

Spis treści

1. Wprowadzenie	3
1.1. Cel	3
1.2. Zakres pracy	3
1.3. Zawartość pracy	3
2. Konstrukcja mechaniczna	5
2.1. Rama i rozlokowanie podzespołów	5
2.2. Silniki	5
3. Urządzenie elektroniczne	7
3.1. Sterownik	7
3.2. Żyroskop	7
3.3. Akcelerometr	8
3.4. Sterownik silników krokowych	8
3.5. Zasilanie	9
4. Modelowanie i identyfikacja	11
4.0.1. Model nieliniowy	11
4.0.2. Model liniowy	13
4.0.3. Symulacja oraz wizualizacja wyników	14
5. Sterowanie predykcyjne	17
5.1. Sterowanie predykcyjne na podstawie równań stanu	17
5.2. Kontrola predykcyjna z pojedynczym oknem optymalizacji	18
5.3. Optymalizacja	19
5.4. Kontrola Predykcyjna z ograniczeniami	20
6. Optymalizacja sterowania	23

1. Wprowadzenie

1.1. Cel

W ramach pracy inżynierskiej powstał dwukołowy robot balansujący. Weryfikacja przyjętych wtedy założeń projektowych uwidoczniła wiele potencjalnych udoskonaleń.

1.2. Zakres pracy

Celem pracy jest budowa robota dwukołowego oraz implementacja algorytmu predykcyjnego stabilizującego robota w górnym, niestabilnym punkcie równowagi. Konstrukcja mechaniczna będzie bazować na robocie wykonanym w ramach pracy inżynierskiej, zostanie ona przebudowana i dopracowana.

Algorytm będzie rozwiązywał w każdym kroku, odpowiednio sformułowane zadanie programowania kwadratowego. Jednym z podstawowych elementów pracy będzie też modelowanie oraz identyfikacja modelu matematycznego robota. Algorytm będzie zaimplementowany na platformie STM32F3 Discovery. Zmiana mikrokontrolera wymaga dostosowanie sterownika do pracy w nowych warunkach i napisanie kodu źródłowego.

1.3. Zawartość pracy

W rozdziale ?? przedstawiono....

2. Konstrukcja mechaniczna

W rozdziale zawarto opis konstrukcji ramy pojazdu oraz jednostki napędowej. Założenia projektowe części mechanicznej to uzyskanie maksymalnej trwałości przy utrzymaniu niskich kosztów pojazdu oraz możliwość łatwej modyfikacji i napraw.

2.1. Rama i rozlokowanie podzespołów

Podstawowym założeniem podczas projektu ramy robota oraz rozlokowywaniu komponentów było zabezpieczenie przed uszkodzeniami wynikającymi z ewentualnych upadków. Rama została wykonana z płyty pleksiglasowej o grubości 4mm. Pozwoliło to utrzymać niską masę przy stosunkowo dużej wytrzymałości konstrukcji. Poszczególne części sterownika zostały przykręcone do ramy w górnej części pojazdu i zabezpieczone przed uderzeniami dodatkowymi fragmentami ramy.

Zastosowanie dwóch kul podporowych, po obu stronach robota, pozwala także na jazdę w pozycji poziomej. Akumulatory zostały umieszczone w dolnej części pojazdu, rozwiązanie to znacznie obniża środek ciężkości. Pozwala to na bardziej dynamiczną jazdę lecz utrudnia stabilizację w górnym punkcie równowagi. W kolejnych podpunktach omówione zostaną kluczowe elementy mechaniczne konstrukcji. Zdjęcie pogładowe robota zostało przedstawione na rysunku 2.1

Rys. 2.1. Zdjęcie pogładowe robota. Zdjęcie własne

2.2. Silniki

Jako element napędowy robota zastosowano dwa silniki krokowe 42HS40-0504. Stosunkowo duży moment obrotowy przy niskiej prędkości obrotowej, możliwość kontroli pozycji w pętli otwartej oraz możliwość częstej zmiany kierunku czynią silniki krokowe zasadnym wyborem do pojazdu tego typu. Zastosowane w poprzedniej iteracji projektu silniki zostały jednak wymienione na silniki hybrydowe o większym momencie obrotowym oraz wyższej liczbie kroków na obrót. Podstawowe parametry wykorzystanych silników przedstawione zostały w tabeli 2.1

Model	42HS40-0504
Liczba kroków	200
Napięcie znamionowe	12V
Pobór prądu na cewkę	0,5A
Rezystancja cewki	24Ohm
Moment trzymający	0,44Nm
Wymiary bez wału	42 x 42 x 40mm

Tabela 2.1. Parametry silników

3. Urządzenie elektroniczne

Rozdział zawiera opis części elektronicznej robota z uwzględnieniem mikro kontrolera, czujników inercyjnych oraz interfejsu mocy do silników krokowych. W dalszych podrozdziałach opisane zostały najważniejsze właściwości użytych podzespołów sterownika oraz sposób ich integracji.

3.1. Sterownik

Sterownik robota oparty jest o płytke ewaluacyjną STM32F3 Discovery z mikroprocesorem STM32F303VCT6. Rozwiązanie to udostępnia znaczną moc obliczeniową i pozwala na zastosowanie bardziej zaawansowanych algorytmów sterujących. Najważniejsze cechy modułu STM32F3 Discovery:

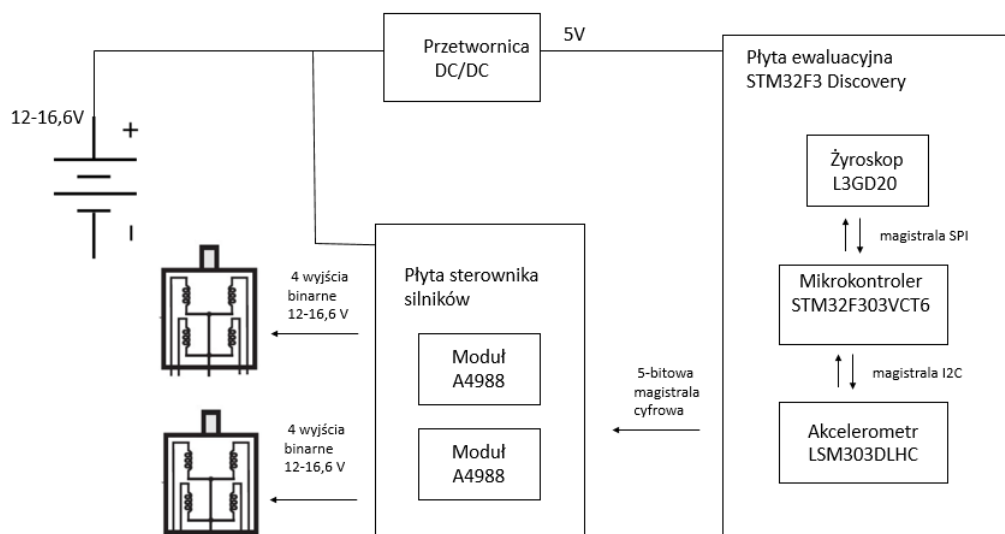
- Rdzeń: Cortex M4F
- Taktowanie 72 MHz
- 256 kB Flash
- 48kB RAM
- 3- osiowy, cyfrowy żyroskop L3GD20
- 3- osiowy, cyfrowy akcelerometr z magnetometrem LSM303DLHC
- 10 diod LED
- Dwa przyciski

Wykorzystanie elementów umieszczonych na płytce ewaluacyjnej pozwala zredukować wykorzystanie dodatkowych podzespołów o niezbędne w projekcie czujniki inercyjne oraz ułatwiające proces projektowy diody i przyciski. Schemat demonstrujący sposób komunikacji podzespołów sterownika znajduje się na rysunku 3.1

3.2. Żyroskop

Wykorzystane żyroskop to zawarty na płytce ewaluacyjnej układ L3GD20. Jest to trójosiowy żyroskop cyfrowy. Jego kluczowe właściwości to:

- Trzy dostępne skale pomiarowe (250/500/2000 dps)
- Interfejs komunikacyjny I^2C lub SPI
- 16 bitowa rozdzielczość pomiarów



Rys. 3.1. Schemat sterownika. Grafika własna

W projekcie komunikacja z czujnikiem przebiega przy użyciu protokołu SPI. Wykorzystywana skala pomiarowa to: 500 dps.

3.3. Akcelerometr

Kolejnym wykorzystanym elementem płytki ewaluacyjnej jest trójosiowy akcelerometr z magnetometrem LSM303DLHC.

- Interfejs komunikacyjny: I2Cm
- Akcelerometr, zakres: ± 2 g, ± 4 g, ± 8 g oraz ± 16 g
- Magnetometr, zakres: od $\pm 1,3$ do $\pm 8,1$ gauss
- 16 bitowa rozdzielczość pomiarów

Sensor został skonfigurowany na zakres ± 2 g. Magnetometr nie został wykorzystany ze względu na znaczne zakłócenia wskazań poprzez niewielką odległość od silników.

3.4. Sterownik silników krokowych

Do kontroli silników krokowych wykorzystany został sterownik stworzony na potrzeby pracy inżynierskiej. Jest on oparty o 2 układy A4988, każdy z nich pozwala sterować pojedynczym silnikiem za pomocą dwóch wyprowadzeń cyfrowych. Zbocze narastające na drugim powoduje wykonanie kroku w kierunku ustawionym przez stan drugiego wyprowadzenia. Sterowniki pozwalają na sterowanie mikrokrowe w trybie 1/2, 1/4, 1/8 i 1/16 kroku. W projekcie silniki zostały skonfigurowane na pracę w trybie

1/8 kroku, pojedynczy krok w tym trybie obraca wał o 0.225 stopnia. Ustawienie to dostarcza wystarczającą precyzję sterowania. Użycie pracy w trybie 1/16 kroku zmniejszyłoby moment obrotowy silników a dalsze zwiększanie precyzji sterowania nie jest konieczne.

3.5. Zasilanie

Robot jest zasilany przy pomocy 4 ogniw litowo-jonowych połączonych szeregowo. Są to niezależne ogniwa 18650 o napięciu znamionowym 3.7V i pojemności około 2200mAh. Napięcie całego pakietu waha się od 16.8V przy pełnym naładowaniu do 12V w stanie rozładowania akumulatorów. Elektronika zasilana jest przy pomocy regulowanej przetwornicy impulsowej dostarczającej napięcie 5V. Sterownik silników zasilany jest bezpośrednio z akumulatorów w związku z czym ich zachowanie zależne jest od stopnia rozładowania.

4. Modelowanie i identyfikacja

Wstępne obliczenia matematyczne oraz symulacje są ważnym elementem procesu projektowego umożliwiającym wczesną weryfikację przyjętych założeń. Model matematyczny oparty o wstępnie oszacowane parametry umożliwia przetestowanie projektu przed zużyciem czasu i środków na potencjalnie błędne rozwiązanie.

4.1. Model nieliniowy

Model matematyczny robota bazuje na nieliniowych równaniach różniczkowych uzyskanych przy pomocy równań Lagrange'a. Wyprowadzenie zostało w pełni oparte o [X] oraz zweryfikowane przy pomocy symulacji opisanej w punkcie X.X.

4.1.1. Opis matematyczny obiektu

Przyjęte założenia: oba koła poruszają się z tą samą prędkością obrotową bez poślizgu między powierzchnią koła a podłożem.

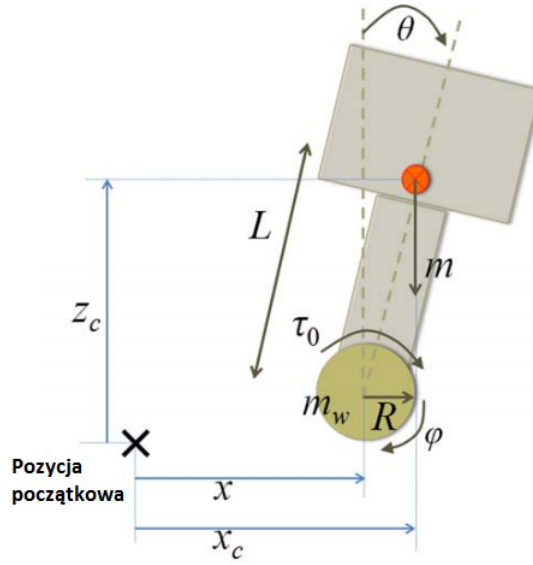
Zmienne opisujące system:

- x - pozioma odległość środka koła od przyjętego układu współrzędnych
- x_c - pozioma odległość środka masy od przyjętego układu współrzędnych
- z_c - pionowa odległość środka masy od przyjętego układu współrzędnych
- ϕ - położenie katowe koła
- θ - położenie katowe robota względem pozycji pionowej

Pozostałe parametry to:

- m - Masa korpusu robota
- m_w - Masa kół
- R - Promień koła
- L - Odległość pomiędzy środkiem masy a osią koła
- τ_0 - zastosowany moment obrotowy
- I - moment bezwładności korpusu robota
- I_w - moment bezwładności kół

Wizualizacja przyjętych symboli została przedstawiona na rysunku 4.1



Rys. 4.1. Przyjęty opis stanu obiektu

4.1.1.1. Wyprowadzenie równań modelu

Zmienne $x, x_c, \dot{x}, \dot{x}_c, \dot{z}_c$ można zapisać jako:

$$x = R\phi \quad (4.1)$$

$$x_c = R\phi + L\sin\theta \quad (4.2)$$

$$z_c = R + L\cos\theta \quad (4.3)$$

$$\dot{x} = R\dot{\phi} \quad (4.4)$$

$$\dot{x}_c = R\dot{\phi} + L\dot{\theta}\cos\theta \quad (4.5)$$

$$\dot{z}_c = -L\dot{\theta}\sin\theta \quad (4.6)$$

Na podstawie równań 4.1:4.6 możemy zapisać energię potencjalną E_p i kinetyczną E_k w postaci 4.7:4.9

$$E_p = mg(R + L\cos\theta) - mg(R + L) = mgl(\cos\theta - 1) \quad (4.7)$$

$$E_k = 1/2m_w\dot{x}^2 + 1/2I_w\dot{\phi}^2 + 1/2m\dot{x}_c^2 + 1/2m\dot{z}_c^2 + 1/2mI\dot{\theta}^2 \quad (4.8)$$

$$= 1/2(I_w + m_wR^2 + mR^2)\dot{\phi}^2 + mRL\cos\theta\dot{\phi}\dot{\theta} + 1/2 * (I + mL^2)\dot{\theta}^2 \quad (4.9)$$

Lagranżian \mathcal{L} może być zapisany jako różnica pomiędzy energią kinetyczną a potencjalną

$$\mathcal{L} = E_k - E_p \quad (4.10)$$

$$= 1/2(I_w + m_wR^2 + mR^2)\dot{\phi}^2 + mRL\cos\theta\dot{\phi}\dot{\theta} + 1/2(I + mL^2)\dot{\theta}^2 + mgL(\cos\theta - 1) \quad (4.11)$$

W związku z czym równania dynamiczne dla składowych θ i ϕ mogą zostać wyznaczone w 4.12:4.17

$$\frac{\partial \mathcal{L}}{\partial \dot{\phi}} = (I_w + m_w R^2 + m R^2) \dot{\phi} + m R L \dot{\theta} \cos \theta \quad (4.12)$$

$$\frac{\partial \mathcal{L}}{\partial \phi} = 0 \quad (4.13)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}} \right) - \frac{\partial \mathcal{L}}{\partial \phi} = (I_w + m_w R^2 + m R^2) \ddot{\phi} + m R L \cos \theta \ddot{\theta} - m R L \sin \theta \dot{\theta}^2 = \mu \quad (4.14)$$

$$\frac{\partial \mathcal{L}}{\partial \dot{\theta}} = m R L \cos \theta \dot{\phi} + (I + m L^2) \dot{\theta} \quad (4.15)$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = -m R L \sin \theta \dot{\phi} \dot{\theta} + m g L \sin \theta \quad (4.16)$$

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = (I + m L^2) \ddot{\theta} + m R L \cos \theta \ddot{\phi} - m R L \sin \theta = \chi \quad (4.17)$$

Gdzie μ i χ to uogólnione siły(momenty obrotowe) dla odpowiednich składowych. Uzyskane w ten sposób równania różniczkowe można zapisać w postaci macierzowej

$$\begin{bmatrix} I_w + (m_w + m) R^2 & m R L \cos \theta \\ m R L \cos \theta & I + m L^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -m R L \sin \theta \dot{\theta} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -m g L \sin \theta \end{bmatrix} = \begin{bmatrix} \mu \\ \chi \end{bmatrix} \quad (4.18)$$

Kolejny krok skupia się na wyrażeniu uogólnionych momentów obrotowych za pomocą znanych parametrów. Momenty obrotowy działający na robota możemy zapisać jako sumę momentów wytworzonych przez silniki oraz rozproszonych poprzez tarcie.

τ_0 - łączny moment obrotowy dostarczony przez silniki

τ_a -moment rozproszony poprzez tarcie w łożyskach

τ_f -moment rozproszony poprzez tarcie kół o podłoże

β_a -współczynnik tarcia w łożyskach

β_f -współczynnik tarcia kół o podłoże

Po wprowadzeniu powyższych parametrów możemy zapisać uogólnione momenty obrotowe w postaci 4.19:4.20

$$\mu = \tau_0 - \tau_a - \tau_f = \tau_0 - \beta_f (\dot{\phi} - \dot{\theta}) - \beta_a \dot{\phi} \quad (4.19)$$

$$\chi = -\tau_0 + \tau_a = -\tau_0 + \beta_f (\dot{\phi} - \dot{\theta}) \quad (4.20)$$

4.2. Model liniowy

Kontrola predykcyjna opiera się na modelu liniowym wyrażonym równaniami stanu. Wprowadzony w punkcie 4.0.1 model robota jest nieliniowy, w związku z czym, wymaga on linearyzacji w punkcie równowagi. Jako punkt linearyzacji modelu wybrany został górny, niestabilny punkt równowagi.

Do w celu umożliwienia linearyzacji konieczne były następujące uproszczenia:

- Niewielki zakres ruchów względem pozycji pionowej
- $\cos\theta \approx 1$
- $\sin\theta \approx 0$

Równania modelu 4.19 po wprowadzeniu tej modyfikacji przedstawione zostały jako równanie 4.21.

$$\begin{bmatrix} I_w + (m_w + m)R^2 & mRL \\ mR & I + mL^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} \beta_a + \beta_f & -\beta_f \\ -\beta_f & \beta_f \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -mgL \end{bmatrix} \theta = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tau_0 \quad (4.21)$$

Równanie 4.21 można przedstawić w zwężym stylu w postaci 4.22

$$E \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} + F \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \end{bmatrix} + G\theta = H\tau_0 \quad (4.22)$$

Na podstawie równania 4.22 możemy zapisać model systemu w postaci równań stanu. Jako zmienne stanu wykorzystane zostaną $\phi, \theta, \dot{\phi}, \dot{\theta}$. Wejściem systemu jest moment obrotowy τ_0 . Jako wyjścia zostały wybrane przemieszczenie robota w osi x oraz położenie katowe korpusu θ . W związku z tymi założeniami równania stanu systemu przyjmują postać 4.23 :4.24

$$\frac{d}{dt} \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} = TO_D U Z E \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \tau_0 \quad (4.23)$$

$$y = \begin{bmatrix} R & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \quad (4.24)$$

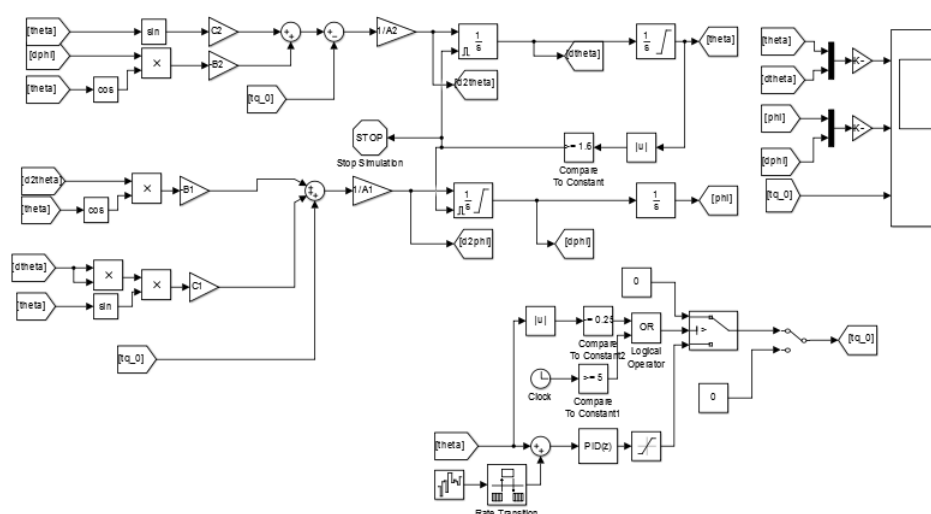
Przyjmując standardową nomenklaturę teorii sterowania finalna postać równań stanu została przedstawiona za pomocą równań: 4.25 :4.26

$$\frac{d}{dt}x = Ax + B\tau_0 \quad (4.25)$$

$$y = Cx \quad (4.26)$$

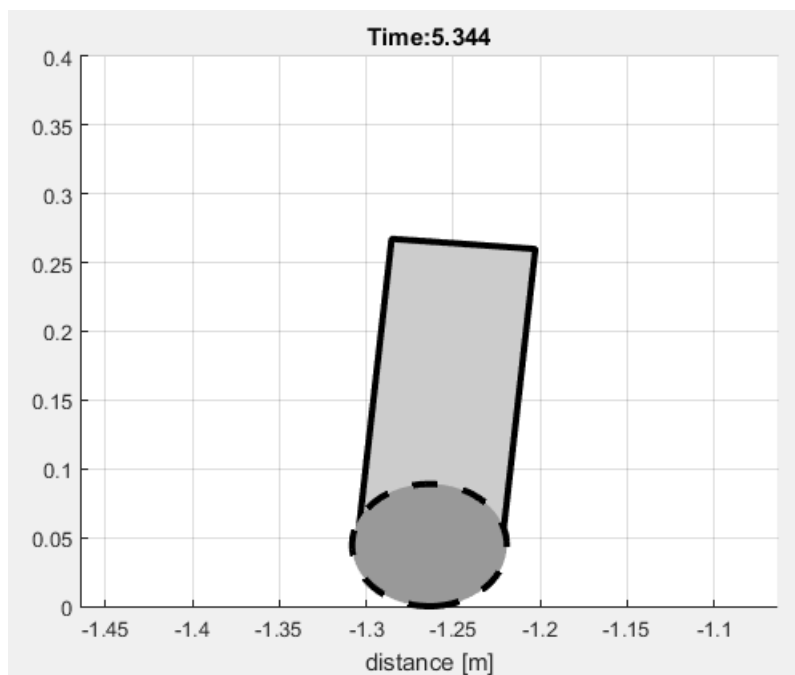
4.3. Symulacja oraz wizualizacja wyników

Symulacja zachowania robota w oparciu o wprowadzony w punkcie 4.0.1.2 model została zaimplementowana w pakiecie Matlab Simulink. Implementacja została wykorzystana do weryfikacji poprawności modelu nieliniowego, jako test planowanych rozwiązań sprzętowych oraz regulatora. Zrzut ekranu zawierający główną część symulacji został przedstawiony na rysunku 4.2



Rys. 4.2. Symulacja modelu nieliniowego

Przed każdym uruchomieniem można wybrać wartości początkowe podstawowych parametrów takich jak prędkość i pozycja kątowa koła oraz korpusu oraz nastawy regulatora i czas symulacji. Parametry modelu ustawiane są w niezależnym skrypcie uruchamianym przed każdą symulacją. Po symulacji wykonywany jest skrypt wyrysowywujący przebiegi kluczowych zmiennych oraz prosta animacja wizualizująca przebieg symulacji. Pozwala to na szybką ocenę czy zachowanie modelu jest bliskie rzeczywistości. Przykładowy wynik symulacji został zaprezentowany na rysunku 4.3



Rys. 4.3. Symulacja modelu nieliniowego

5. Sterowanie predykcyjne

Sterowanie predykcyjne (ang. MPC – model predictive control) to rodzaj sterowania w którym regulator dostosowuje swoje wyjście, nie tylko do aktualnego stanu układu, lecz także do przewidywanych przyszłych trajektorii. Przewidywany stan jest wyznaczany na podstawie aktualnych wejść oraz równań stanu układu. Wyznaczanie sterowania polega na cyklicznym wyznaczaniu optymalnej trajektorii przy warunkach początkowych równych aktualnemu stanowi obiektu. Pierwszy odcinek sekwencji sterowań jest podawany na wejście obiektu, następnie cała procedura zostaje powtórzona dla aktualnego stanu obiektu w celu wyznaczenia nowej trajektorii. [ta książka od mpc]. Przykładowy schemat obrazujący pracę regulatora predykcyjnego został przedstawiony na rysunku X.

-Wyjaśnienie symboli-

5.1. Sterowanie predykcyjne na podstawie równań stanu

Dla uproszczenia podstawy matematyczne kontroli predykcyjnej zaprezentowane zostaną na prostym modelu z jednym wejściem i jednym wyjściem przedstawionym równaniami 5.1:5.2

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \quad (5.1)$$

$$y(k) = C_m x_m(k) \quad (5.2)$$

Gdzie u jest wejściem systemu, y to wyjście procesu a x_m jest wektorem stanu długości n_1 . Na potrzeby kontroli predykcyjnej konieczne będzie zapisanie systemu w postaci różnicowej 5.3

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \quad (5.3)$$

Kolejnym etapem jest powiązanie $\Delta x_m(k)$ z wyjściem $y(k)$. Można to zrobić za pomocą nowego wektora stanu w postaci 5.4

$$x(k) = [\Delta x_m(k)^T + y(k)]^T \quad (5.4)$$

Wprowadzone przekształcenia pozwalają zapisać model 5.1 w postaci rozszerzonej 5.5:5.6 będącej podstawą kontroli predykcyjnej.

$$x(k+1) = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) \quad (5.5)$$

$$y(k) = \begin{bmatrix} o_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (5.6)$$

Uproszczony zapis postaci rozszerzonej został przedstawiony przy pomocy równań 5.7:5.8

$$x(k+1) = Ax(k) + B\Delta u(k) \quad (5.7)$$

$$y(k) = C \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (5.8)$$

5.2. Kontrola predykcyjna z pojedynczym oknem optymalizacji

Po wprowadzeniu modelu rozszerzonego kolejnym etapem jest wyznaczenie przewidywanego wyjścia obiektu na podstawie przyszłych wartości sterowania. Oszacowanie to jest nazywane oknem optymalizacji. Parametrem definiującym okno jest jego długość N_p . Przyjmując, że w dyskretnej chwili k_i wektor $x(k_i)$ jest możliwy do pomiaru, stan $x(k_i)$ dostarcza informację o aktualnym stanie modelu. Trajektora przyszłych sterowań może zostać zapisana jako 5.9

$$\Delta(k_i), \Delta(k_i+1), \dots, \Delta(k_i+N_c-1), \quad (5.9)$$

gdzie N_c to horyzont kontroli oznaczający ilość wyznaczanych przyszłych próbek sterowania. Na podstawie informacji $x(k_i)$ przyszłe zmienne stanu wyznaczone są dla N_p próbek gdzie parametr N_p oznacza horyzont predykcji. Horyzont kontroli musi być mniejszy od równy od horyzontu predykcji. Przyszłe zmienne stanu można zapisać w postaci 5.10

$$x(k_i+1|k_i), x(k_i+2|k_i), \dots, x(k_i+m|k_i), \dots, x(k_i+N_p|k_i), \quad (5.10)$$

Gdzie $x(k_i+m|k_i)$ jest przewidywanym stanem zmiennej w chwili k_i+m na podstawie stanu $x(k_i)$. Na podstawie równań stanu wprowadzonych w punkcie przyszłe zmienne stanu są obliczane sekwencyjnie przy wykorzystaniu parametrów kontrolnych 5.11

$$\begin{aligned} x(k_i+1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\ x(k_i+2|k_i) &= Ax(k_i+1) + B\Delta u(k_i+1) \\ &= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i+1) \\ x(k_i+N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i+1) \\ &\quad + \dots + A^{N_p-N_c}B\Delta u(k_i+N_c-1) \end{aligned} \quad (5.11)$$

Na podstawie przyszłych zmiennych stanu wyznaczone są przyszłe wyjścia 5.12

$$\begin{aligned}
y(k_i + 1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i + 2|k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
y(k_i + N_p|k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i + 1) \\
&\quad + \dots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1)
\end{aligned} \tag{5.12}$$

Kolejnym krokiem jest wprowadzenie wektorów Y i ΔU 5.13

$$\begin{aligned}
Y &= \begin{bmatrix} y(k_i + 1|k_i) & y(k_i + 2|k_i) & y(k_i + 3|k_i) & \dots & y(k_i + N_p|k_i) \end{bmatrix}^T \\
\Delta U &= \begin{bmatrix} \Delta u(k_i) & \Delta u(k_i + 1) & \Delta u(k_i + 2) & \dots & \Delta u(k_i + N_c - 1) \end{bmatrix}^T
\end{aligned} \tag{5.13}$$

Dzięki wektorom Y i ΔU możliwe jest zapisanie 5.11 i 5.12 w postaci 5.14

$$Y = Fx(k_i) + \Phi\Delta U \tag{5.14}$$

Gdzie

$$\begin{aligned}
F &= \begin{bmatrix} CA & CA^2 & CA^3 & \dots & CA^{N_p} \end{bmatrix}^T \\
\Phi &= \begin{bmatrix} CB & 0 & 0 \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}
\end{aligned} \tag{5.15}$$

5.3. Optymalizacja

Dla sygnału $r(k_i)$ reprezentującego wartość zadaną w chwili k_i , w zakresie horyzontu predykcji, celem regulatora predykcyjnego jest sprowadzić przewidywane wyjście, tak blisko jak to możliwe do wartości zadanej. Wartość zadana pozostaje stała w pojedynczym oknie optymalizacji. Wektor optymalnych sterowań ΔU minimalizuje funkcję błędu 5.17 pomiędzy wartością zadaną a przewidywanym wyjściem układu.

$$R_s^T = \begin{bmatrix} 1_2 & 1_2 & \dots & 1_{N_p} \end{bmatrix} r(k_i) \tag{5.16}$$

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \tag{5.17}$$

Pierwszą część równania 5.17 jest odpowiadająca za minimalizację błędu pomiędzy wyjściem obiektu a wartością zadaną. Druga część odpowiada kosztom odpowiadającym sterowaniu. Wyeliminowanie tego elementu sprawi, że duże wartości ΔU nie będą pogarszały wartości funkcji celu.

Aby uzyskać wektor optymalnych sterowań należy znaleźć minimum J przy użyciu 5.14. Równanie 5.17 można wtedy zapisać w postaci 5.18

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (5.18)$$

Warunkiem koniecznym istnienia minimum J jest wyrażony równaniem 5.19

$$\frac{dJ}{d\Delta U} = 0 \quad (5.19)$$

Przy wykorzystaniu równania 5.19 można wyznaczyć wektor optymalnych sterowań w postaci 5.20

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \quad (5.20)$$

Przy założeniu, że $(\Phi^T \Phi + \bar{R})^{-1}$ istnieje. Macierz $(\Phi^T \Phi + \bar{R})^{-1}$ zgodnie z nomenklaturą nazywamy Hessianem.

Związek wektora sterowań optymalnych z wektorem stanu oraz wartością zadaną przedstawia równanie 5.21

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)) \quad (5.21)$$

5.4. Kontrola Predykcyjna z ograniczeniami

Przedstawiona w sekcji 5.3 metoda wyznaczania wektora sterowań optymalnych ΔU nie umożliwia implementacji ograniczeń którym podlegają zmienne stanu oraz sterowanie. W rzeczywistych systemach sterowania wartości fizyczne opisujące obiekt mają ograniczony zakres. W omawianym w rozdziale 4 modelu robota większość ograniczeń wynika z ograniczonej mocy zastosowanych silników. Ich nota katalogowa omówiona w rozdziale 2 pozwala oszacować maksymalny moment obrotowy, przyspieszenie oraz prędkość kątową. Sterowanie wyznaczające wartości poza zakresem nie będzie mogło być zrealizowane. W celu zachowania zgodności systemu sterowania z rzeczywistym obiektem wprowadzone zostaną ograniczenia.

Ograniczenia w postaci 5.22 mogą określać maksymalny przyrost zmiennej podczas jednej iteracji lub całkowitą wartość zmiennej.

$$\begin{aligned} \Delta x^{min} &\leq \Delta x(k) \leq \Delta x^{max} \\ x^{min} &\leq x(k) \leq x^{max} \end{aligned} \quad (5.22)$$

Następnym krokiem po sformułowaniu równań w postaci 5.22 jest wprowadzenie tych zależności do procesu wyznaczania optymalnego sterowania. Ograniczenia przyrostu zmiennej są prostsze do wprowadzenia ze względu na zastosowaną postać różnicową równań stanu. Ograniczenia przyrostowe dla wektora X można przedstawić jako zależność 5.23 która może zostać zapisane pod postacią dwóch nierówności 5.24

$$\Delta X^{min} \leq \Delta X \leq \Delta X^{max} \quad (5.23)$$

$$-\Delta X \leq \Delta X^{min} \quad (5.24)$$

$$\Delta X \leq \Delta X^{max}$$

Nierówności 5.24 można zapisać w postaci macierzowej 5.25

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta X \leq \begin{bmatrix} -\Delta X^{min} \\ \Delta X^{max} \end{bmatrix} \quad (5.25)$$

Ograniczenia rzeczywistej wartości zmiennej wymagają drobnej modyfikacji. W celu ograniczenia zakresu wartości przyjmowanych przez zmienną w systemie przyrostowym, ograniczona musi zostać suma wszystkich elementów rozważanego wektora ΔX oraz wartości początkowej $x(k_0)$. Zamieniając macierze jednostkowe macierzą Toeplitza T przedstawionej równaniem 5.26

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (5.26)$$

Zapis ten w formie macierzowej przyjmuje postać 5.27

$$\begin{bmatrix} -T \\ T \end{bmatrix} \Delta X \leq \begin{bmatrix} -X^{min} + x(k_0) \\ X^{max} - x(k_0) \end{bmatrix} \quad (5.27)$$

Znając wartości max , Δmax , min oraz Δmin dla każdej zmiennej można sformułować dla niej ograniczenia w postaci macierzowej 5.25 oraz 5.27. Wszystkie ograniczenia mają postać nierówności liniowych. Hessian $(\Phi^T \Phi + \bar{R})^{-1}$ jest pozytywnie określony a funkcja kosztu J jest kwadratowa. Spełniając powyższe założenia problem wyznaczenia optymalnego sterowania ΔU staje się standardowym zadaniem optymalizacji kwadratowej z ograniczeniami

6. Optymalizacja sterowania

W rozdziale 5 sformułowane zostało zadanie optymalizacji kwadratowej z ograniczeniami, którego rozwiązanie pozwoli wyznaczyć finalne sterowanie robota. Rozdział bieżący przedstawia zastosowane podejście do rozwiązania przedstawionego zadania zarówno w symulacji zaimplementowanej w pakiecie Matlab Simulink oraz w rzeczywistym sterowniku robota.