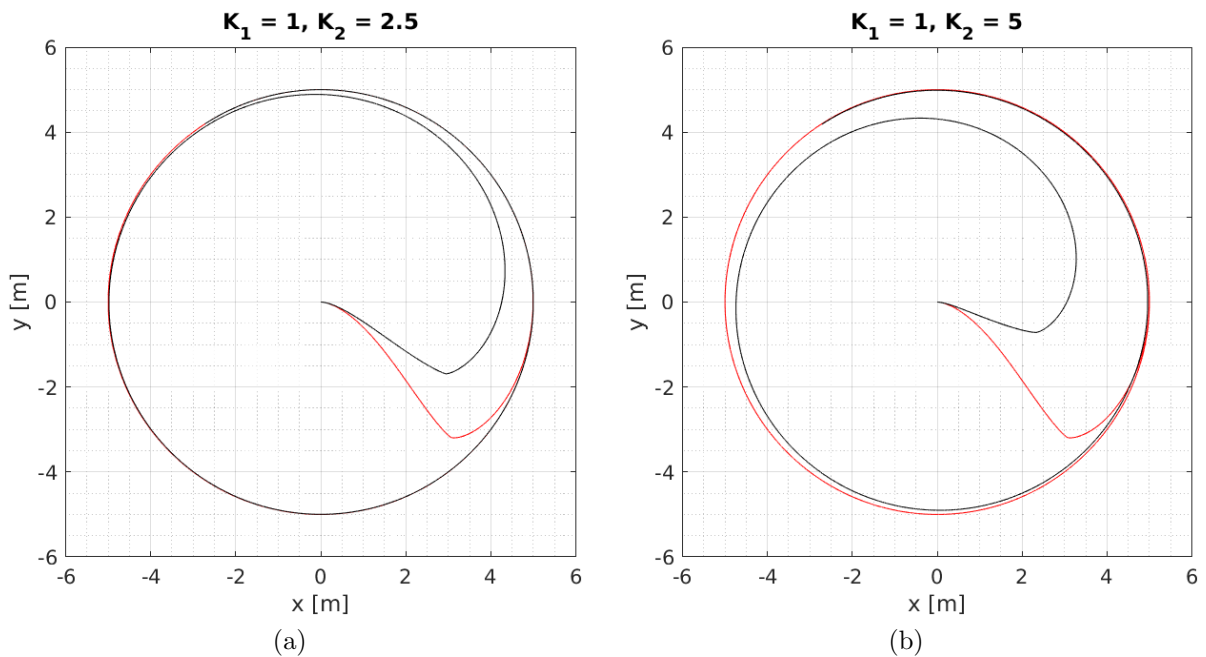
Zmienna wartość K_2

Wielkość K_2 związana jest z błędem orientacji θ – kiedy rośnie, wartość chwilowej prędkości kątowej ω maleje i na odwrót. Zbyt duża prędkość kątowa powoduje, że obiekt jest trudno sterowalny, pojawiają się oscylacje. Natomiast mała prędkość kątowa sprawia, że

układ wolno reaguje na błąd orientacji, a co za tym idzie wolniej zbliża się do żądanej trajektorii. Na wykresach 4.7–4.8 pokazano zachowanie obiektu dla wybranych K_2 .



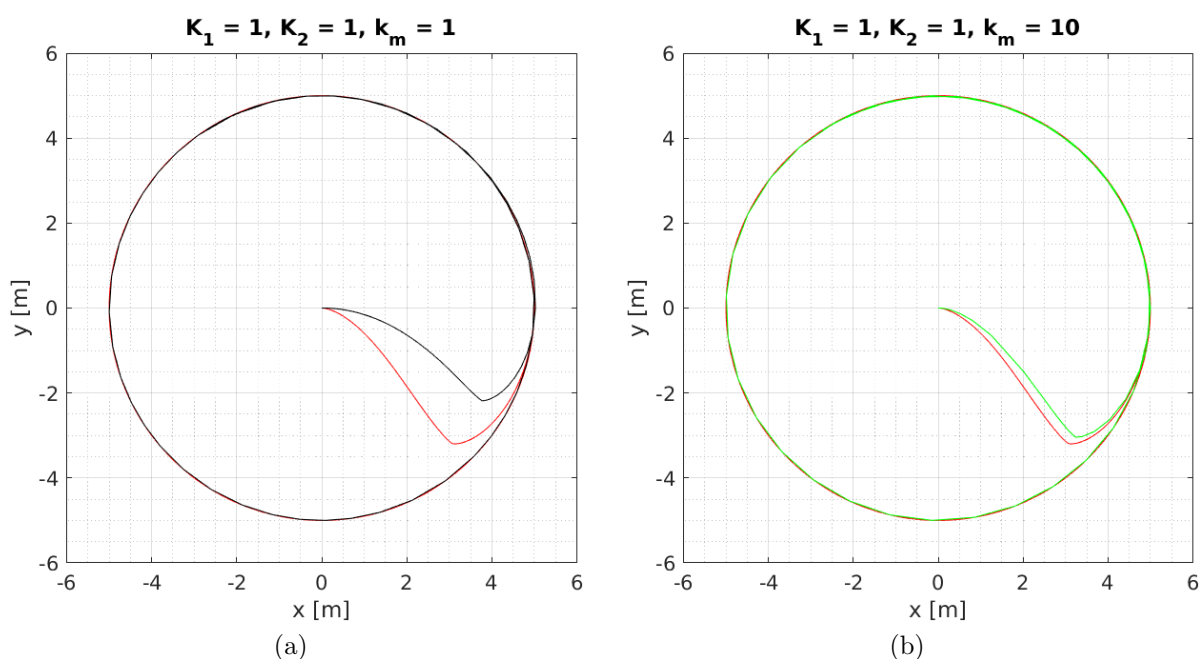
Rysunek 4.7: Rozwiązanie odniesienia oraz badana trajektoria, (a) $K_1 = 1$, $K_2 = 0.25$, (b) $K_1 = 1$, $K_2 = 0.5$



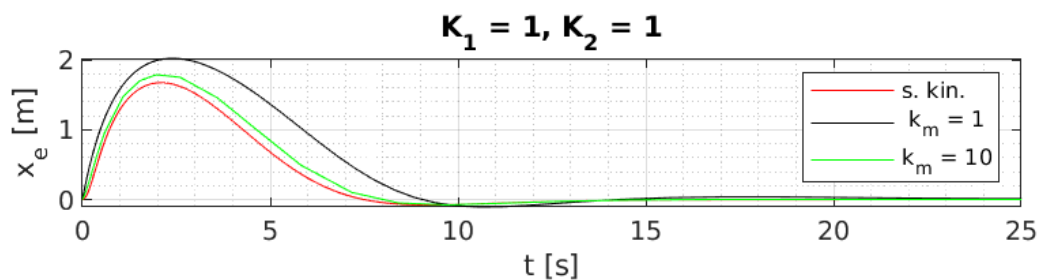
Rysunek 4.8: Rozwiązanie odniesienia oraz badana trajektoria, (a) $K_1 = 1$, $K_2 = 2.5$, (b) $K_1 = 1$, $K_2 = 5$

4.6.2 Wpływ wartości wzmacnienia k_m

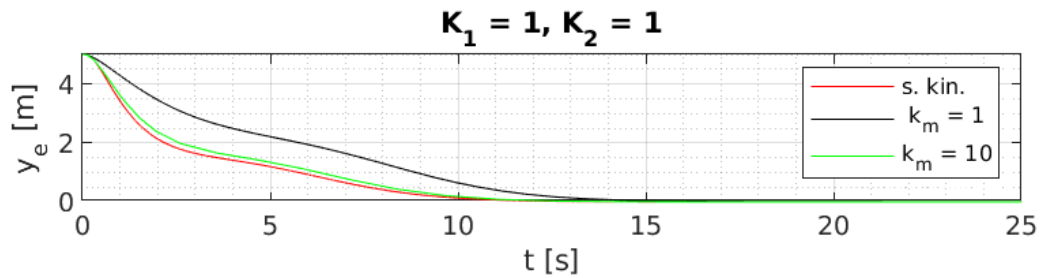
Po podłączeniu do układu sterownika dynamicznego rozpoczęto badania od wartości nastawy $k_m = 1$. Wraz z jej wzrostem rozwiązanie ulegało poprawie. Działo się to jednak na tyle wolno, że do badań zwiększano tę wartość o rząd. Wykresy 4.9–4.16 pokazują osiągnięte trajektorie oraz błędy referencyjne w porównaniu do rozwiązania odniesienia. Na ich podstawie można zauważyć, że przy wartości wzmacnienia równej $k_m = 1000$ rzeczywista trajektoria robota jest prawie identyczna do tej wygenerowanej ze sterownika kinematycznego.



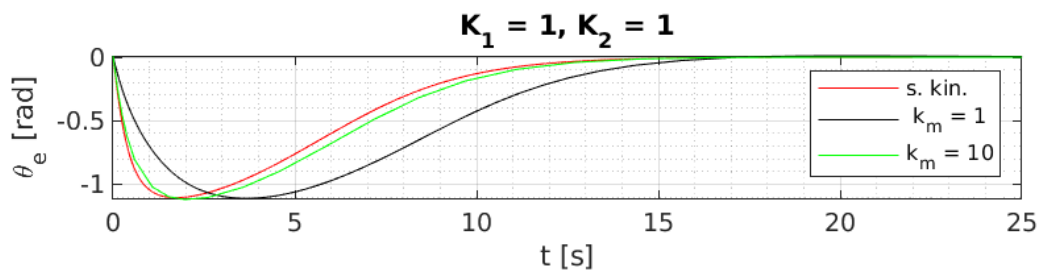
Rysunek 4.9: Rozwiązanie odniesienia oraz trajektoria rzeczywista, (a) $K_1 = 1, K_2 = 1, k_m = 1$, (b) $K_1 = 1, K_2 = 1, k_m = 10$



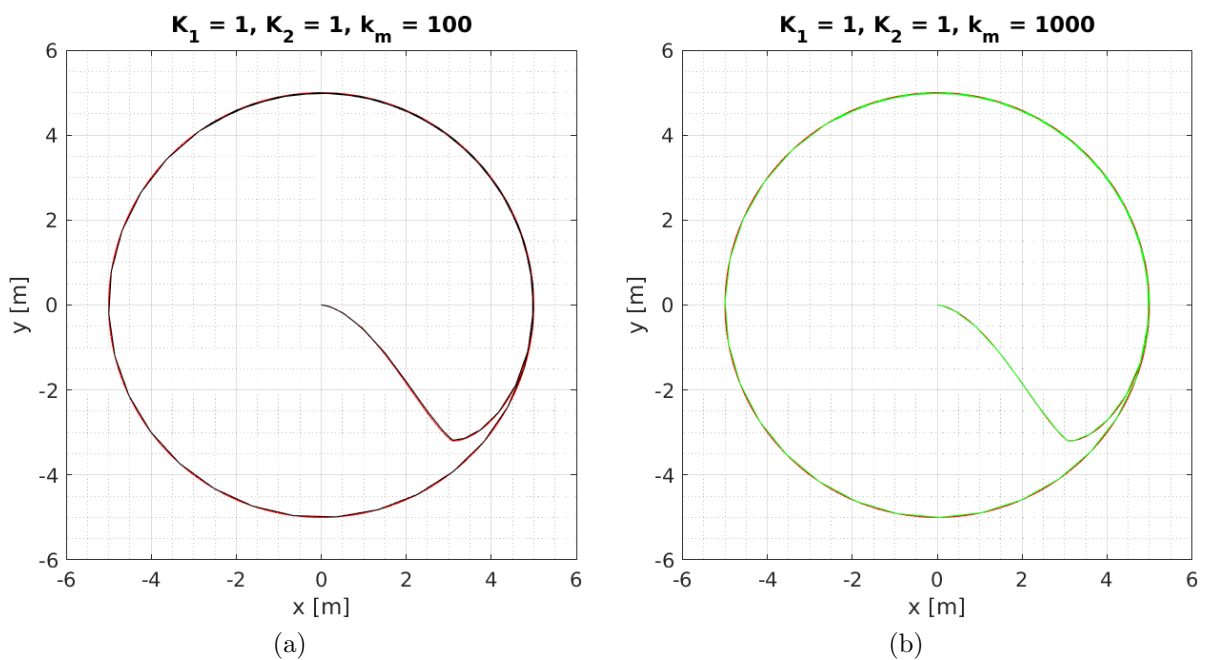
Rysunek 4.10: Błędy referencyjne x_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 1$ i $k_m = 10$



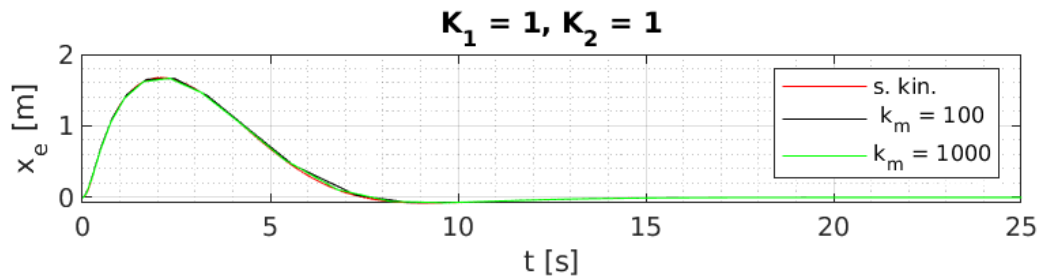
Rysunek 4.11: Błędy referencyjne y_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 1$ i $k_m = 10$



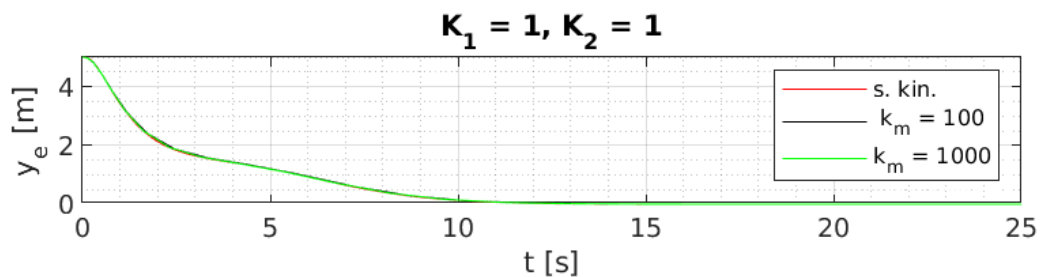
Rysunek 4.12: Błędy referencyjne θ_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 1$ i $k_m = 10$



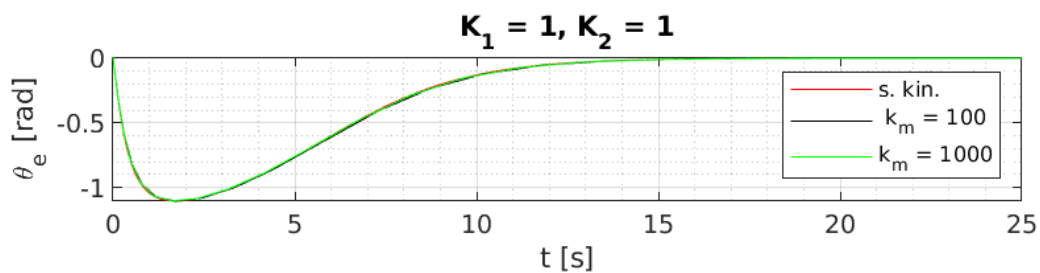
Rysunek 4.13: Rozwiązanie odniesienia oraz trajektoria rzeczywista, (a) $K_1 = 1, K_2 = 1, k_m = 100$, (b) $K_1 = 1, K_2 = 1, k_m = 1000$



Rysunek 4.14: Błędy referencyjne x_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 100$ i $k_m = 1000$



Rysunek 4.15: Błędy referencyjne y_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 100$ i $k_m = 1000$



Rysunek 4.16: Błędy referencyjne θ_e układu ze sterownikiem kinematycznym (rozwiązanie odniesienia) oraz rzeczywistego dla $k_m = 100$ i $k_m = 1000$

4.6.3 Podsumowanie

Na podstawie powyższych wykresów i wniosków widać, że odpowiednio duże wartości wzmocnienia sterownika dynamicznego sprowadzają błędy śledzenia praktycznie do zera. Problem pojawia się podczas dobierania nastaw sterownika kinematycznego. W zależności od stawianych wymagań (jak najszybszy czas osiągnięcia żądanej trajektorii, brak przesterowań, brak gwałtownych zmian kierunku jazdy) należy znaleźć odpowiednie wartości K_1 i K_2 .

Rozdział 5

Algorytm stabilizacji w punkcie dla monocykła

Poniższy rozdział dotyczy zadania sterowania do punktu – układ ma osiągnąć określoną konfigurację. Znalezienie odpowiedniego sterowania jest jednym z trudniejszych zadań robotyki. Trudności te zostały przedstawione przez Brocketta [2].

W tym rozdziale pokazane zostanie przekształcenie monocykła jako układu nieholonomicznego do postaci typowej oraz opis algorytmu Astolfiego wykorzystanego do sterowania robotem. Następnie pokazane zostaną wyniki przeprowadzonych badań.

5.1 Postać typowa układu

Układy robotyczne często przedstawia się za pomocą ogólnej postaci, tzw. postaci typowej. Dla układów liniowych postacią typową jest łańcuch integratorów (postać Brunovsky’ego), natomiast dla układów nieholonomicznych będzie to postać łańcuchowa. Została ona podana przez Murray’a i Sastry’ego [7] i wygląda następująco

$$\begin{cases} \dot{x}_1 = u_1, \\ \dot{x}_2 = u_2, \\ \dot{x}_3 = x_2 u_1, \\ \vdots \\ \dot{x}_n = x_{n-1} u_1, \end{cases} \quad (5.1)$$

gdzie n to liczba zmiennych stanu.

System przedstawiony w tej postaci jest w pełni sterowalny co oznacza, że dla każdej konfiguracji jest możliwe znalezienie pary sygnałów u_1 i u_2 przeprowadzających układ z dowolnego stanu początkowego do dowolnego stanu końcowego. Sprowadzenie układu do postaci typowej pozwala na sterowanie różnymi klasami robotów, z których każda ma własne, często skomplikowane ograniczenia nieholonomiczne, za pomocą jednego algorytmu. Wtedy sterowanie polega na transformacji kinematyki układu do postaci typowej, zastosowaniu konkretnego, ogólnego algorytmu oraz transformacji odwrotnej. Głównym problemem pozostaje znalezienie odpowiedniej transformacji, która musi być odwracalna.

5.2 Algorytm Astolfiego

Z pracy Brocketta wynika, że kinematyki układów nieholonomicznych nie można ustabilizować poprzez gładkie sprzężenie zwrotne zależne od stanu [2]. Astolfi zaproponował

rozwiązanie tego problemu – stabilizację poprzez nieciągłe sprzężenie zwrotne [1]. Założył on, że wartość u_1 jest stała i wynosi

$$u_1 = \tilde{k}.$$

Wtedy system można opisać równaniem macierzowym postaci [6]

$$\begin{pmatrix} \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \tilde{k} & 0 & 0 & \dots & 0 & 0 \\ 0 & \tilde{k} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \tilde{k} & 0 \end{bmatrix} \begin{pmatrix} x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u_2. \quad (5.2)$$

Następnie Astolfi zaproponował, by sterowanie u_2 wynosiło

$$u_2 = \tilde{p}_2 x_2 + \tilde{p}_3 x_3 + \dots + \tilde{p}_n x_n.$$

System (5.2) można więc zapisać jako

$$\dot{x}_{ob} = \mathbf{A} x_{ob},$$

gdzie:

$$\mathbf{x}_{ob} = (x_2 \ x_3 \ \dots \ x_n)^T, \\ \mathbf{A} = \begin{bmatrix} \tilde{p}_2 & \tilde{p}_3 & \tilde{p}_4 & \dots & \tilde{p}_{n-1} & \tilde{p}_n \\ \tilde{k} & 0 & 0 & \dots & 0 & 0 \\ 0 & \tilde{k} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \tilde{k} & 0 \end{bmatrix}.$$

Ponieważ zamknięta pętla systemu musi być asymptotycznie stabilna, zmienne \tilde{p} muszą zostać tak dobrane, aby wszystkie wartości własne macierzy \mathbf{A} znajdowały się w otwartej lewej półpłaszczyźnie zespolonej

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0 \quad \Rightarrow \quad \lambda_1, \dots, \lambda_n \in \mathbb{Z}_-,$$

$$p_A(\lambda) = \lambda^{n-1} - \tilde{p}_2 \lambda^{n-2} - \tilde{p}_3 \tilde{k} \lambda^{n-3} - \dots - \tilde{p}_{n-1} \tilde{k}^{n-3} \lambda - \tilde{p}_n \tilde{k}^{n-2}. \quad (5.3)$$

Jeśli założyć, że $\tilde{k} = -kx_1$, system w postaci łańcuchowej można zapisać jako

$$\begin{cases} \dot{x}_1 = -kx_1, \\ \dot{x}_2 = \tilde{p}_2 x_2 + \tilde{p}_3 x_3 + \dots + \tilde{p}_n x_n, \\ \dot{x}_3 = -kx_1 x_2, \\ \vdots \\ \dot{x}_n = -kx_1 x_{n-1}. \end{cases}, \quad (5.4)$$

z warunkami początkowymi $x(0) = (x_1(0) \ x_2(0) \ x_3(0) \ \dots \ x_n(0))$ i przy założeniu, że $x_1(0) \neq 0$. Aby zapewnić stabilność systemu wartości \tilde{p} muszą mieć postać

$$\tilde{p}_i = \frac{p_i}{x_1^{i-2}}, \quad \forall i = 2, \dots, n.$$

Podsumowując, zaproponowane przez Astolfiego sprzężenie zwrotne ma postać

$$\mathbf{u} = \begin{pmatrix} -kx_1 \\ p_2 x_2 + p_3 \frac{x_3}{x_1} + \dots + p_{n-1} \frac{x_{n-1}}{x_1^{n-3}} + p_n \frac{x_n}{x_1^{n-2}} \end{pmatrix}. \quad (5.5)$$

5.3 Zastosowanie algorytmu do monocykła nieholonomicznego

Robota klasy $(2, 0)$, o kinematyce opisanej równaniem (4.1), można przedstawić w postaci łańcuchowej jako układ równań

$$\begin{cases} \dot{z}_1 = u_1, \\ \dot{z}_2 = u_2, \\ \dot{z}_3 = z_2 u_1, \end{cases} \quad (5.6)$$

poprzez odwracalną transformację \mathbf{T}

$$\mathbf{T} = \begin{cases} z_1 = \theta, \\ z_2 = -x \cos \theta - y \sin \theta, \\ z_3 = -x \sin \theta + y \cos \theta, \end{cases} \quad (5.7)$$

oraz odpowiednie sprzężenie zwrotne. Wyznaczenie tego sprzężenia zwrotnego dla systemu przebiega następująco

$$\begin{cases} \dot{z}_1 = \dot{\theta} = \omega, \\ \dot{z}_2 = -\dot{x} \cos \theta + x \sin \theta \dot{\theta} - \dot{y} \sin \theta - y \cos \theta \dot{\theta}, \\ \dot{z}_3 = -\dot{x} \sin \theta - x \cos \theta \dot{\theta} + \dot{y} \cos \theta - y \sin \theta \dot{\theta}, \end{cases}$$

$$\begin{cases} \dot{z}_1 = \omega, \\ \dot{z}_2 = -\dot{\theta}(-x \sin \theta + y \cos \theta) - (v \cos^2 \theta + v \sin^2 \theta), \\ \dot{z}_3 = \dot{\theta}(-x \cos \theta - y \sin \theta) - v \sin \theta \cos \theta + v \sin \theta \cos \theta, \end{cases}$$

$$\begin{cases} \dot{z}_1 = \omega, \\ \dot{z}_2 = -\omega z_3 - v, \\ \dot{z}_3 = \omega z_2. \end{cases} \quad (5.8)$$

Układ równań (5.8) spełnia wymagania postaci typowej (5.6). Zmienne u wynoszą więc odpowiednio

$$\begin{cases} u_1 = \omega, \\ u_2 = -\omega z_3 - v = -u_1 z_3 - v. \end{cases} \quad (5.9)$$

Zgodnie z (5.5) wektor sterowań wynosi

$$\mathbf{u} = \begin{pmatrix} -kx_1 \\ p_2 x_2 + p_3 \frac{x_3}{x_1} \end{pmatrix}. \quad (5.10)$$

Wartości p_2 oraz p_3 wyznacza się z wielomianu charakterystycznego przekształconej macierzy \mathbf{A}

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0$$

$$\det \begin{bmatrix} \lambda - p_2 & -p_3 \\ -k & \lambda - k \end{bmatrix} = (\lambda - p_2)(\lambda - k) + p_3 k = 0,$$

$$\lambda^2 - k\lambda - p_2\lambda + p_2k + p_3k = 0,$$

$$\lambda^2 - \lambda(k + p_2) + k(p_2 + p_3) = 0. \quad (\Delta)$$

5.4 Badania symulacyjne

Podczas badań przyjęto, że punktem do którego robot ma dążyć jest początek układu współrzędnych $(0, 0)$. Wybrano wartości własne macierzy \mathbf{A} równe $\lambda_1 = \lambda_2 = -2$

$$(\lambda + 2)^2 = 0,$$

$$\lambda^2 + 4\lambda + 4 = 0. \quad (\square)$$

Porównując (\triangle) oraz (\square) wyliczone zostały parametry p

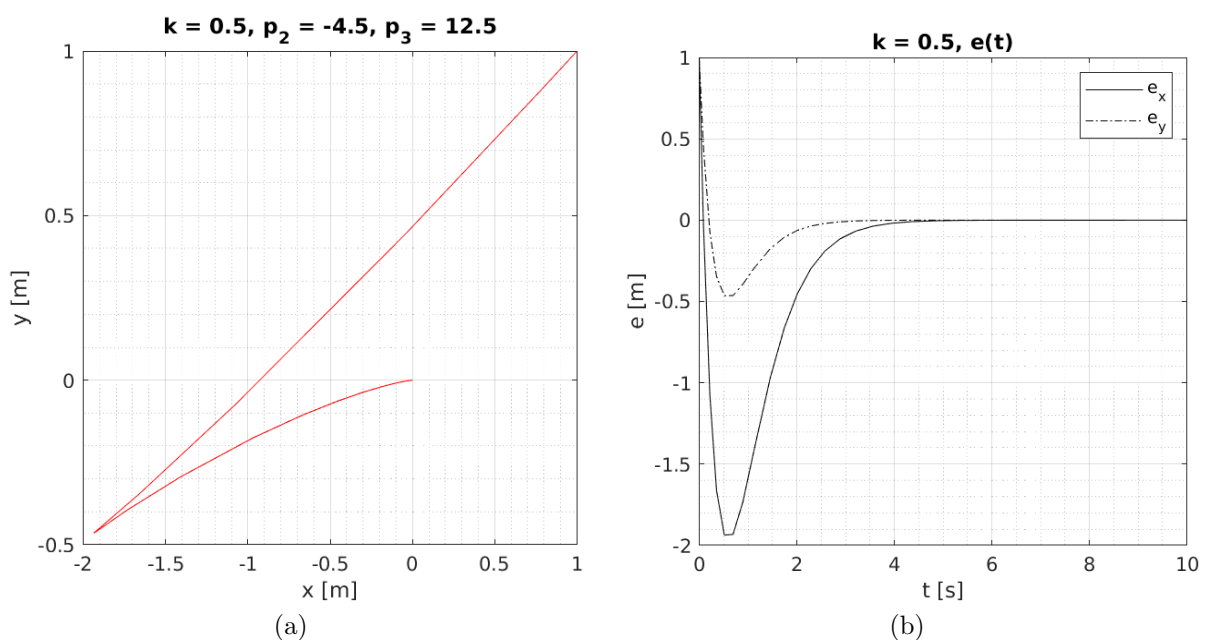
$$p_2 = -(4 + k) \quad \wedge \quad p_3 = \frac{4}{k} - p_2.$$

Początkowa konfiguracja robota wyglądała następująco

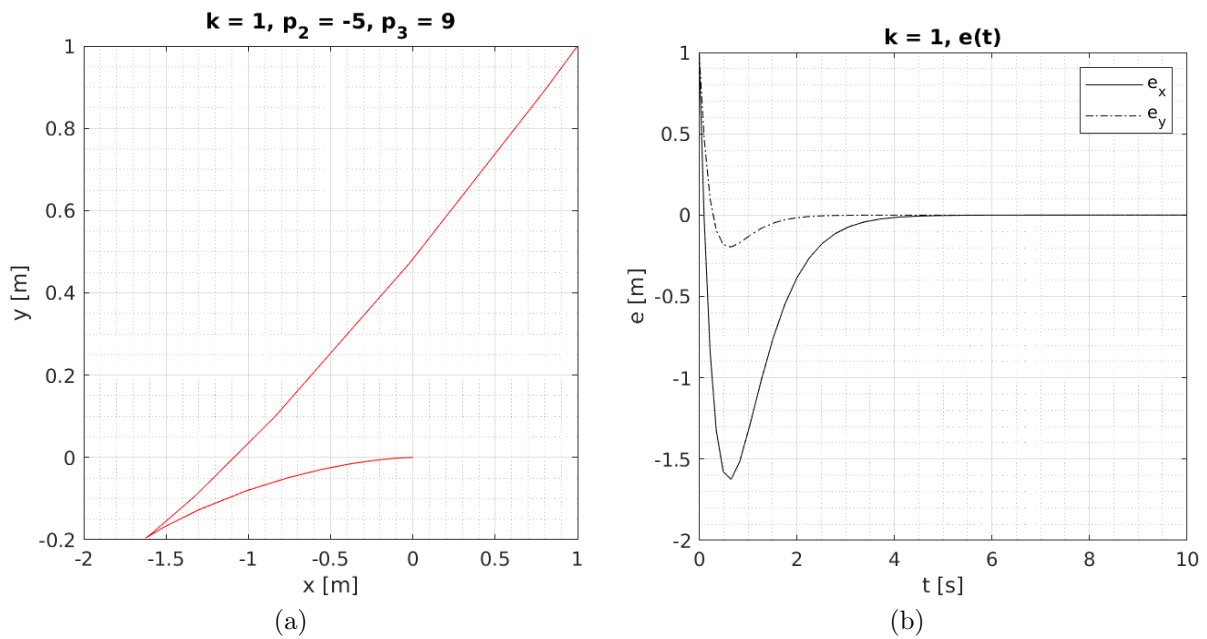
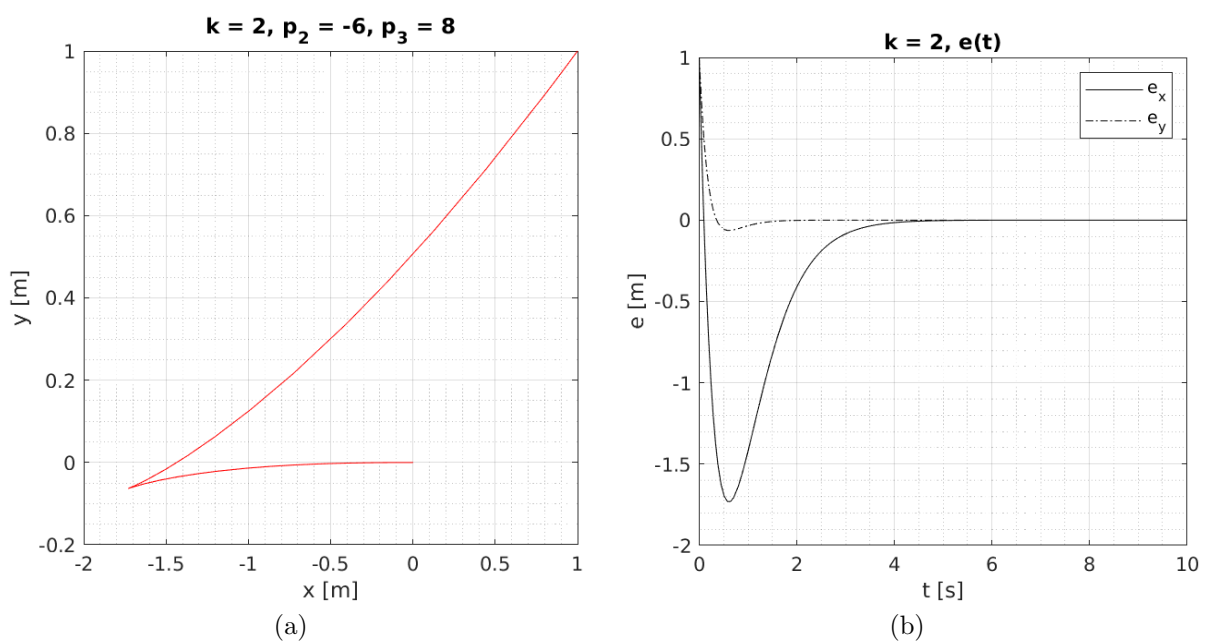
$$\begin{cases} x(0) = 1, \\ y(0) = 1, \\ \theta(0) = 0.5. \end{cases}$$

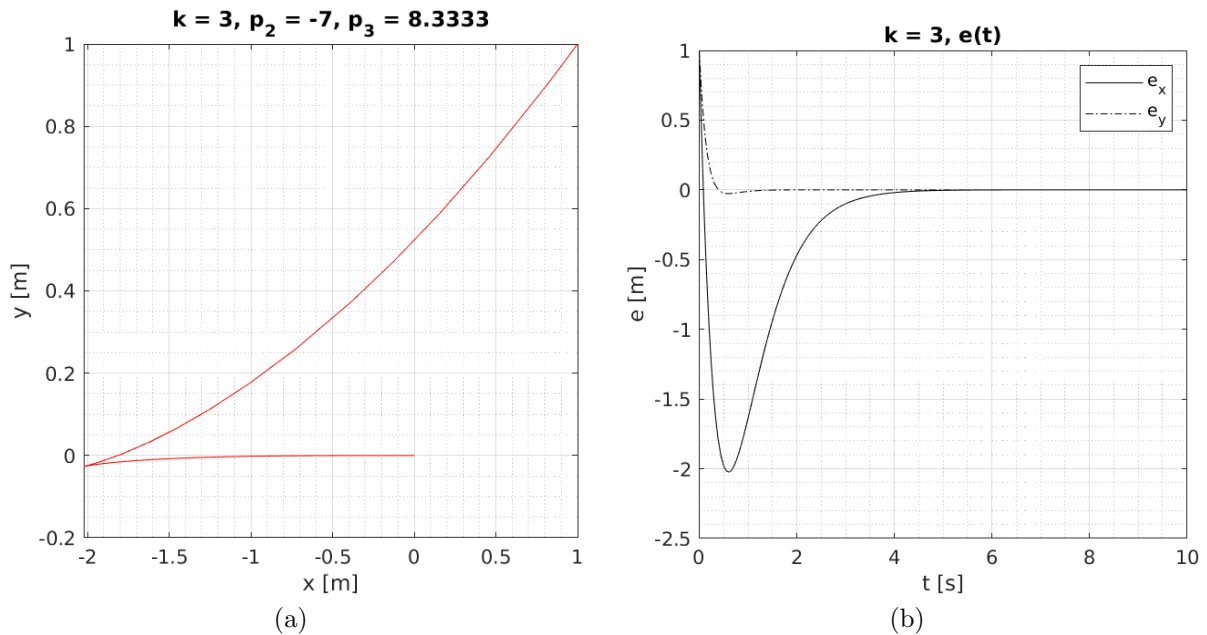
5.4.1 Wpływ wartości parametru k na zbieżność algorytmu

Analizując wykresy przebiegu błędów w czasie widać, że dla wszystkich wartości k czas osiągnięcia zadanego punktu jest identyczny. Dopiero przy dużych wartościach $k \geq 4$ pojawia się przeregulowanie obiektu wzdłuż osi x , co wydłuża czas zbieżności (rys. 5.5b). Parametr k , przy małych wartościach, wpływa na przebytą przez układ drogę. Zwiększanie tego parametru redukuje przemieszczanie się wzdłuż osi x , natomiast nie można takiej zależności określić dla osi y .

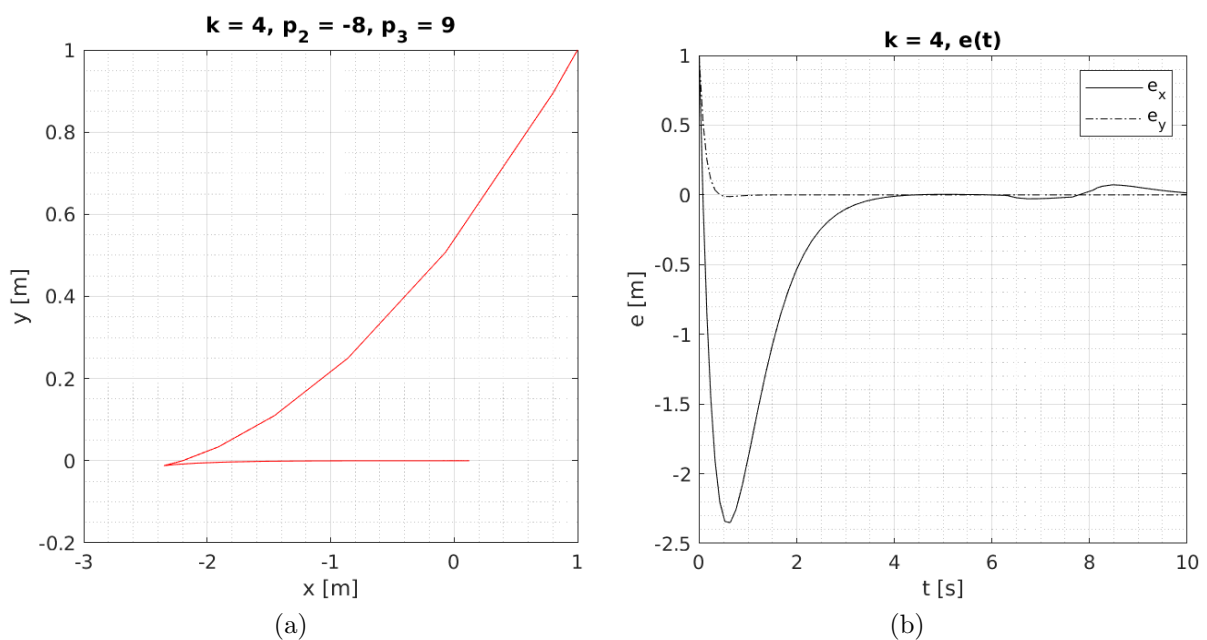


Rysunek 5.1: Parametr $k = 0.5$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń

Rysunek 5.2: Parametr $k = 1$; (a) przebyta przez układ trasa, (b) błędy przemieszczeńRysunek 5.3: Parametr $k = 2$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń



Rysunek 5.4: Parametr $k = 3$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń

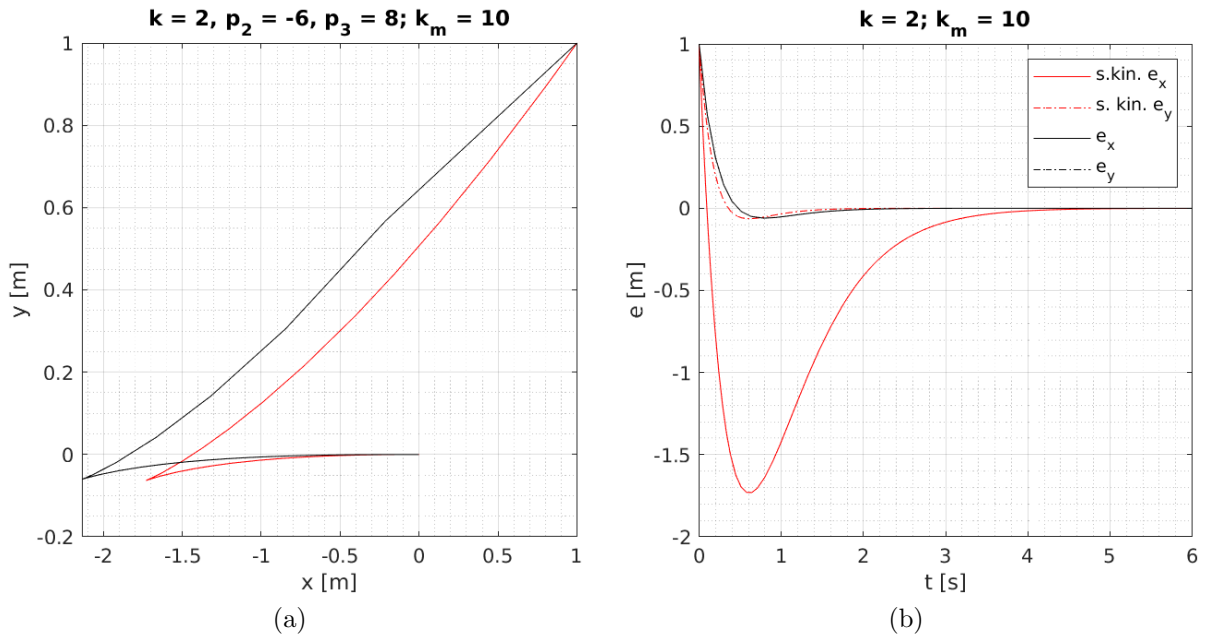


Rysunek 5.5: Parametr $k = 4$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń

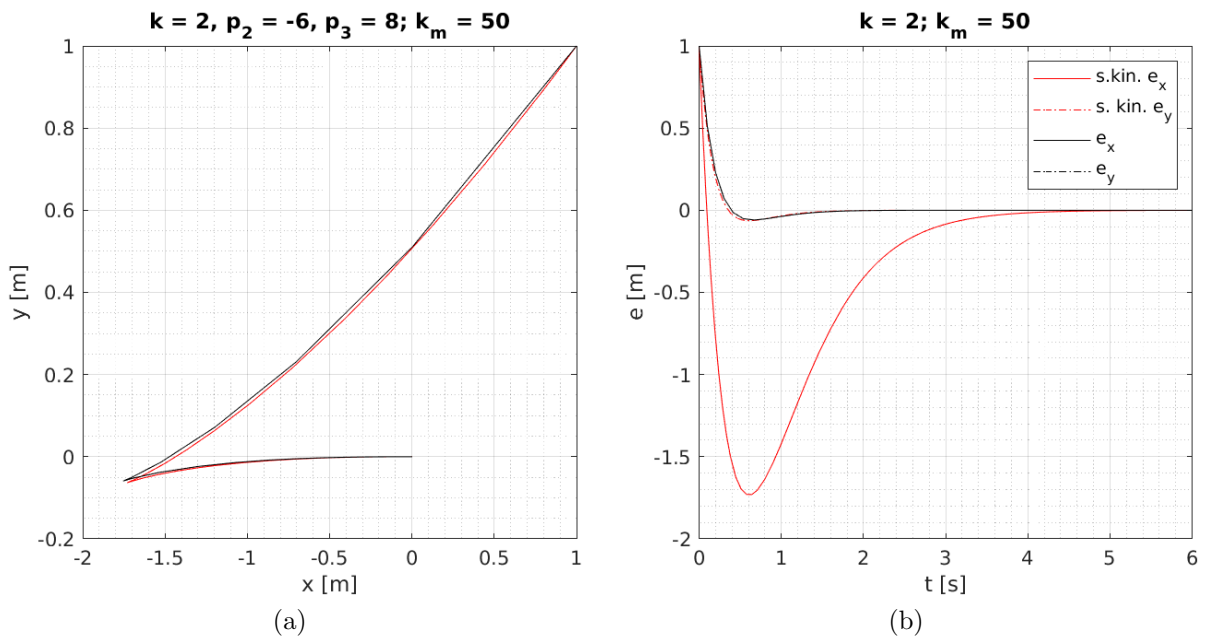
5.4.2 Wpływ wartości wzmocnienia k_m

Po podłączeniu do układu sterownika kinematycznego sprawdzono jak wartość wzmocnienia k_m wpływa na zbieżność algorytmu. Jako rozwiązanie wzorcowe przyjęto wykres przebytej drogi przy $k = 2$. Na poniższych wykresach zaznaczono je kolorem czerwonym.

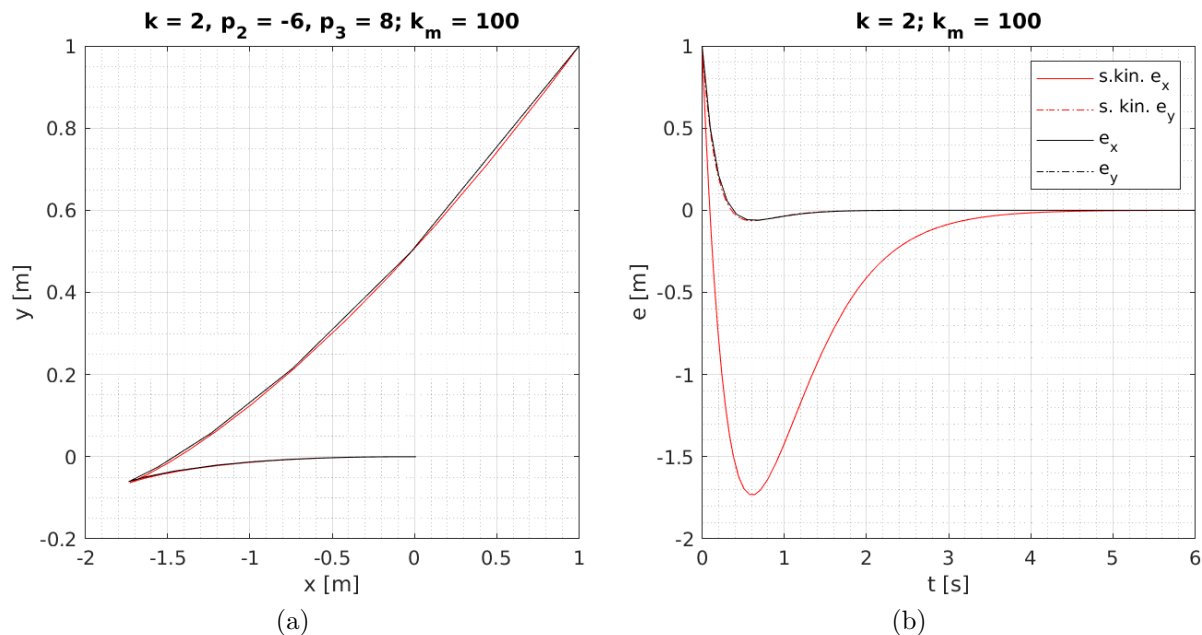
Układ stosunkowo szybko reaguje na wzrost wartości wzmocnienia – przy $k_m = 100$ rzeczywista przebyta trasa prawie pokrywa się z trasą wzorcową (rys. 5.8a). Analizując wykresy błędów widać, że niezależnie od k_m przemieszczenie wzdłuż osi x jest identyczne jak wzorcowe. Różnica pojawia się przy przemieszczeniu wzdłuż osi y .



Rysunek 5.6: Parametry $k = 2$, $k_m = 10$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń



Rysunek 5.7: Parametry $k = 2$, $k_m = 50$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń



Rysunek 5.8: Parametry $k = 2$, $k_m = 100$; (a) przebyta przez układ trasa, (b) błędy przemieszczeń

5.4.3 Podsumowanie

Podobnie jak w przypadku zadania śledzenia trajektorii, wybór odpowiednich wartości wzmocnienia w sterowniku kinematycznym, będzie zależeć od wymagań dotyczących trasy, którą robot pokonuje. Głównie trzeba zwrócić uwagę na pojawiające się przeregulowania wzdłuż osi – niektóre z nich można wyeliminować (oś x) lub zmniejszyć (oś y).

Rozdział 6

Algorytm śledzenia ścieżki dla monocykla

Ścieżka, to krzywa sparametryzowana odległością krzywoliniową s . Zadanie śledzenia ścieżki nie wymaga osiągnięcia przez robota określonej konfiguracji w danej chwili czasu, a jedynie podążanie wzdłuż zadanej krzywej. To oznacza, że prędkość układu może być dobrana w sposób umożliwiający realizację tych założeń [9].

Śledzenie ścieżki wymaga użycia innej metody opisu położenia układu niż zadanie śledzenia trajektorii. Jednym z najpopularniejszych sposobów jest użycie współrzędnych Freneta [4].

W tym rozdziale zostanie przedstawiony opis robota względem krzywej P , algorytm sterowania dla klasy $(2, 0)$ oraz przeprowadzone badania symulacyjne.

6.1 Opis układu w układzie Serreta-Freneta

Układ Freneta jest określony na krzywej $P(s)$ poprzez wersory normalny (x_n) i styczny do tej krzywej ($\frac{dr}{ds}$). Punkt M' jest rzutem środka masy platformy na P , a współrzędne punktu M względem układu Freneta to $(0, l)$, gdzie $l = \|MM'\|$ [5].

Ścieżkę P można opisać w zmiennych zależnych od czasu t jako

$$r(t) = (r_1(t), r_2(t))^T,$$

oraz za pomocą krzywizny $c(s)$

$$c(s) = \left\| \frac{d}{dr} \frac{dr}{ds} \right\| = \sqrt{\left(\frac{s^2 r_1}{ds^2} \right)^2 + \left(\frac{d^2 r_2}{ds^2} \right)^2}. \quad (6.1)$$

Analizując rysunek 6.1 widać, że

$$\frac{dr_1}{ds} = \left\| \frac{dr}{ds} \right\| \cos \theta_r, \quad \frac{dr_2}{ds} = \left\| \frac{dr}{ds} \right\| \sin \theta_r. \quad (6.2)$$

Równanie (6.1) można więc zapisać jako

$$c(s) = \sqrt{\left(-\sin \theta_r \frac{d\theta_r}{ds} \right)^2 + \left(\cos \theta_r \frac{d\theta_r}{ds} \right)^2}.$$
$$c(s) = \left| \frac{d\theta_r}{ds} \right| \Rightarrow \frac{d\theta_r}{dt} = \pm c(s) \dot{s}. \quad (6.3)$$

Podstawowym problemem podczas planowania zadania pozostaje zapewnienie ścieżki o odpowiednich właściwościach. Aby można było jednoznacznie zdefiniować punkt M' :

- krzywizna ścieżki $c(s)$ nie może być większa niż $\frac{1}{r_{min}}$, gdzie r_{min} to pewna liczba dodatnia,
- parametryzacja obowiązuje tylko w pobliżu ścieżki – jest lokalna.

6.2 Algorytm Samsona

Najczęściej przy projektowaniu sterownika kinematycznego realizującego zadanie śledzenia ścieżki wykorzystuje się algorytm Pometa [8]. Jego główną zaletą jest możliwość zastosowania do każdej klasy robotów nieholonomicznych. Cechuje go jednak powolna zbieżność oraz częsta zmiana kierunków ruchu. Pozwala on więc głównie na zbliżanie się do ścieżki. Aby układ podążał krzywą najskuteczniejsze jest skorzystanie z dedykowanego danej klasie algorytmu. Dla kołowych platform klasy $(2, 0)$ jest to algorytm Samsona [10].

Na podstawie wzorów (6.4) oraz (6.5) można przedstawić błędy śledzenia ścieżki jako

$$\begin{aligned} \dot{l} &= -\sin \theta_r \dot{x} + \cos \theta_r \dot{y} = \sin \theta - \theta_r v = \\ &= v \sin \tilde{\theta}, \\ \dot{s} &= \frac{\cos \tilde{\theta}}{1 \mp c(s)l} v, \\ \dot{\tilde{\theta}} &= \dot{\theta} - \dot{\theta}_r = \dot{\theta} - c(s)\dot{s} = \omega - \frac{c(s)}{1 \mp c(s)l} (\cos \theta_r \dot{x} + \sin \theta_r \dot{y}) = \\ &= \omega - \frac{c(s) \cos \tilde{\theta}}{1 \mp c(s)l} v = u, \end{aligned}$$

gdzie: $u = -k_2 l v_r \frac{\sin \tilde{\theta}}{\theta} - k_3 \tilde{\theta}$ to nowe sterowanie.

Algorytm Samsona wymaga, by prędkość liniowa v nie zanikała do zera. W dalszych rozważaniach przyjęto, że $v(t) = \text{const} \neq 0$. W takim przypadku równania sterownika kinematycznego zaproponowane przez Samsona wynoszą

$$v_{ref} = \text{const}, \tag{6.6}$$

$$\omega_{ref} = -k_2 l v_{ref} \frac{\sin \tilde{\theta}}{\theta} - k_3 \tilde{\theta} + v_{ref} \cos \tilde{\theta} \frac{c(s)}{1 \mp c(s)l}, \tag{6.7}$$

gdzie $k_2, k_3 > 0$ to nastawy regulatora.

6.3 Badania symulacyjne

We wszystkich przeprowadzonych badaniach zadaną ścieżką był okrąg o środku w początku układu współrzędnych $(0, 0)$ oraz o promieniu $r = 1$. Początkową konfigurację robota przyjęto jako

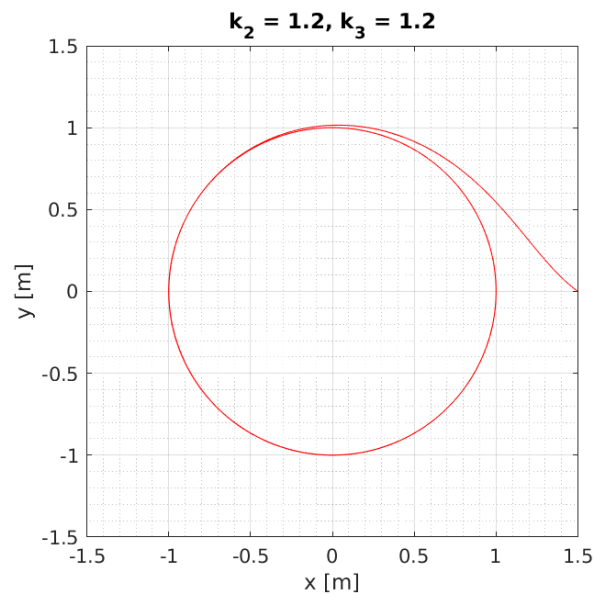
$$\begin{cases} x(0) = 1.5, \\ y(0) = 0, \\ \theta(0) = 0.8\pi. \end{cases}$$

Dla takich warunków parametry ścieżki wynosiły

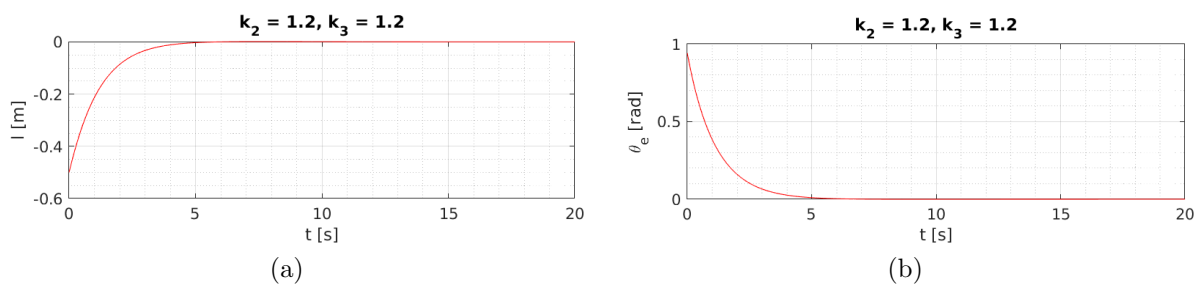
$$\begin{cases} l(0) = -0.5, \\ \theta_r(0) = 0.5\pi, \\ \tilde{\theta} = \theta - \theta_r = 0.3\pi, \\ v_r = \text{const} = 0.5. \end{cases}$$

6.3.1 Nastawy sterownika kinematycznego

Najlepsze wyniki zbieżności algorytmu otrzymano dla wartości $k_2 = k_3 = 1.2$. Była to optymalna konfiguracja zapewniająca odpowiednie reagowanie na poszczególne błędy. Wykresy 6.2–6.3 przedstawiają przebytą drogę i błędy wielkości opisujących układ.



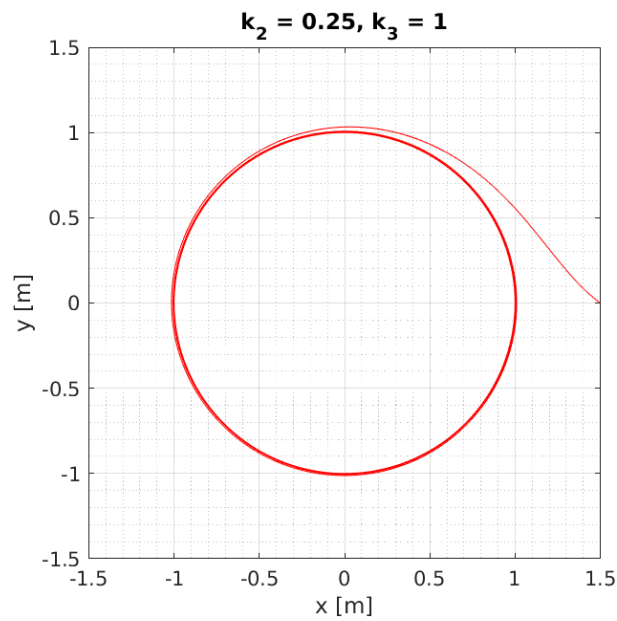
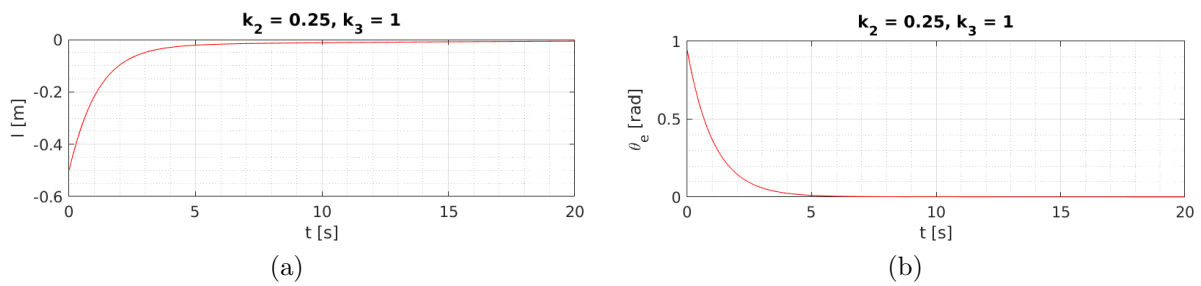
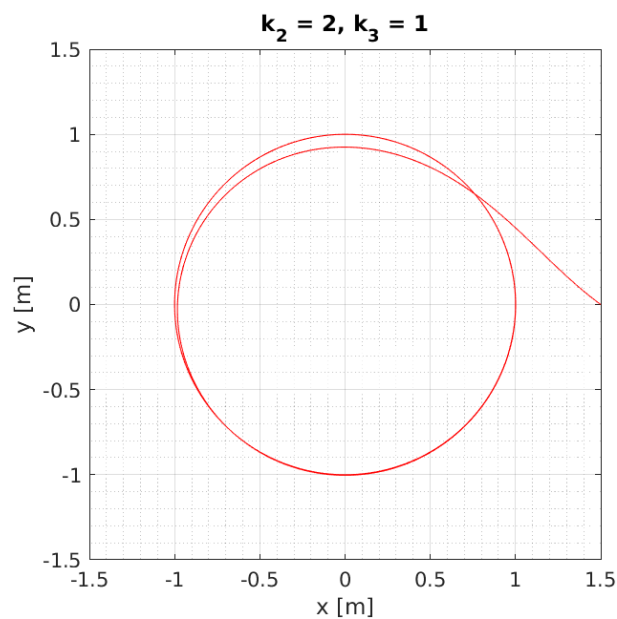
Rysunek 6.2: Trajektoria referencyjna układu dla $k_2 = 1.2$ i $k_3 = 1.2$

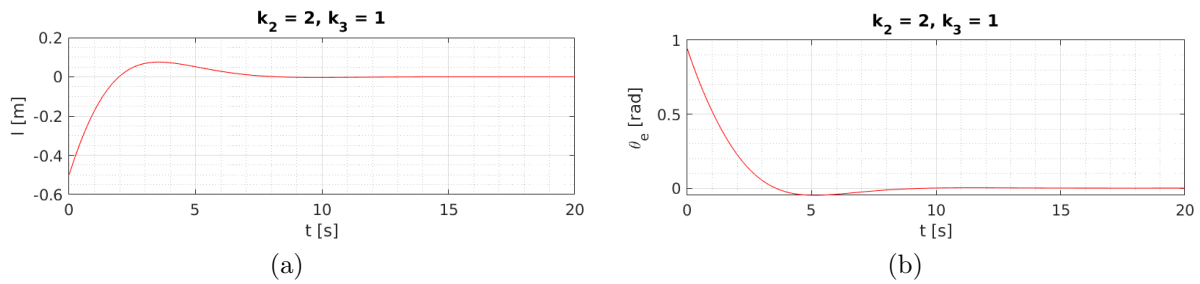


Rysunek 6.3: Parametry $k_2 = 1.2$ i $k_3 = 1.2$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$

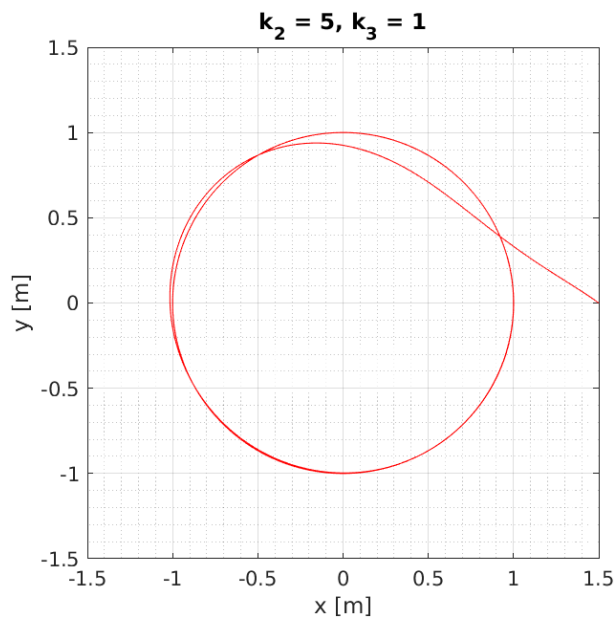
Zmienna wartość k_2

Zgodnie ze wzorem (6.7) wielkość k_2 wpływa głównie na błąd odległości robota od ścieżki (dla małych wartości kąta θ wyrażenie $\frac{\sin \theta}{\theta} \approx 1$). Im większa odległość robota od ścieżki, tym większa ω . Kiedy wzmocnienie jest małe układ dojeżdża w pobliże ścieżki, a następnie bardzo wolno na nią wjeżdża. Duże wartości powodują, że robot zbliża się pod zbyt dużym kątem, mijają ścieżkę i dopiero potem na nią wjeżdża. Poprawia to jednak zbieżność algorytmu.

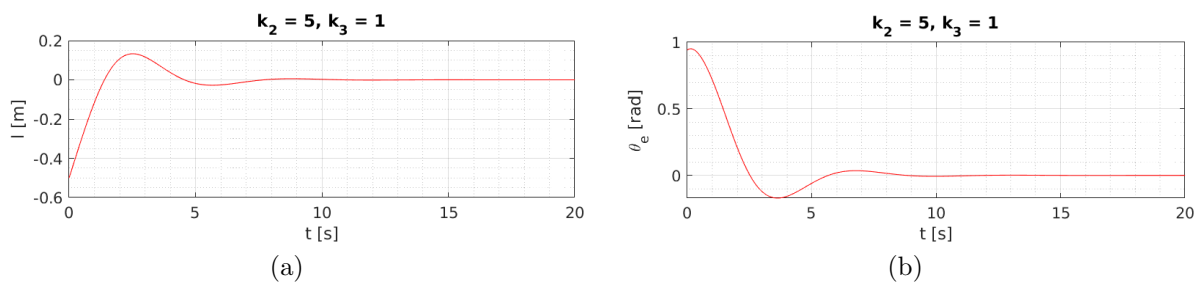
Rysunek 6.4: Trajektoria referencyjna układu dla $k_2 = 0.25$ i $k_3 = 1$ Rysunek 6.5: Parametry $k_2 = 0.25$ i $k_3 = 1$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$ Rysunek 6.6: Trajektoria referencyjna układu dla $k_2 = 2$ i $k_3 = 1$



Rysunek 6.7: Parametry $k_2 = 2$ i $k_3 = 1$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$



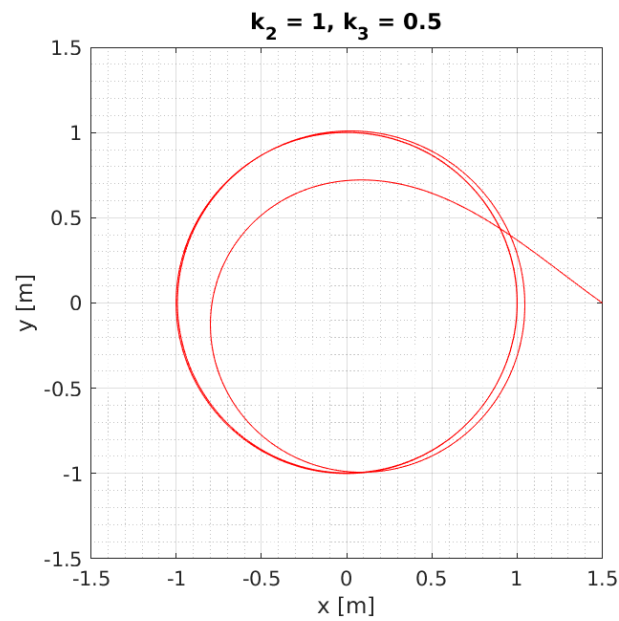
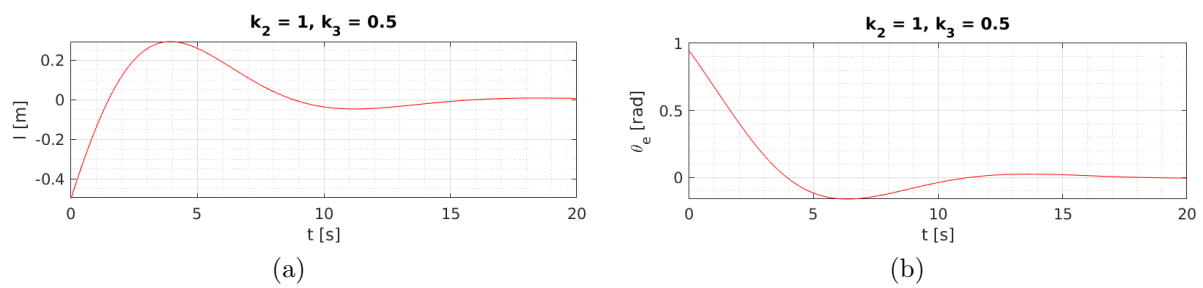
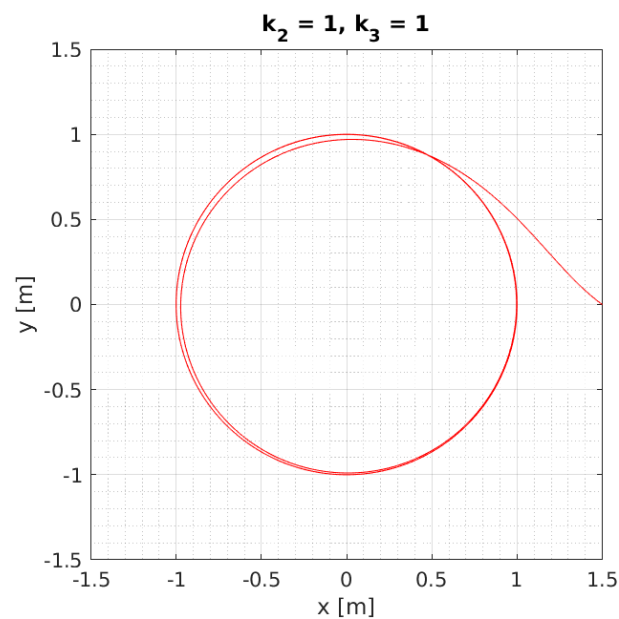
Rysunek 6.8: Trajektoria referencyjna układu dla $k_2 = 5$ i $k_3 = 1$

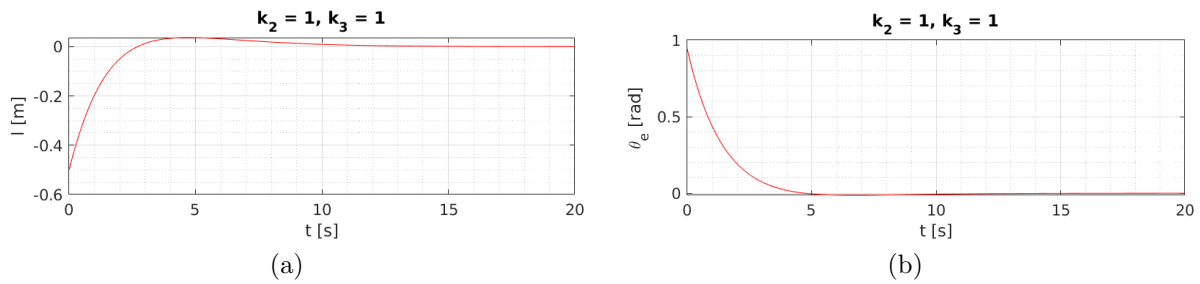


Rysunek 6.9: Parametry $k_2 = 5$ i $k_3 = 1$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$

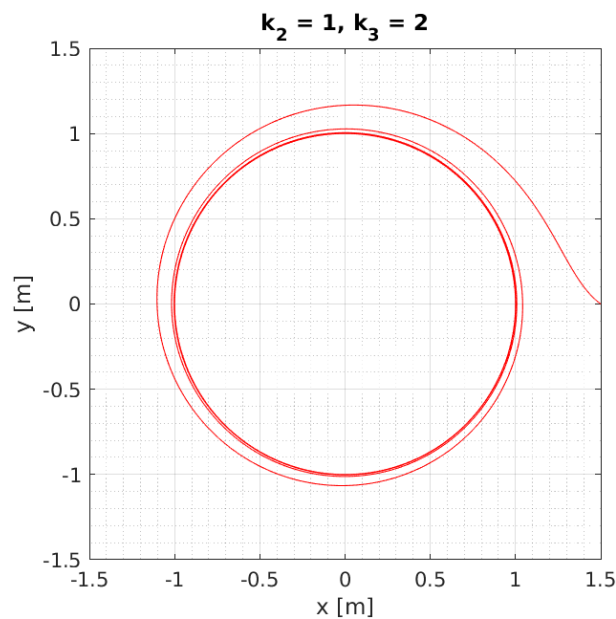
Zmienna wartość k_3

Wzmocnienie k_3 odpowiedzialne jest za reagowanie na błąd orientacji. Im większa będzie ta wartość, tym mniejszą prędkość kątową osiąga układ. Oznacza to, że przy małych k_3 robot na początku ostro koryguje swoje położenie w konsekwencji czego zbliża się do zadanej ścieżki od wewnątrz. Duże wartości tego parametru powodują, że układ porusza się ruchem spiralnym wjeżdżając na ścieżkę od zewnątrz. Wydłuża to tym samym czas zbieżności.

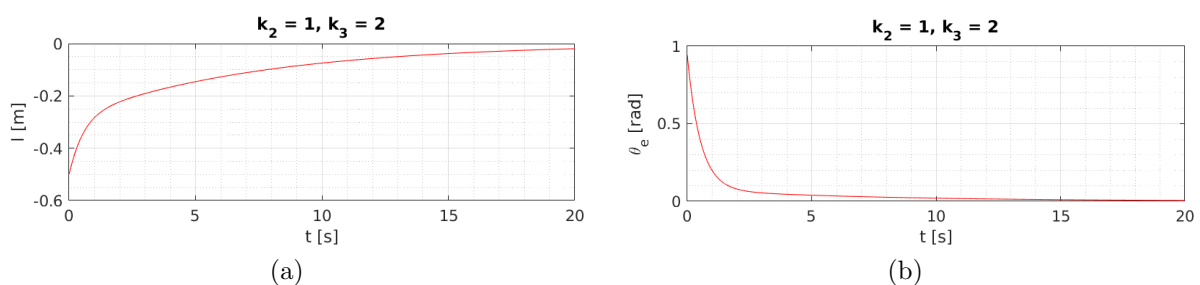
Rysunek 6.10: Trajektoria referencyjna układu dla $k_2 = 1$ i $k_3 = 0.5$ Rysunek 6.11: Parametry $k_2 = 1$ i $k_3 = 0.5$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$ Rysunek 6.12: Trajektoria referencyjna układu dla $k_2 = 1$ i $k_3 = 1$



Rysunek 6.13: Parametry $k_2 = 1$ i $k_3 = 1$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$



Rysunek 6.14: Trajektoria referencyjna układu dla $k_2 = 1$ i $k_3 = 2$

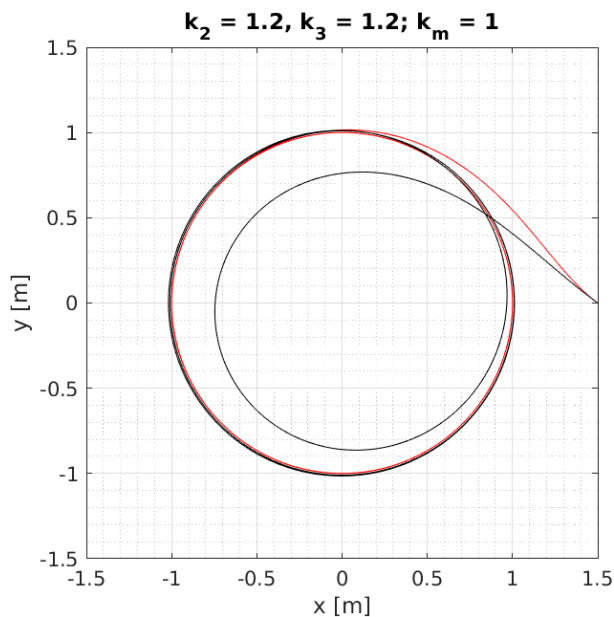
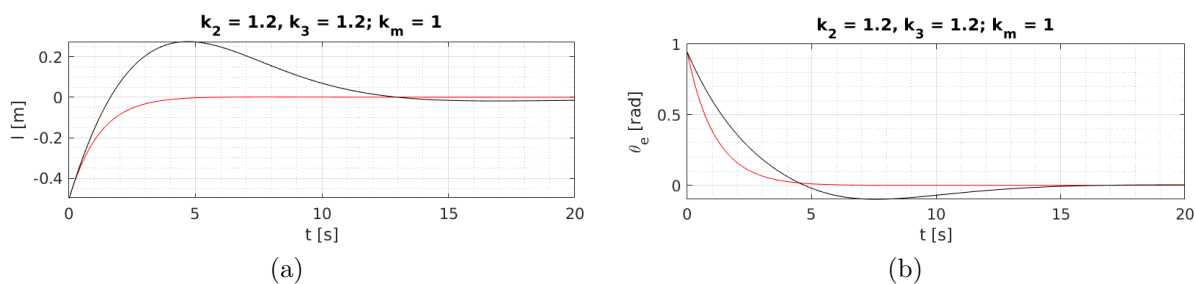
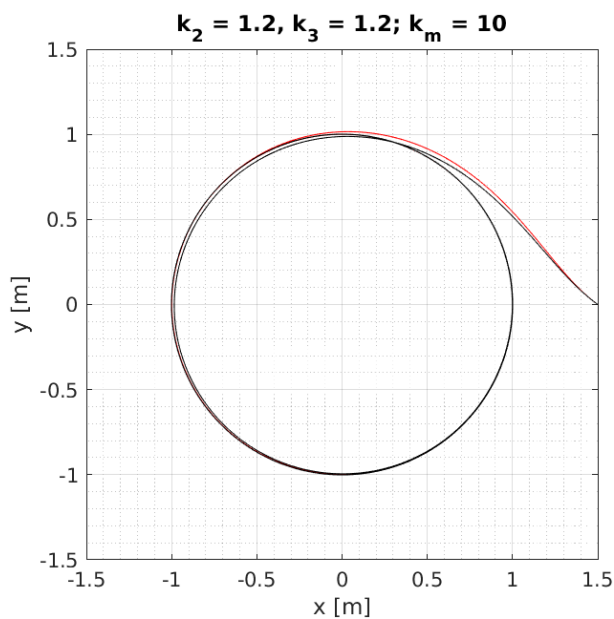


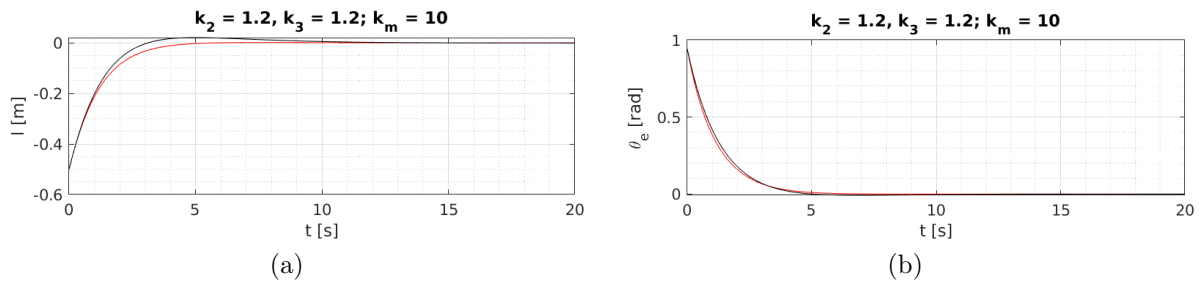
Rysunek 6.15: Parametry $k_2 = 1$ i $k_3 = 2$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$

6.3.2 Wpływ sterownika dynamicznego na zbieżność algorytmu

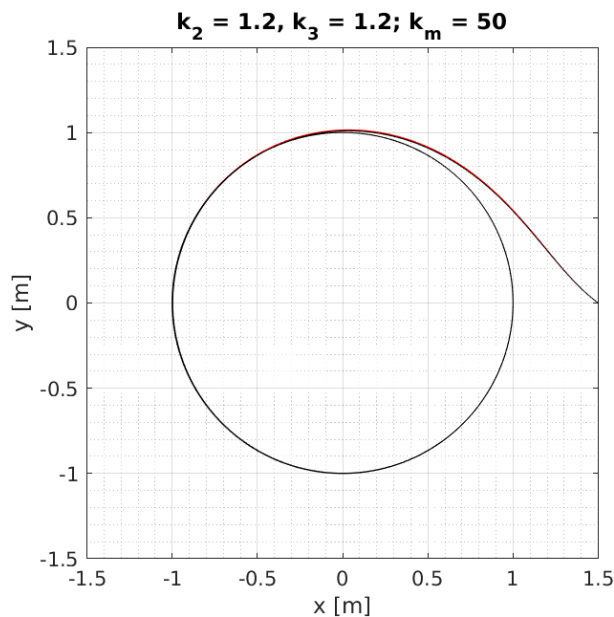
Sprawdzono zachowanie obiektu dla różnych wartości wzmocnienia k_m . Jako rozwiązanie wzorcowe przyjęto drogę przebytą przez obiekt przy nastawach $k_2 = k_3 = 1.2$. Na poniższych wykresach zostało ono zaznaczone kolorem czerwonym.

Analizując wykresy 6.16-6.21 wyraźnie widać, że im większa wartość k_m tym lepiej układ reaguje. Przy wartości $k_m = 50$ rzeczywista trasa robota jest prawie identyczna z wzorcową.

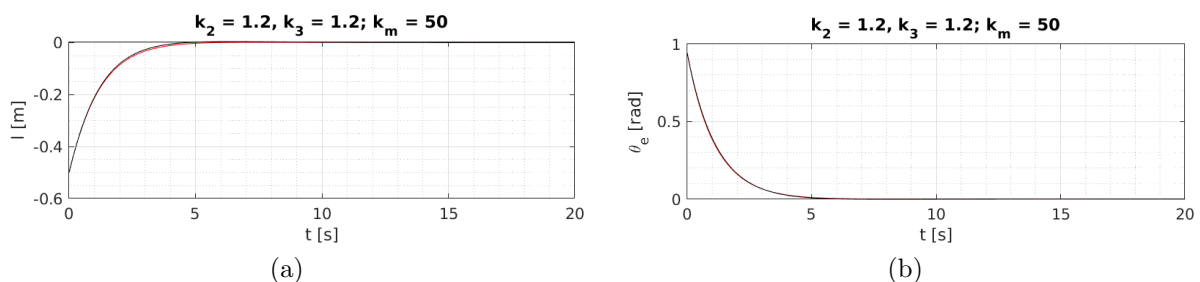
Rysunek 6.16: Trajektoria referencyjna oraz rzeczywista dla $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 1$ Rysunek 6.17: Parametry $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 1$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$ Rysunek 6.18: Trajektoria referencyjna oraz rzeczywista dla $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 10$



Rysunek 6.19: Parametry $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 10$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$



Rysunek 6.20: Trajektoria referencyjna oraz rzeczywista dla $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 50$



Rysunek 6.21: Parametry $k_2 = 1.2$, $k_3 = 1.2$ i $k_m = 50$, (a) błąd położenia l , (b) błąd orientacji $\tilde{\theta}$

6.3.3 Podsumowanie

Ponownie dobór wartości k_2 oraz k_3 będzie zależeć od rezultatów, które mają zostać osiągnięte. Aby rozwiązanie było optymalne, należy zastanowić się z której strony robot ma wjechać na ścieżkę, czy może ją przekroczyć lub pod jakim kątem ma się do niej zbliżać. Wartości trzeba zawsze dopasować do konkretnego przypadku.

Rozdział 7

Podsumowanie

Celem pracy było zaznajomienie się z robotem mobilnym klasy $(2, 0)$, zadaniami, które może wykonywać oraz algorytmami, które tą realizację zapewniają.

Model obiektu został przedstawiony we współrzędnych uogólnionych, z których następnie wyliczono opis we współrzędnych pomocniczych. To właśnie te współrzędne wykorzystuje się przy projektowaniu sterownika kinematycznego. W poszczególnych algorytmach zaistniała konieczność wyrażenia modelu przy pomocy postaci typowej (algorytm Astolfiego wykorzystywany do sterowania do punktu) oraz we współrzędnych Freneta (algorytm Samsona do śledzenia ścieżki).

Dla każdego z algorytmów przeprowadzono symulacje pozwalające określić wpływ nastaw sterowników na zbieżność algorytmu. W każdym zadaniu wykorzystano identycznie zbudowany sterownik dynamiczny. Pozwoliło to wyciągnąć wniosek, że im większą wartość ma wzmocnienie tego sterownika, tym bardziej rzeczywiste rozwiązanie zbliża się do wzorcowego wygenerowanego ze sterownika kinematycznego. Jedne algorytmy wymagały wartości wzmocnienia na poziomie 50, inne 1000.

Niektóre wartości nastaw sterownika kinematycznego powodowały błędy obliczeniowe w programie *Matlab*. Warto byłoby wypróbować inne opcje symulacji, takie jak zmiana algorytmu obliczeniowego czy wielkości kroku pomiędzy kolejnymi punktami dla których liczona jest wartość. Z powodu tych problemów możliwe było tylko ogólne przybliżenie wartości wzmocnień, dla których zbieżność algorytmu jest optymalna.

Zalecane byłoby również zaprojektowanie sterowników dla innych algorytmów rozwiązujących postawione zadania i porównanie ich działania.

W pracy korzystano z szerokiego zakresu literatury; starano się w miarę możliwości sięgać do oryginalnych prac autorów poszczególnych algorytmów.

Cel pracy został zrealizowany.

Literatura

- [1] A. Astolfi. Exponential stabilization of a car-like vehicle. *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 2:1391–1396, 1995.
- [2] R. W. Brockett. Asymptotic stability and feedback stabilization. In *Differential Geometric Control Theory*, volume 27, pages 181–191. Birkhauser, 1983.
- [3] G. Campion, G. Bastin, and B. D’Andrea-Novel. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *Proceedings IEEE International Conference on Robotics and Automation*, 1:462–469, 1993.
- [4] F. Frenet. Sur les courbes à double courbures. *Journal de mathématiques pures et appliquées 1 série*, 17:437–447, 1852.
- [5] A. Mazur. *Sterowanie oparte na modelu dla nieholonomicznych manipulatorów mobilnych*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2009.
- [6] A. Mazur and J. Płaskonka. Different kinematic controllers stabilizing nonholonomic mobile manipulators of (nh, nh) type about desired configuration. *Journal of Automation Mobile Robotics and Intelligent Systems*, 5(3):41–48, 2011.
- [7] R. M. Murray and S. S. Sastry. Steering nonholonomic systems in chained form. In *Proceedings of the 30th IEEE Conference on Decision and Control*, volume 2, pages 1121–1126, 1991.
- [8] J. B. Pomet. Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift. *Systems and Control Letters*, 18:147–158, 1992.
- [9] J. Płaskonka. *Algorytmy śledzenia ścieżki dla nieholonomicznych manipulatorów mobilnych*. PhD thesis, 2010.
- [10] C. Samson. Path following and time-varying feedback stabilization of a wheeled mobile robot. *Second International Conference on Automation, Robotics and Computer Vision*, 3, 1992.
- [11] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. volume 2, pages 1136–1141, 1991.
- [12] K. Tchoń, A. Mazur, R. Hossa, I. Dulęba, and R. Muszyński. *Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 2000.
- [13] Y. Yamamoto and X. Yun. Effect of the dynamic interaction on coordinated control of mobile manipulators. *IEEE Transactions on Robotics and Automation*, 12(5):816–824, 1996.