

Numer ćwiczenia	-	<u>Tytuł ćwiczenia:</u> Optymalizacja sterowania 2-osiowym robotem SCARA
Data wykonania ćwiczenia:	03-05.2022	<u>Imię, nazwisko i numer albumu:</u>
Data oddania sprawozdania:	12.06.2022	1) Mateusz Głuch 305744 2) Kamil Głoski 305742 3) Olgierd Duda 305736
Nr grupy lab./Kierunek:	1a/ISS/AiR	

Spis treści

1. Cel ćwiczenia.....	2
2. Modelowanie systemu	2
3. Kinematyka prosta i odwrotna	3
3.1. Równania kinematyki prostej.....	3
3.2. Równania kinematyki odwrotnej	3
4. Sterowanie czasooptymalne	4
4.1. Funkcja celu	4
4.2. Hamiltonian i równania sprzężone	4
4.3. Sterowanie bang-bang i jego konsekwencje	4
4.4. Metodyka rozwiązywania numerycznego zadania optymalizacji.....	5
4.5. Wyniki i analiza rezultatów	6
5. Sterowanie z optymalizacją kwadratowej funkcji celu	12
5.1. Funkcja celu	12
5.2. Hamiltonian i równania sprzężone – modyfikacja.....	12
5.3. Metodyka rozwiązania równań różniczkowych	12
5.4. Wyniki dla różnych trajektorii dla zadania podążania za trajektorią	13
6. Sterowanie predykcyjne – Model Predictive Control	19
6.1. Implementacja MPC ze sterowaniem optymalnym	19
6.2. Analiza jakościowa regulacji MPC – horyzont i krok predykcji.....	19
7. Wnioski.....	26

1. Cel ćwiczenia

Celem ćwiczenia jest zaprojektowanie oprogramowania pozwalającego na wyznaczenie sterowania optymalnego (tj. minimalizującego zadaną funkcję celu) dla modelu matematycznego 2-osiowego robota SCARA. Regulator ten został rozpatrzony w wariancie czasooptymalnym oraz w wariancie z kwadratową funkcją celu. Następnie został zaprojektowany regulator Model Predictive Control, pozwalający zarówno na skrócenie czasu obliczeń poprzez zastosowanie skończonego horyzontu, ale także na zastosowanie sprzężenia zwrotnego od stanu obiektu.

2. Modelowanie systemu

Robot 2-osiowy SCARA został zamodelowany jako obiekt nieliniowy z dwoma sterowaniami (ze względu na ruchliwość mechanizmu robota równą 2). Przy modelowaniu uwzględniono zarówno niezerową masę członów robota, jak i nieliniowy model tarcia.

Równanie ruchu manipulatora ma postać:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{12} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} - \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

Oznaczenia:

$$\begin{aligned} M_{11} &= I_1 + I_2 + m_2 d_1^2 + 2B \\ M_{12} &= B = m_2 d_1 (x_{c2} \cos x_2 - y_{c2} \sin x_2) \\ M_{22} &= I_2 \\ B_2 &= -m_2 d_1 (x_{c2} \sin x_2 + y_{c2} \cos x_2) \\ V_1 &= B_2 (2\dot{x}_1 \dot{x}_2 + \dot{x}_2^2) \\ V_2 &= -B_2 \dot{x}_1^2 \\ T_1 &= s_1 \operatorname{sgn}(\dot{x}_1) + f_1 \dot{x}_1 \\ T_2 &= s_2 \operatorname{sgn}(\dot{x}_2) + f_2 \dot{x}_2 \end{aligned}$$

Macierz odwrotna do macierzy momentów bezwładności została oznaczona symbolem \bar{M} i jest równa:

$$\begin{bmatrix} \bar{M}_{11} & \bar{M}_{12} \\ \bar{M}_{12} & \bar{M}_{22} \end{bmatrix} = \frac{1}{M_{11} * M_{22} - M_{12}^2} \begin{bmatrix} M_{22} & -M_{12} \\ -M_{12} & M_{11} \end{bmatrix}$$

W modelu manipulatora został zastosowany przybliżony, ciągły model tarcia:

$$T_1 = s_1 \tanh(k\dot{x}_1) + f_1 \dot{x}_1$$

$$T_2 = s_2 \tanh(k\dot{x}_2) + f_2 \dot{x}_2$$

Poprzez oznaczenie pozostałych zmiennych stanu jako $x_3 = \dot{x}_1$, $x_4 = \dot{x}_2$, można zapisać równania stanu manipulatora w postaci jednorodnej względem sterowania:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2$$

Gdzie:

$$f(x) = \begin{bmatrix} x_3 \\ x_4 \\ -\bar{M}_{11}(V_1 + T_1) - \bar{M}_{12}(V_2 + T_2) \\ -\bar{M}_{12}(V_1 + T_1) - \bar{M}_{22}(V_2 + T_2) \end{bmatrix}, g_1(x) = \begin{bmatrix} 0 \\ 0 \\ \bar{M}_{11} \\ \bar{M}_{12} \end{bmatrix}, g_2(x) = \begin{bmatrix} 0 \\ 0 \\ \bar{M}_{12} \\ \bar{M}_{22} \end{bmatrix}$$

Parametry manipulatora ustawiono zgodnie z poniższą tabelą:

Numer członu	I [kg*m ²]	m[kg]	d[m]	x _{ci} [m]	y _{ci} [m]	s _i [Nm]	f _i [W ⁻¹]	u _{min} [Nm]	u _{max} [Nm]
1	1.00	10.0	1.00	0.50	0.00	0.10	0.01	-1.00	1.00
2	1.00	5.0	0.70	0.35	0.00	0.10	0.01	-1.00	1.00

3. Kinematyka prosta i odwrotna

3.1. Równania kinematyki prostej

Robot SCARA porusza się po przestrzeni roboczej odnosząc się do położeń swoich przegubów, wyrażonych jako odpowiednie kąty obrotu (Rys. 4.1). Aby możliwe było wyznaczenie położenia robota w przestrzeni kartezjańskiej, konieczne jest rozwiązanie tzw. zadania prostego kinematyki robota, czyli przekształcenia współrzędnych kątowych na kartezjańskie. Dla 2-osiowego płaskiego robota o konfiguracji SCARA równania kinematyki prostej przedstawiają się następująco (poprzez macierze rotacji):

Oznaczenia:

φ_1 - położenie kątowe przegubu 1

φ_2 - położenie kątowe przegubu 2

Macierz rotacji przegubu 1:

$$R_1 = \begin{bmatrix} \cos(\varphi_1) & -\sin(\varphi_1) \\ \sin(\varphi_1) & \cos(\varphi_1) \end{bmatrix}$$

Macierz rotacji przegubu 2:

$$R_2 = \begin{bmatrix} \cos(\varphi_2) & -\sin(\varphi_2) \\ \sin(\varphi_2) & \cos(\varphi_2) \end{bmatrix}$$

Równania kinematyki prostej:

$$\begin{bmatrix} x \\ y \end{bmatrix} = R_1 * \begin{bmatrix} d_1 \\ 0 \end{bmatrix} + R_1 * R_2 * \begin{bmatrix} d_2 \\ 0 \end{bmatrix}$$

3.2. Równania kinematyki odwrotnej

Aby możliwe było przekształcenie zadanej trajektorii w postaci parametrycznej $P(t) = (x(t), y(t))$, konieczne jest rozwiązanie tzw. zadania odwrotnego kinematyki, tj. przekształcenie współrzędnych kartezjańskich na położenia kątowe odpowiednich przegubów robota. Dodatkowym problemem jest fakt, że o ile równania kinematyki prostej robota mają zawsze dokładnie jedno rozwiązanie (dla każdego położenia kątowego robota istnieje dokładnie jeden punkt i orientacja, w której robot się znajduje), o tyle równania kinematyki odwrotnej na ogół mają kilka rozwiązań – konfiguracji robota. W tym przypadku mamy do czynienia z dwoma konfiguracjami – „above” i „below” (zob. rys. 4.2). Równania kinematyki odwrotnej przedstawione są poniżej:

$$\varphi_1 = \alpha - \arctg(y/x) \quad \varphi_2 = \pm \arccos(b)$$

Gdzie zmienne α, b, r wyrażone są za pomocą:

$$\alpha = \arctg\left(\frac{\pm d_2 * \sqrt{1 - b^2}}{d_1 + d_2 * b}\right)$$

$$b = \frac{r^2 - d_1^2 - d_2^2}{2d_1d_2}$$

$$r = \sqrt{x^2 + y^2}$$

4. Sterowanie czasooptymalne

4.1. Funkcja celu

Funkcja celu w algorytmie czasooptymalnym została dobrana w ten sposób, aby gwarantować zarówno optymalizację pod kątem czasu sterowania, ale także pod kątem osiągnięcia założonego celu; tj. stan w chwili końcowej symulacji musi odpowiadać stanowi referencyjnemu. Funkcja celu przedstawia się w następujący sposób:

$$J(x, T) = T + (x(T) - x_f) * W * (x(T) - x_f)$$

Gdzie:

W – macierz wag związana ze stanem układu ($W \geq 0$)

T – czas symulacji

x(T) – stan obiektu w chwili końcowej

x_f – założony stan obiektu w chwili końcowej

4.2. Hamiltonian i równania sprzężone

Hamiltonian dla tego przypadku jest wyrażony wzorem:

$$H(\psi, x, u_1, u_2) = \psi^T (f(x) + g_1(x)u_1 + g_2(x)u_2)$$

Gdzie funkcje f, g₁ i g₂ są wyrażone jak w punkcie 3.

Równania sprzężone mają postać:

$$\dot{\psi} = -\frac{\partial H(\psi, x, u_1, u_2)}{\partial x} = -A\psi \quad \psi(T) = \frac{\partial J(x, T)}{\partial x} = W(x(T) - x_f)$$

Gdzie pochodna hamiltonianu względem stanu obiektu przybiera postać:

$$\frac{\partial H(\psi, x, u_1, u_2)}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & A_{23} & A_{24} \\ 1 & 0 & A_{33} & A_{34} \\ 0 & 1 & A_{43} & A_{44} \end{bmatrix}$$

Gdzie poszczególne współczynniki macierzy:

$$A_{23} = \partial_{x_2} M_{11} * (V_1 + T_1 - u_1) + \bar{M}_{11} \partial_{x_2} V_1 + \bar{M}_{12} \partial_{x_2} V_2 + \partial_{x_2} M_{12} * (V_2 + T_2 - u_2)$$

$$A_{24} = \partial_{x_2} M_{12} * (V_1 + T_1 - u_1) + \bar{M}_{12} \partial_{x_2} V_1 + \bar{M}_{22} \partial_{x_2} V_2 + \partial_{x_2} M_{22} * (V_2 + T_2 - u_2)$$

$$A_{33} = \bar{M}_{11} (\partial_{x_3} V_1 + \partial_{x_3} T_1) + \bar{M}_{12} \partial_{x_3} V_2$$

$$A_{34} = \bar{M}_{12} (\partial_{x_3} V_1 + \partial_{x_3} T_1) + \bar{M}_{22} \partial_{x_3} V_2$$

$$A_{43} = \bar{M}_{11} \partial_{x_4} V_1 + \bar{M}_{12} \partial_{x_4} T_2$$

$$A_{44} = \bar{M}_{12} \partial_{x_4} V_1 + \bar{M}_{22} \partial_{x_4} T_2$$

4.3. Sterowanie bang-bang i jego konsekwencje

W związku z tym, że celem sterowania jest osiągnięcie minimalnego czasu sterowania układu, zasadnym jest założenie, że sterowanie optymalne będzie przybierało wartości z dyskretnego zbioru (zob. tw. Feldbauma):

$$u \in \{u_{max}, u_{min}\} = \{-1, 1\}$$

Oznacza to, że mimo osiągnięcia przez hamiltonian swojego maksimum (zgodnie z zasadą maksimum Pontriagina), pochodna hamiltonianu względem sterowań nie będzie równa 0 – przyjmie ona następujące wartości:

$$\phi_1(t) = \frac{\partial H(\psi, x, u_1, u_2)}{\partial u_1} = \begin{cases} \phi_1 > 0 \Leftarrow u_1 = u_{max} \\ \phi_1 < 0 \Leftarrow u_1 = u_{min} \end{cases}$$

$$\phi_2(t) = \frac{\partial H(\psi, x, u_1, u_2)}{\partial u_2} = \begin{cases} \phi_2 > 0 \Leftarrow u_2 = u_{max} \\ \phi_2 < 0 \Leftarrow u_2 = u_{min} \end{cases}$$

Ze względu na powyższy fakt, zasadnym jest nazwanie funkcji $\phi(t) = [\phi_1(t); \phi_2(t)]$ funkcją przełączającą, ponieważ dla sterowania optymalnego zachodzą poniższe zależności (rzecz jasna, **jedynie dla pewnego szczególnego przypadku, gdy** $|u_i(t)| \leq 1$, natomiast jest to wygodny sposób na przedstawienie funkcji przełączającej – poprzez funkcję signum):

$$u_1(t) = \text{sgn}(\phi_1(t)); u_2(t) = \text{sgn}(\phi_2(t))$$

Jawnie, dla tego problemu funkcja przełączająca wyrazi się wzorami:

$$\phi_1(t) = \psi^T g_1(x) \quad \phi_2(t) = \psi^T g_2(x)$$

4.4. Metodyka rozwiązywania numerycznego zadania optymalizacji

Zadanie optymalizacji jest rozwiązywane numerycznie – ze względu na to, że trajektorię stanu złożonego, nieliniowego modelu można w rozsądny sposób otrzymać jedynie jako rezultat zastosowania metod numerycznych.

Do rozwiązania równań stanu, jak i równań różniczkowych zastosowano metodę Rungego-Kutty 4-go rzędu, opisaną w następujący sposób: jeśli równanie różniczkowe ma postać $\dot{x} = f(x, t)$ oraz równanie różniczkowe rozwiązywane jest z krokiem h , to kolejne wartości zmiennych stanu równania różniczkowego wyrażają się poprzez równania:

$$x_{n+1} = x_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \quad t_{n+1} = t_n + h$$

Gdzie k_1, k_2, k_3, k_4 są wyrażone poprzez:

$$k_1 = f(x_n, t_n) \quad k_2 = f\left(t_n + \frac{h}{2}, x_n + h\frac{k_1}{2}\right) \quad k_3 = f\left(t_n + \frac{h}{2}, x_n + h\frac{k_2}{2}\right) \quad k_4 = f(t_n + h, x_n + hk_3)$$

Do przeprowadzenia odpowiedniego algorytmu optymalizacyjnego konieczne jest także zdefiniowanie odpowiednich gradientów ze względu na czasy przełączeń τ_i . Gradient ten wyraża się poprzez:

$$\frac{\partial J(x, T)}{\partial \tau_{1i}} = (u_{1(i+1)} - u_{1i}) * \phi_1(\tau_i) \quad \frac{\partial J(x, T)}{\partial \tau_{2i}} = (u_{2(i+1)} - u_{2i}) * \phi_2(\tau_i)$$

Algorytm obliczania kolejnych sterowań odbywa się wg następującego schematu:

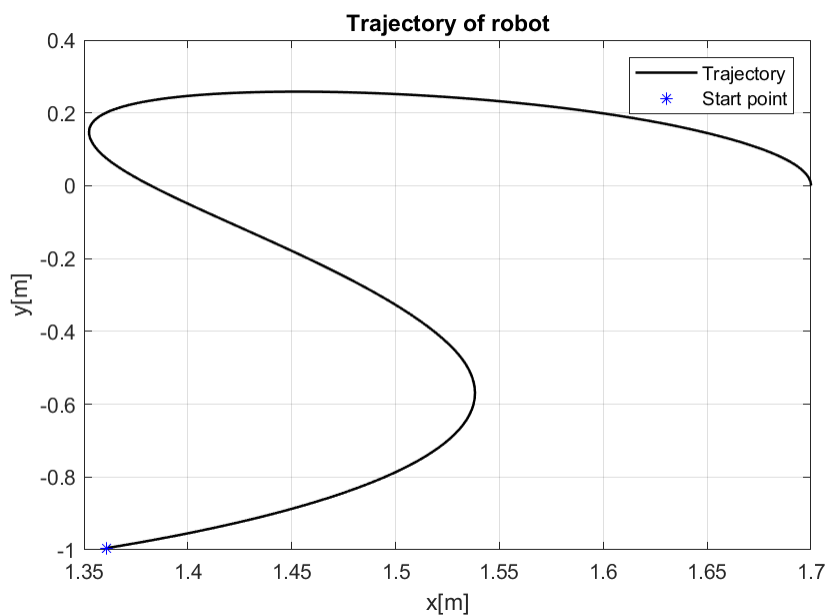
- rozwiązanie równań stanu dla obecnego sterowania (u_n),
- obliczenie końcowych wartości zmiennych sprzężonych ($\psi(T)$),
- rozwiązanie równań sprzężonych (uwaga: równania sprzężone rozwiązuje się z czasem do tyłu),
- obliczenie gradientów hamiltonianu
- wyznaczenie nowych czasów przełączeń – wykonywane przez funkcję *fmincon* w Matlabie, bądź wyznaczane za pomocą metody gradientu prostego:

$$\tau_{1j}(i+1) = \tau_{1j}(i) - \alpha \frac{\partial J(x, T)}{\partial \tau_{1i}} \quad \tau_{2j}(i+1) = \tau_{2j}(i) - \alpha \frac{\partial J(x, T)}{\partial \tau_{2i}}$$

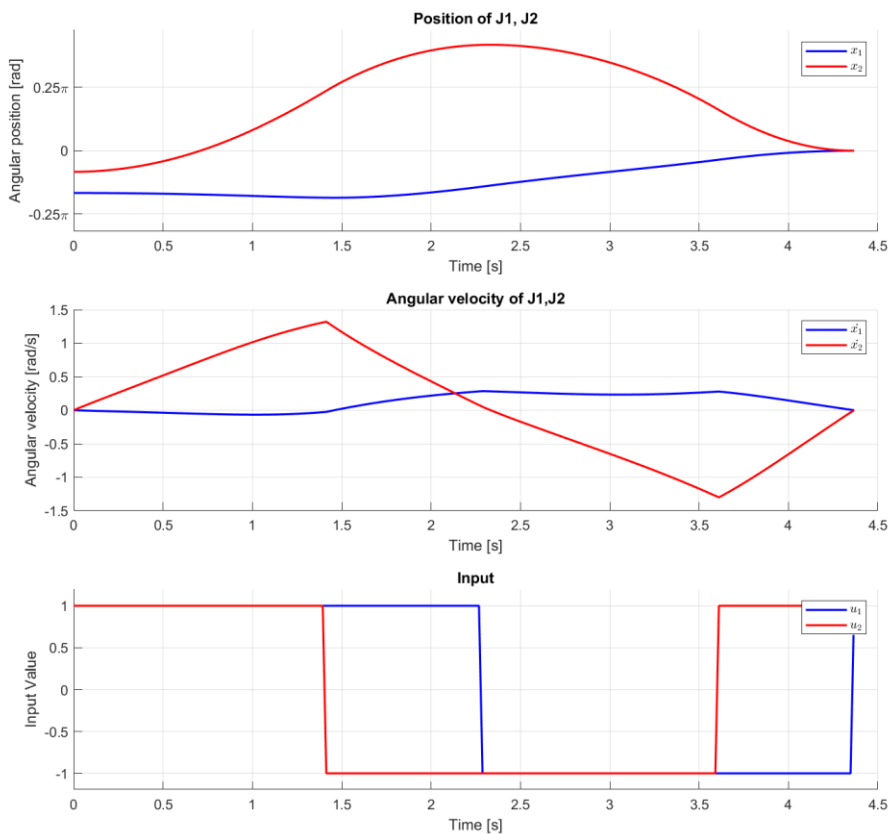
Kryterium zatrzymania algorytmu jest albo domyślnie ustalone w implementacji funkcji *fmincon*, albo jest nim maksymalna ilość iteracji wykonana przez algorytm gradientu prostego.

4.5. Wyniki i analiza rezultatów

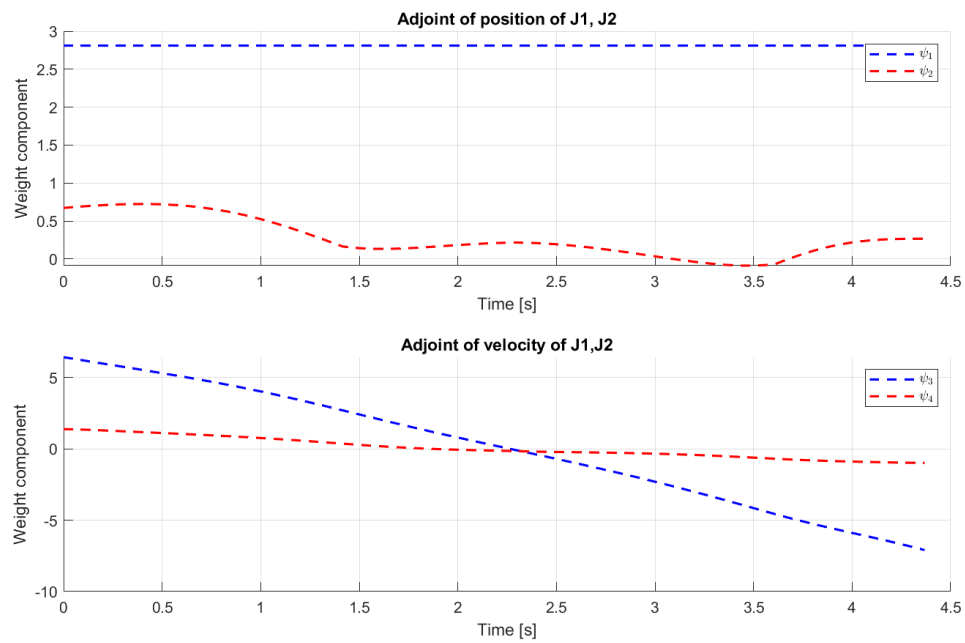
Wyniki przeprowadzonych testów dla różnych parametrów przedstawiono na rysunkach 4.1-4.4:
Dla parametru $X_0 = [-0.52; -0.26; 0; 0]$:



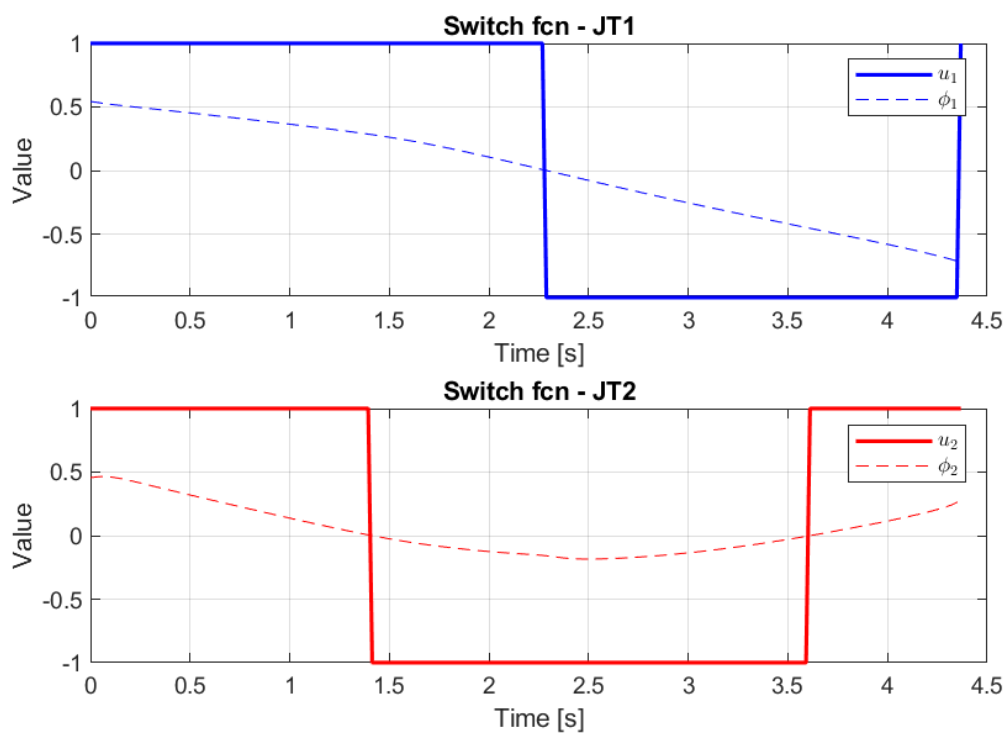
Rysunek 4.1a Wykres położenia



Rysunek 4.1b Wykresy zmiennych stanu

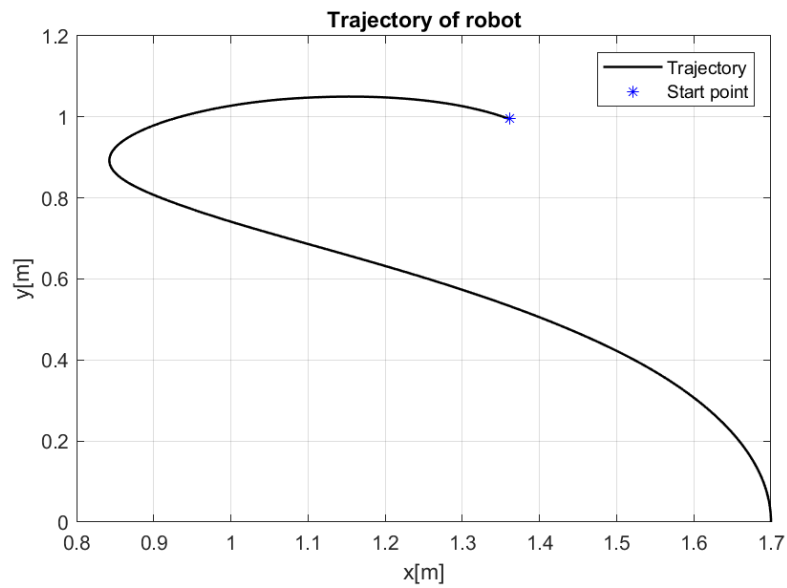


Rysunek 4.1c Wykres zmiennych sprzężonych.

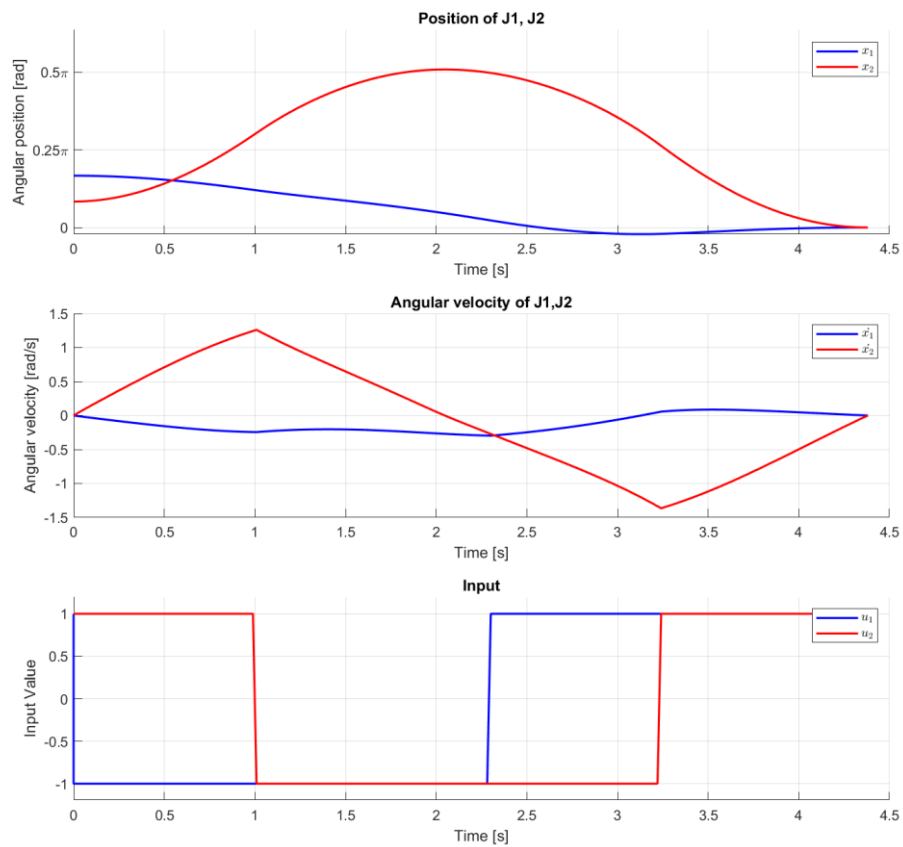


Rysunek 4.1d Wykres funkcji przełączającej dla

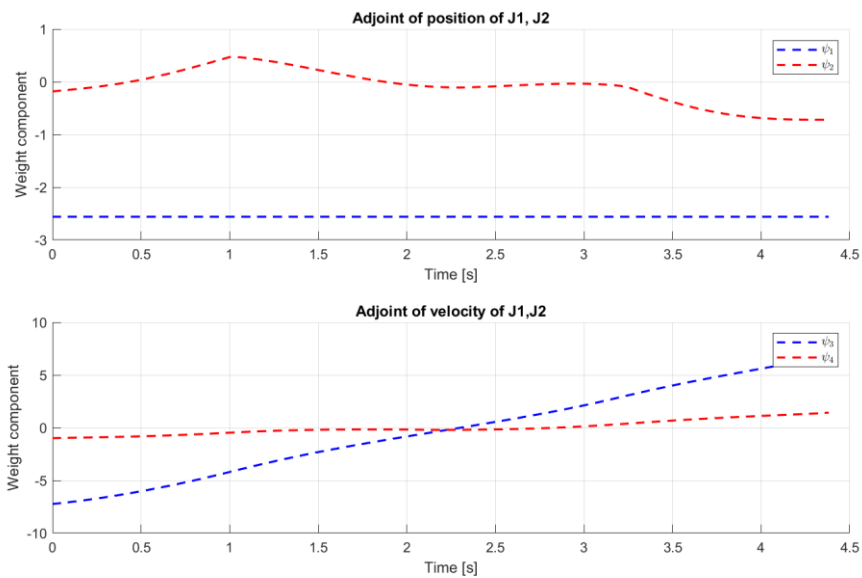
Dla parametru $X_0 = [-0.52; -0.26; 0; 0]$:



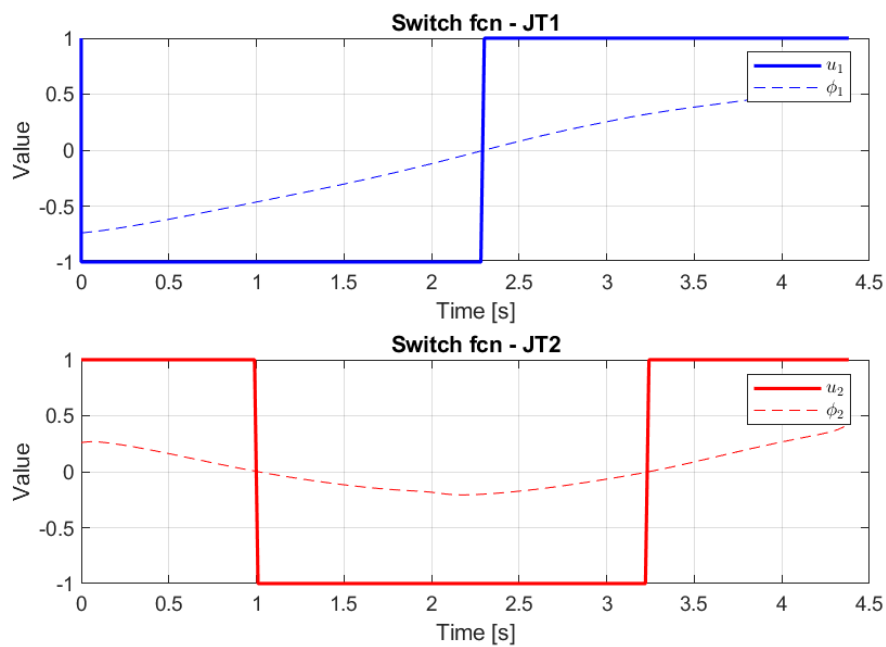
Rysunek 4.2a Wykres położenia



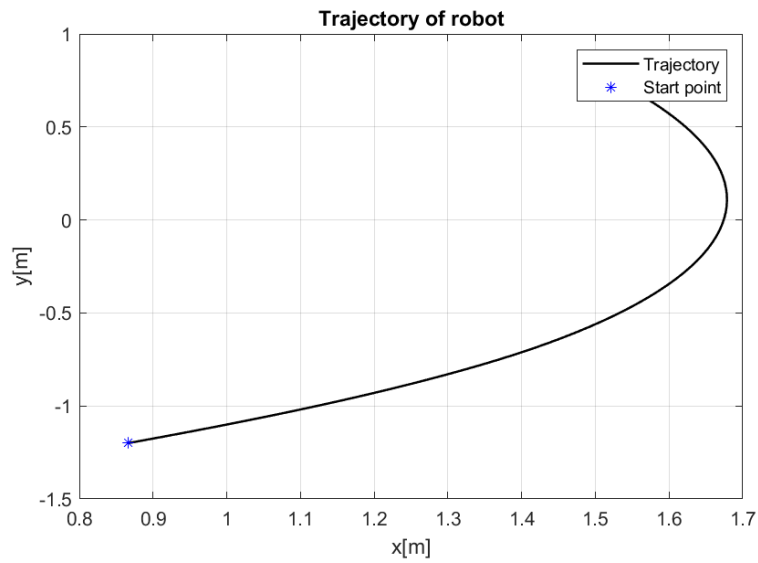
Rysunek 4.2b Wykresy zmiennych stanu



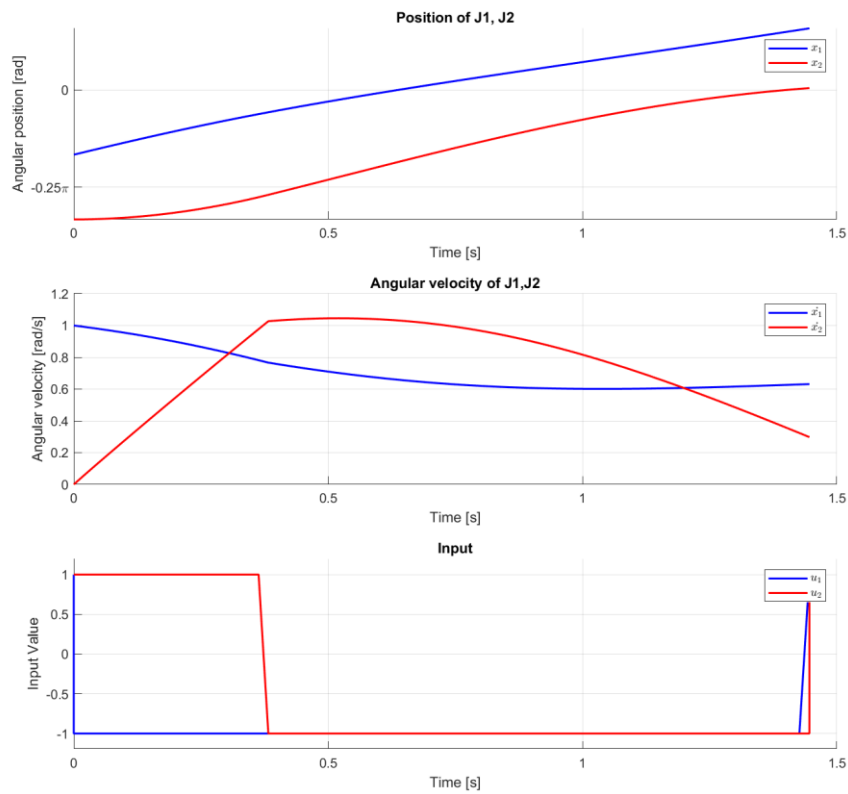
Rysunek 4.2c Wykres zmiennych sprzężonych



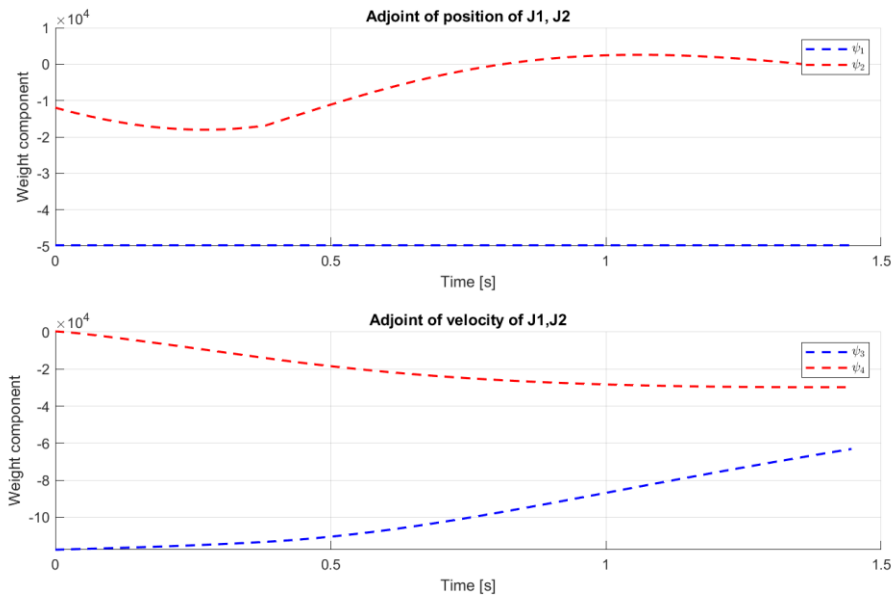
Rysunek 4.2d Wykres funkcji przełączających

Dla parametru $X_0 = [-0.52; -1.05; 0; 0]$ 

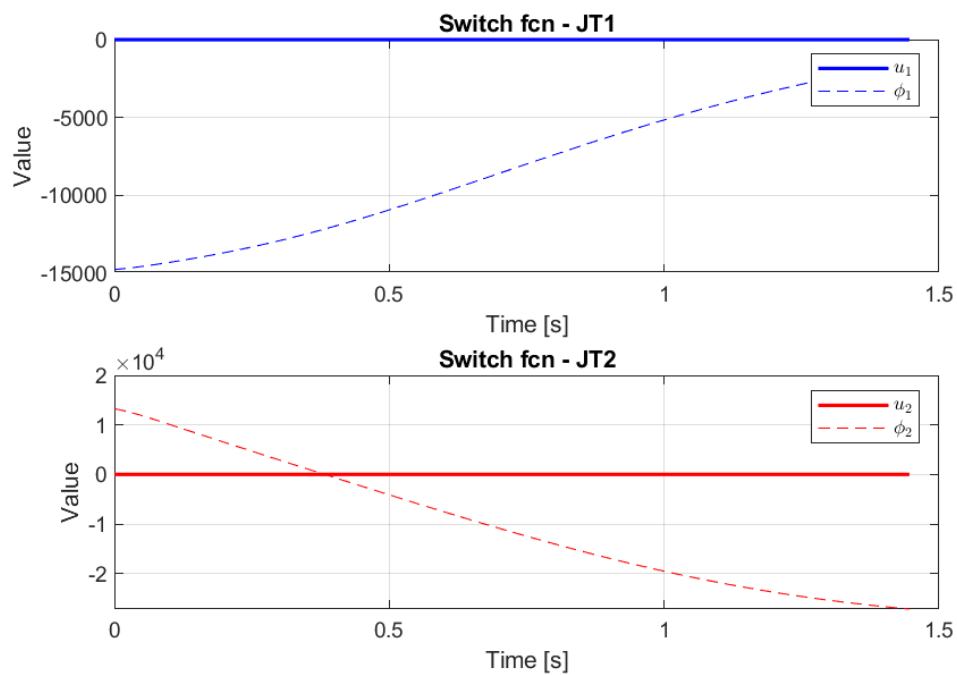
Rysunek 4.3a Wykres położenia



Rysunek 4.3b Wykres zmiennych stanu



Rysunek 4.3c Wykres zmiennych sprzężonych



Rysunek 4.3d Wykres funkcji przełączającej

5. Sterowanie z optymalizacją kwadratowej funkcji celu

5.1. Funkcja celu

Funkcja celu w tym wariantcie problemu została dobrana tak, żeby gwarantować Funkcja celu w tym problemie jest zadana wzorem:

$$J(x, u) = (x(T) - x_r)^T W_{stat} (x(T) - x_f) + \int_0^T [u(t)^T R u(t) + (x(t) - x_{ref}(t))^T W (x(t) - x_{ref}(t))] dt$$

5.2. Hamiltonian i równania sprzężone – modyfikacja

Hamiltonian układu, w przypadku wystąpienia w funkcji celu składnika całkowego, przyjmie w tym problemie postać zadaną wzorem poniżej – wyrażenie podcałkowe staje się składnikiem hamiltonianu:

$$H(x, \psi, u_1, u_2) = \psi^T (f(x) + g_1(x)u_1 + g_2(x)u_2) - L(x, u)$$

$$H(x, \psi, u_1, u_2) = \psi^T (f(x) + g_1(x)u_1 + g_2(x)u_2) + u_1^T r u_1 + u_2^T r u_2 + (x - x_{ref}(t))^T W (x - x_{ref}(t))$$

Równania sprzężone mają postać:

$$\dot{\psi} = -\frac{\partial H(\psi, x, u_1, u_2)}{\partial x} = -A\psi - \frac{\partial L(x, u)}{\partial x}$$

Gdzie macierz A jest zdefiniowana jak w zadaniu czasooptymalnym, natomiast dodatkowy składnik jest równy:

$$\frac{\partial L(x, u)}{\partial x} = W * (x(t) - x_{ref}(t))$$

Funkcja przełączająca (której nie można już nazywać funkcją przełączającą, ze względu na to, że rozwiązaniem zadania optymalizacji będzie zazwyczaj *sterowanie osobliwe*), dla tego przypadku przybierze postać:

$$\phi_1(t) = \psi^T g_1(t) + r u_1 \quad \phi_2(t) = \psi^T g_2(t) + r u_2$$

5.3. Metodyka rozwiązania równań różniczkowych

Do rozwiązania zadania optymalizacji konieczne jest zastosowanie pewnej aproksymacji sterowania – w tym zadaniu skorzystano z faktu, że obliczenia są wykonywane z dość dużą dokładnością dla metody Rungego-Kutty 4 rzędu już dla dużych kroków (0.1s) i uznano, że do skutecznej optymalizacji wystarczy wyznaczyć sterowanie dla każdej chwili czasowej z rozdzielczością równą rozdzielczości symulacji.

Aby w efektywny sposób rozwiązać zadanie optymalizacji, konieczne jest także obliczenie całki znajdującej się w definicji funkcji celu. Jest to osiągnięte poprzez dołożenie dodatkowego równania do równań stanu:

$$\dot{x}_5 = L(x, u) = r(u_1^2 + u_2^2) + (x(t) - x_{ref}(t))^T W (x(t) - x_{ref}(t))$$

Rozwiązanie tak zdefiniowanych równań stanu sprawi, że oprócz odpowiedzi układu na zadane sterowanie, zostanie także otrzymana aproksymacja całki występującej w def. funkcji celu.

Do skutecznego rozwiązania zadania optymalizacji konieczne jest także zdefiniowanie odpowiednich gradientów hamiltonianu:

$$\frac{\partial J(x, u)}{\partial u_{1,i}} = - \int_{t_i}^{t_{i+1}} \phi_1(t) dt \quad \frac{\partial J(x, u)}{\partial u_{2,i}} = - \int_{t_i}^{t_{i+1}} \phi_2(t) dt$$

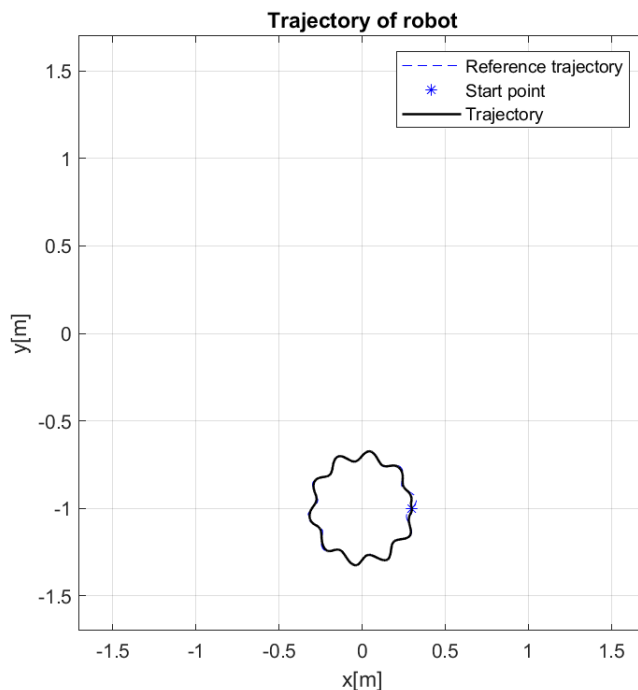
W tym przypadku również konieczne jest obliczenie pewnej całki, w definicji równań sprzężonych zostały dołączone dwa dodatkowe równania:

$$\dot{\psi}_5 = \phi_1(t) = \psi^T * g_1(x) + ru_1 \quad \dot{\psi}_6 = \phi_2(t) = \psi^T * g_2(x) + ru_2$$

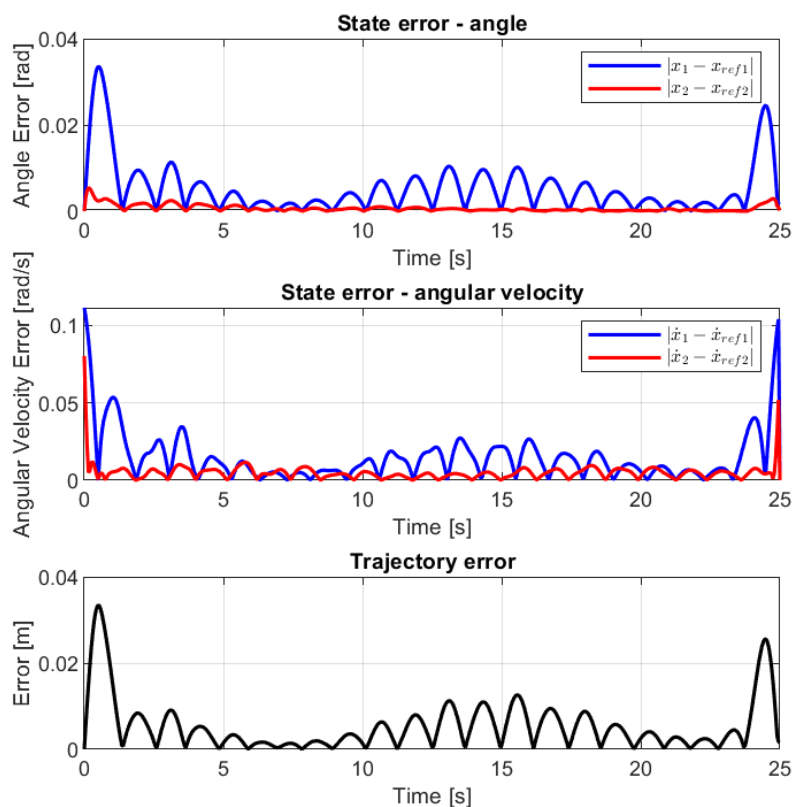
Pozostała część obliczeń wykonywana jest w taki sam sposób, co w przypadku regulatora czasooptymalnego.

5.4. Wyniki dla różnych trajektorii dla zadania podążania za trajektorią

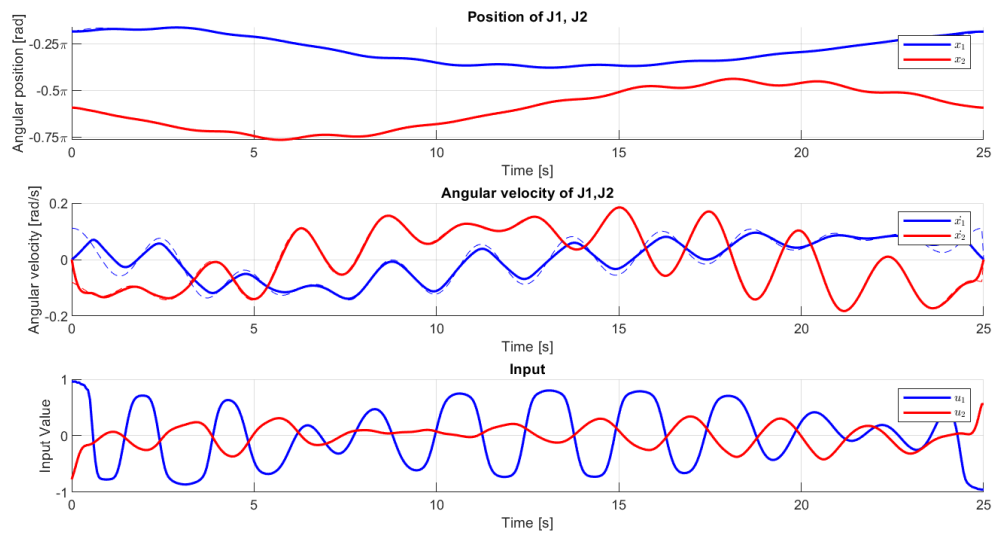
Dla parametrów: T=25; H=25; S=25, R=0.3 - *wobble*



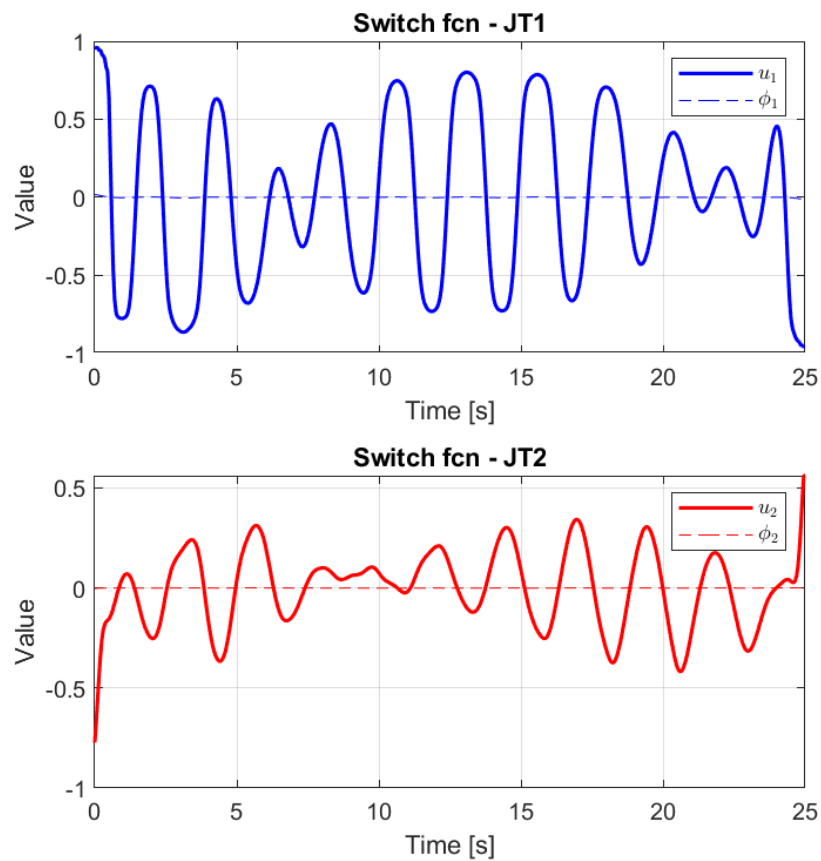
Rysunek 5.1a Wykres położenia



Rysunek 5.1b Wykres błędu

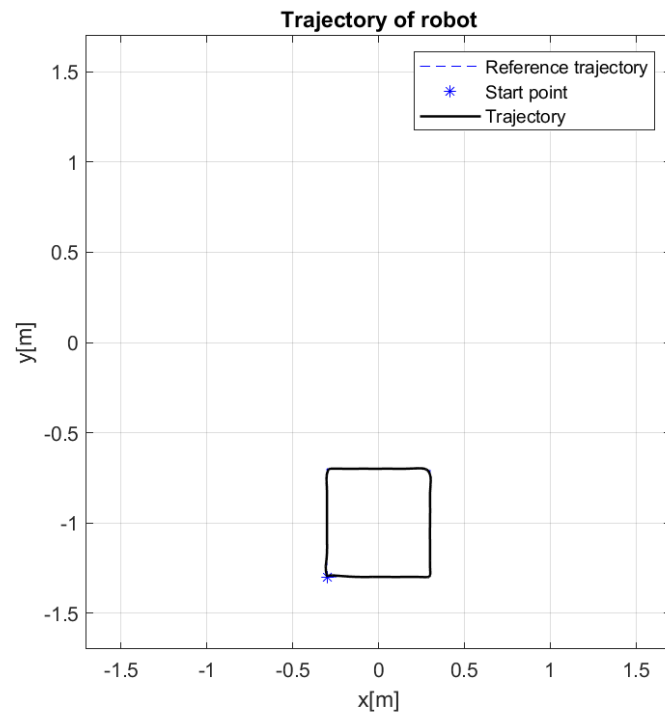


Rysunek 5.1c Wykres zmiennych stanu

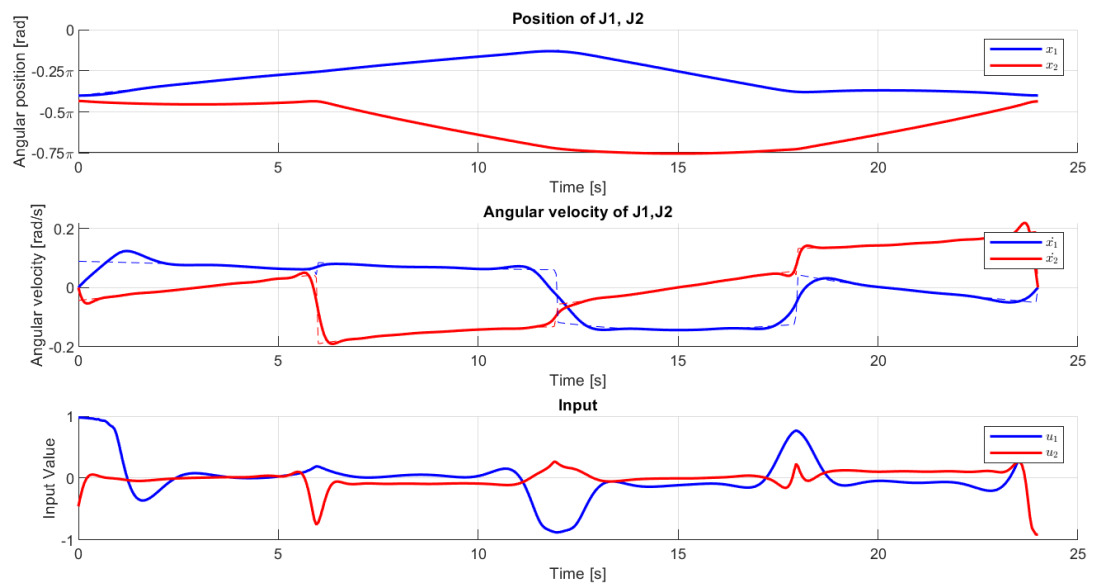


Rysunek 5.1d Wykres funkcji przełączających

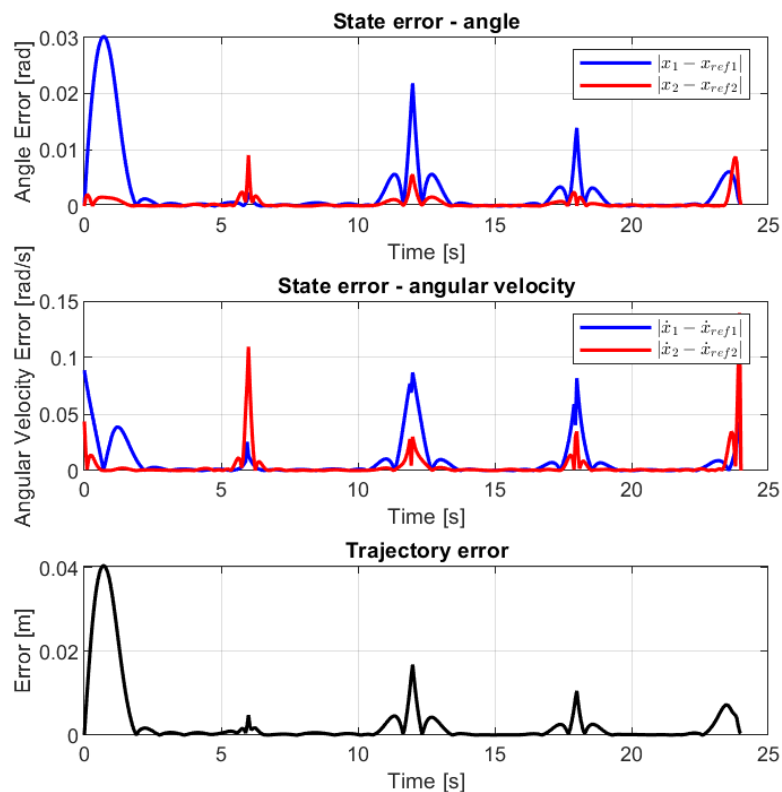
Dla parametrów: $T=24$; $H=24$; $S=24$, $R=0.3$ - square



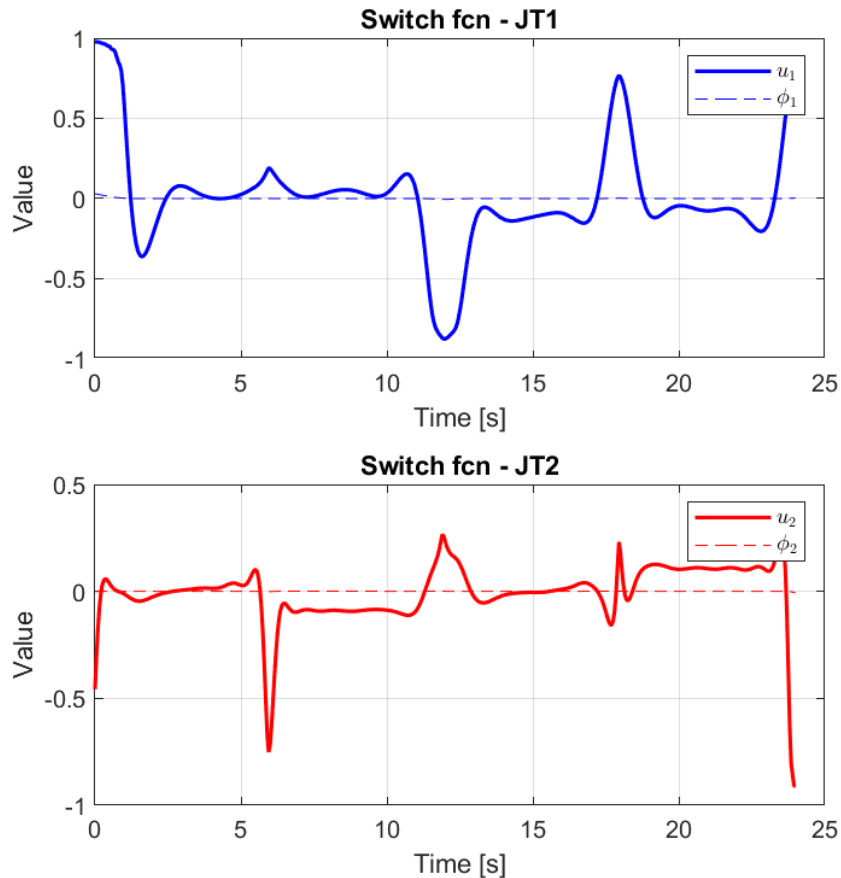
Rysunek 5.2a Wykres położenia



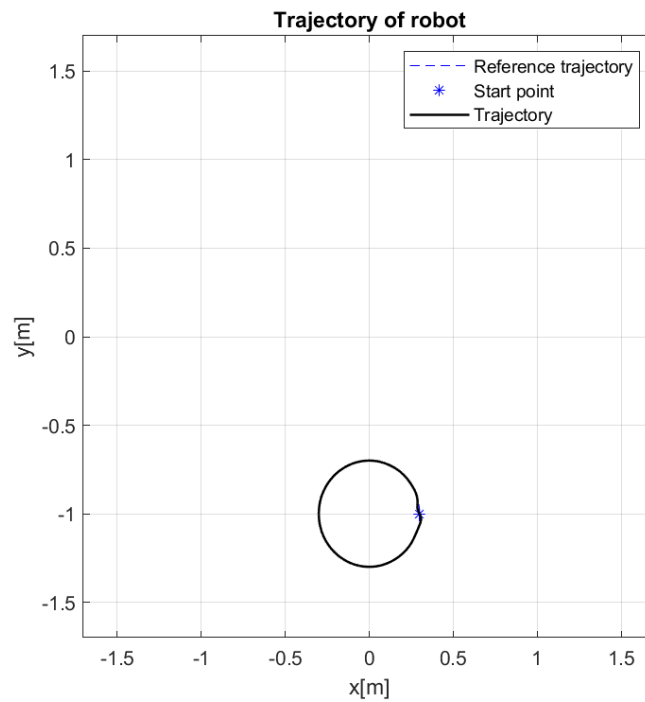
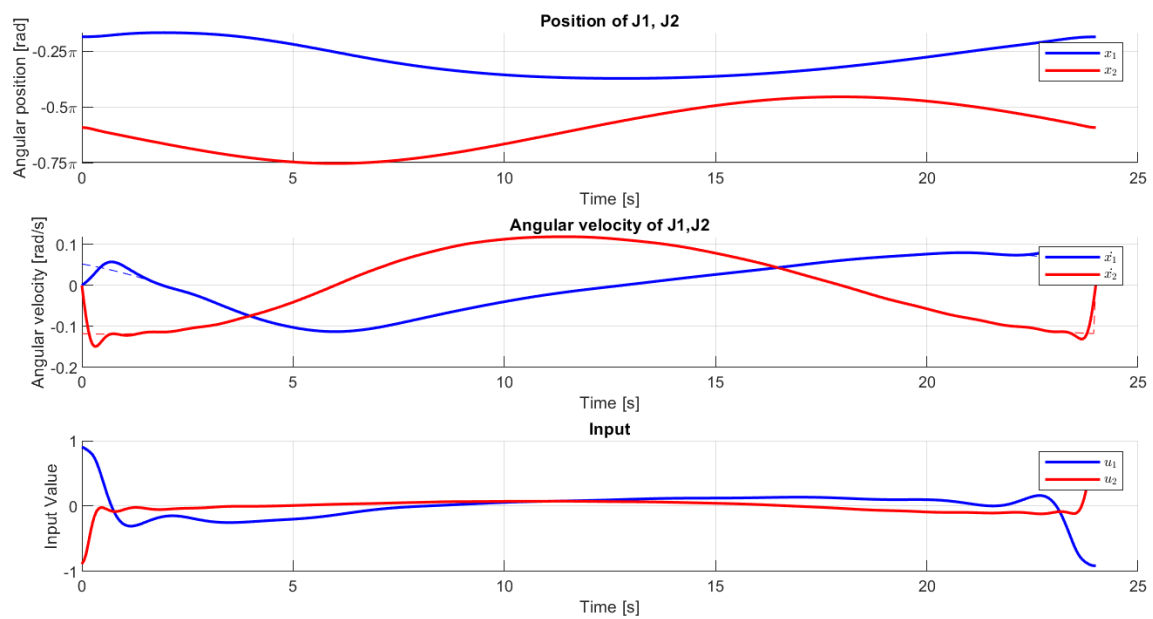
Rysunek 5.2b Wykres zmiennych stanu

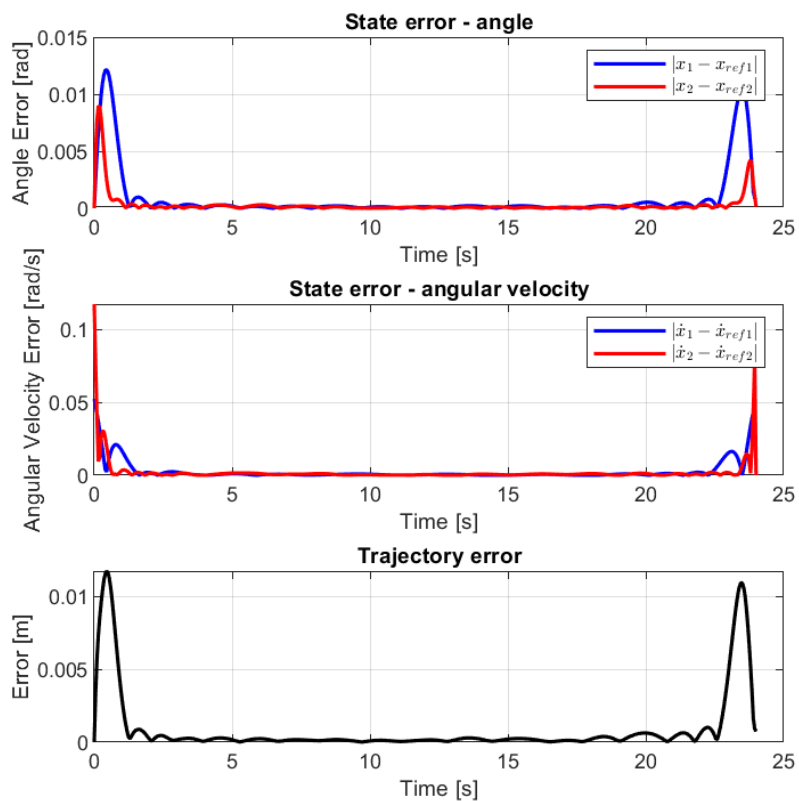


Rysunek 5.2c Wykres błędu

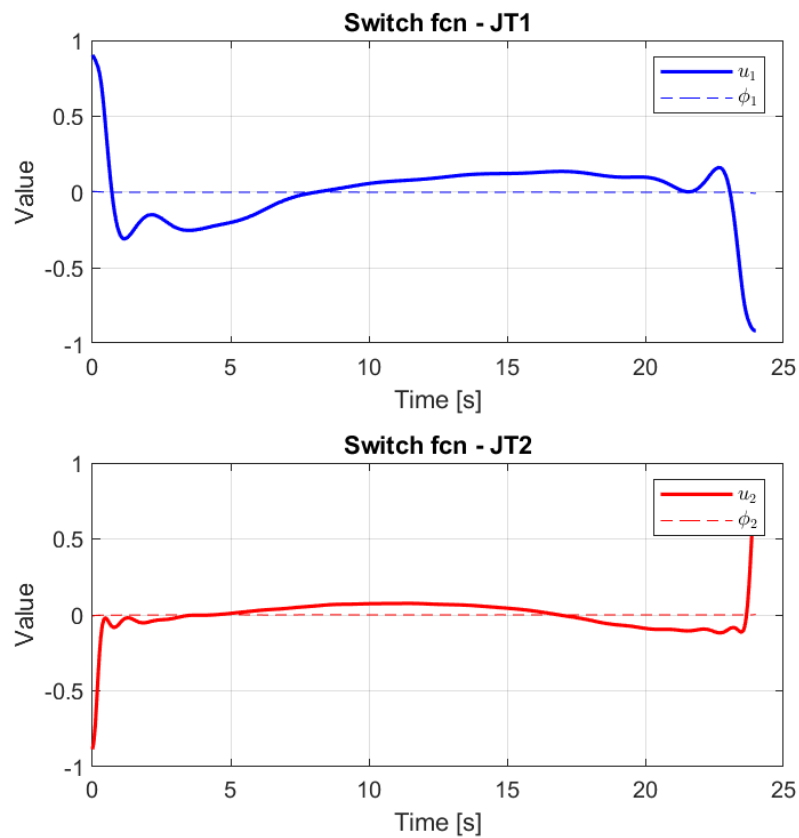


Rysunek 5.2d Wykres funkcji przełączających

Dla parametrów: $T=24$; $H=24$; $S=24$, $R=0.3$ - circle*Rysunek 5.3a Wykres położenia**Rysunek 5.3b Wykres zmiennych stanu*



Rysunek 5.3c Wykres błędu

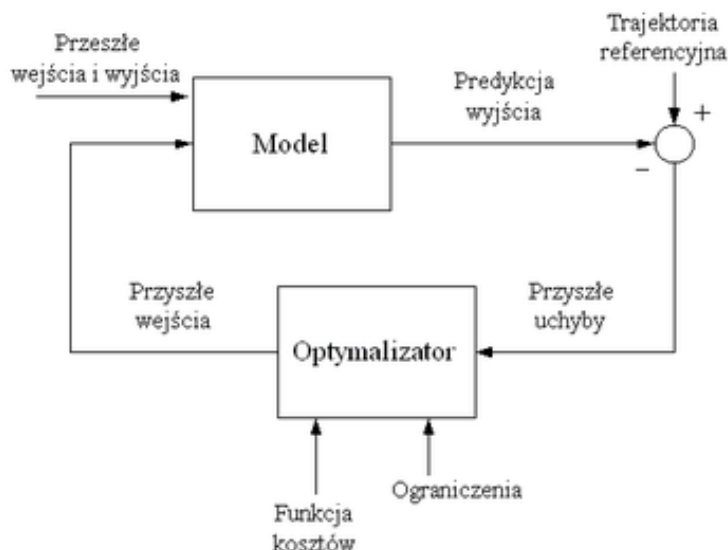


Rysunek 5.3d Wykres funkcji przełączających

6. Sterowanie predykcyjne – Model Predictive Control

6.1. Idea regulacji predykcyjnej o skończonym horyzoncie

Regulator *Model Predictive Control*, w przeciwieństwie do tradycyjnych regulatorów ze sprzężeniem zwrotnym, dostosowuje swoje działanie z pewnym wyprzedzeniem, zanim nastąpią stosowne zmiany w wielkościach mierzonych (obserwowanych) układu. Metoda ta polega na cyklicznym rozwiązywaniu zadania optymalizacji z warunkiem początkowym równym zaobserwowanemu stanowi obiektu.



Rys 6.1 Schemat działania regulatora *Model Predictive Control*

Algorytm działania regulatora Model Predictive Control opiera się na iteracyjnym wykonaniu następującego algorytmu:

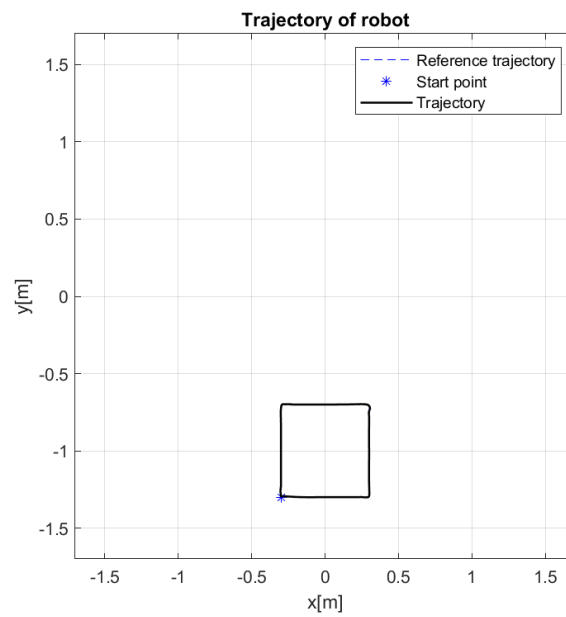
- rozwiązanie zadania optymalizacji przy zadanym *horyzoncie* optymalizacji,
- zastosowanie na wejście obiektu części obliczonego, optymalnego sterowania – w wersji dyskretnej MPC jest to jedna wartość, natomiast w tym problemie zaimplementowano zastosowanie na wejście obiektu określonej długości sygnału sterowania (*step*),
- przesunięcie horyzontu o wartość kroku algorytmu (*step*),
- Jeśli został osiągnięty zadany czas regulacji, algorytm automatycznie skraca horyzont predykcji, natomiast w przypadku, gdy horyzont predykcji osiąga wartość 0 – algorytm kończy swoje działanie.

Ze względu na skończony horyzont predykcji, wymiarowość zadania optymalizacji zmniejsza się, dzięki czemu, w przeciwieństwie do algorytmów omawianych w poprzednich punktach, algorytm ten pozwala na obliczanie, przy odpowiednio zaawansowanym kontrolerze sterującym, kolejnych sterowań w czasie rzeczywistym.

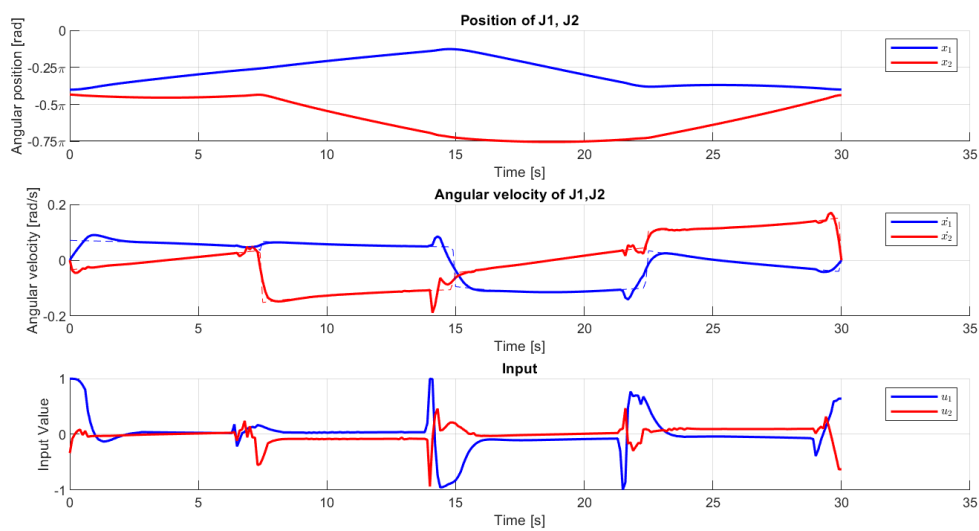
6.2. Implementacja MPC ze sterowaniem optymalnym – rezultaty

Otrzymane wyniki zaprezentowano na rysunkach 6.2-6.4.

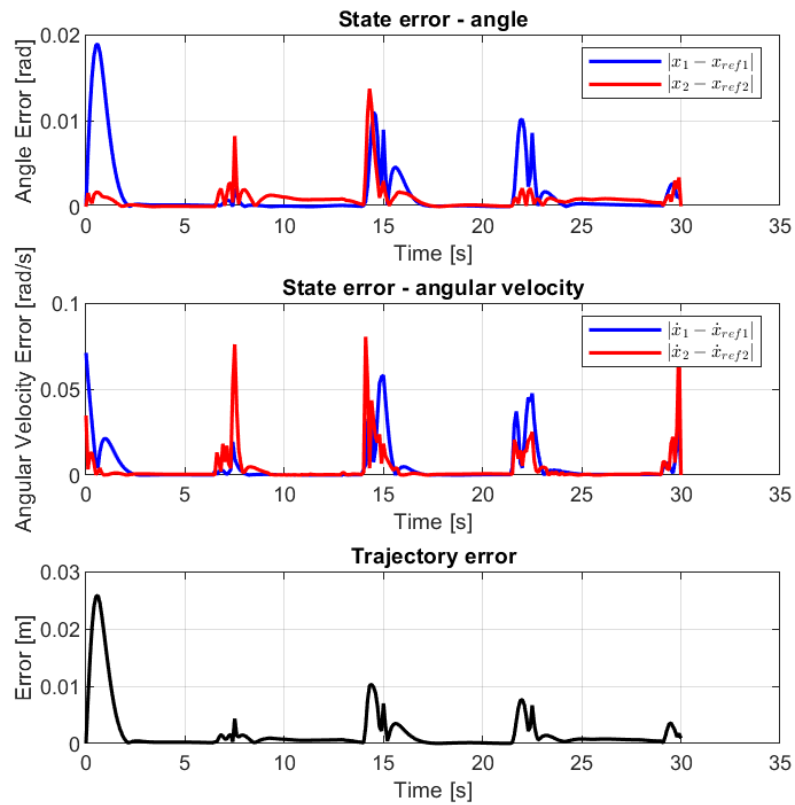
Kształt – kwadrat:
Horyzont = 1



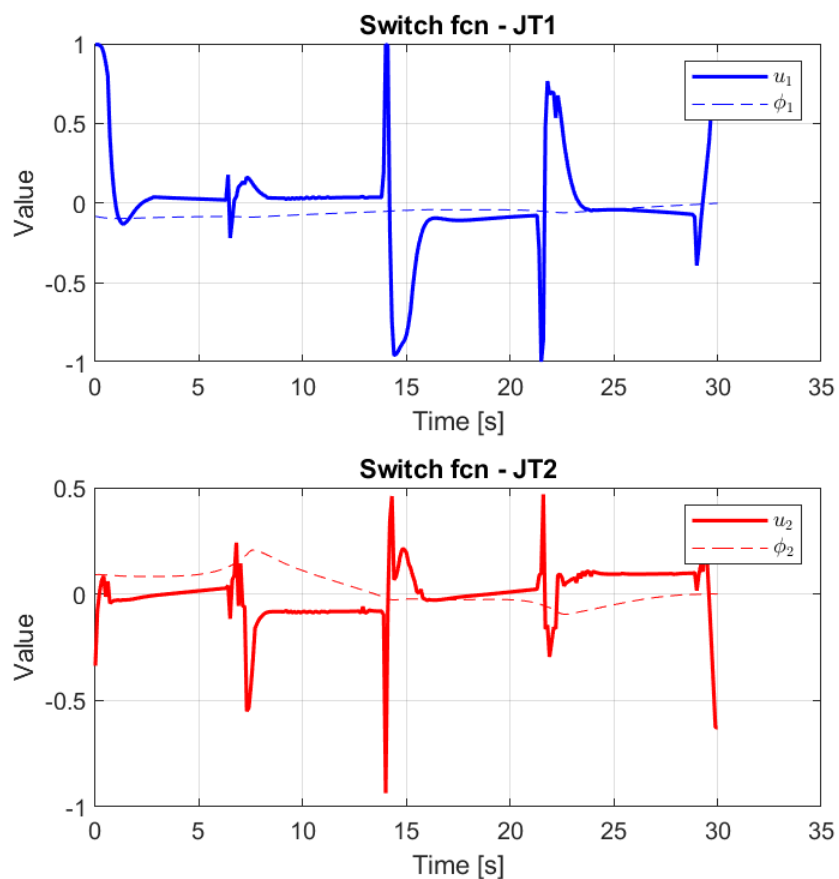
Rysunek 7.2a Wykres położenia



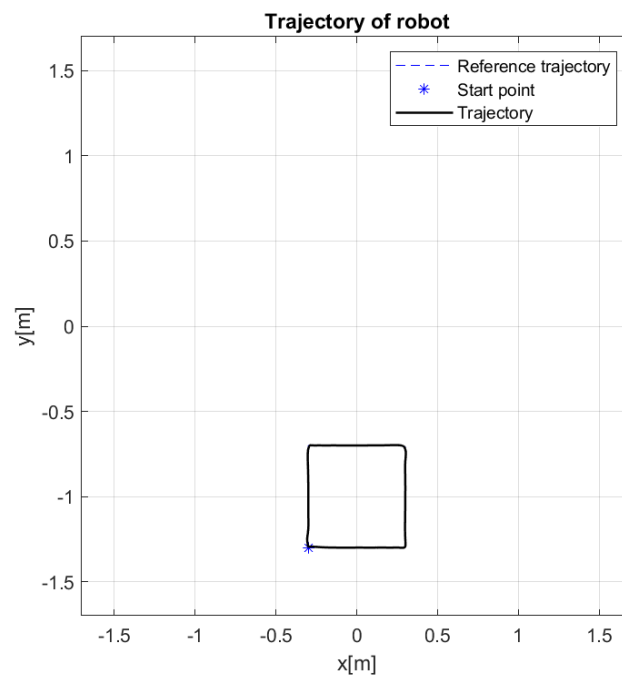
Rysunek 7.2b Wykres zmiennych stanu



Rysunek 7.2c Wykres błędów

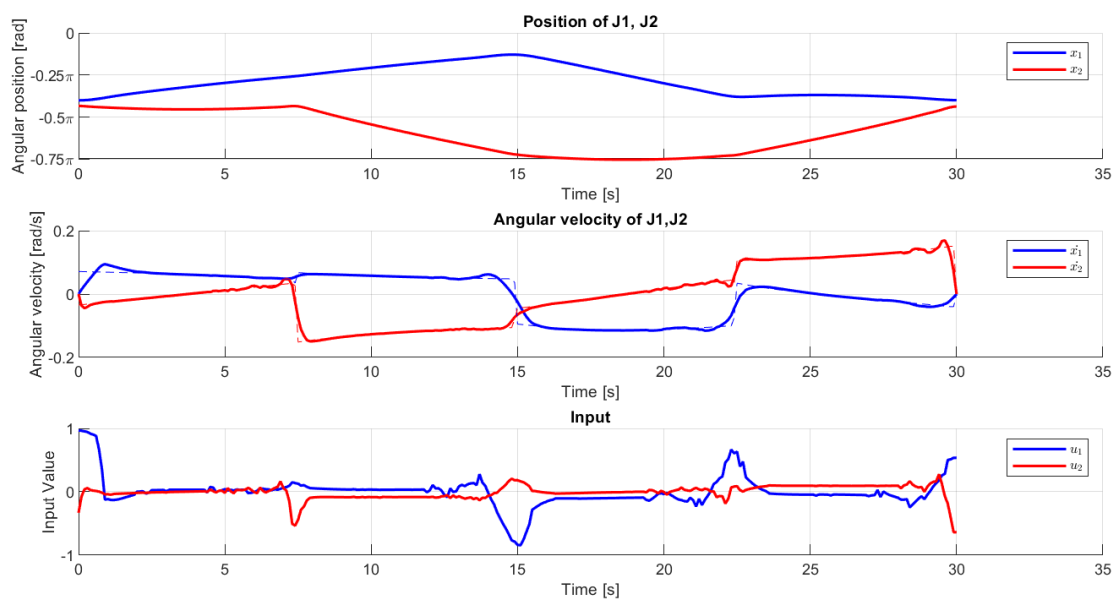


Rysunek 7.2d Wykres funkcji przełączających

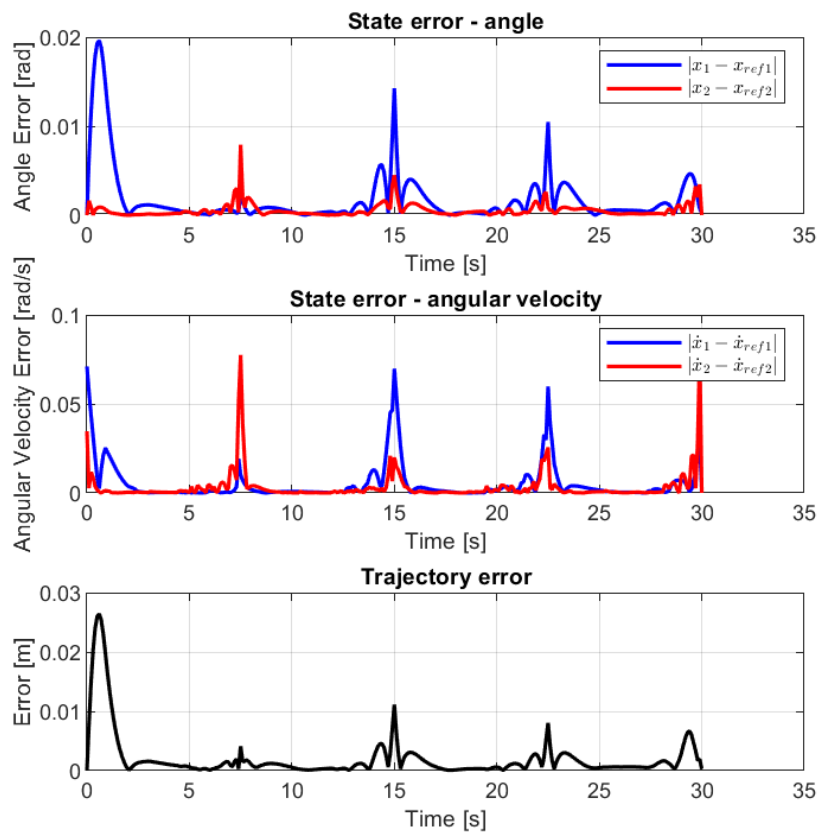


Rysunek 7.3a Wykres położenia

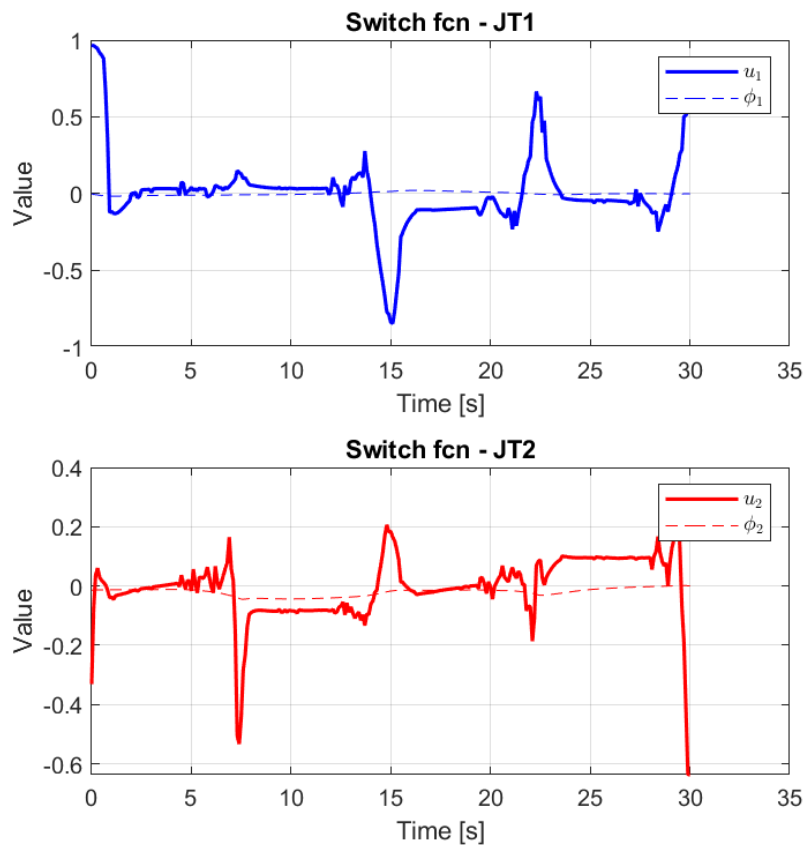
Horyzont = 3



Rysunek 7.3b Wykres zmiennych stanu



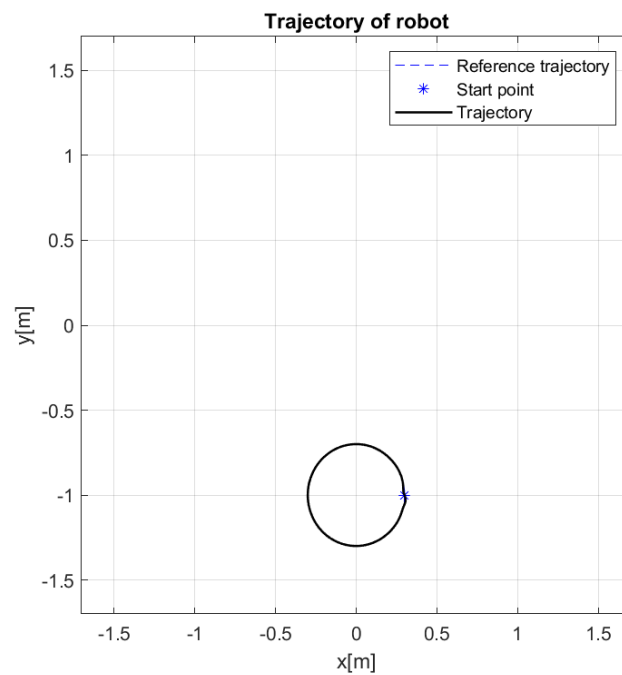
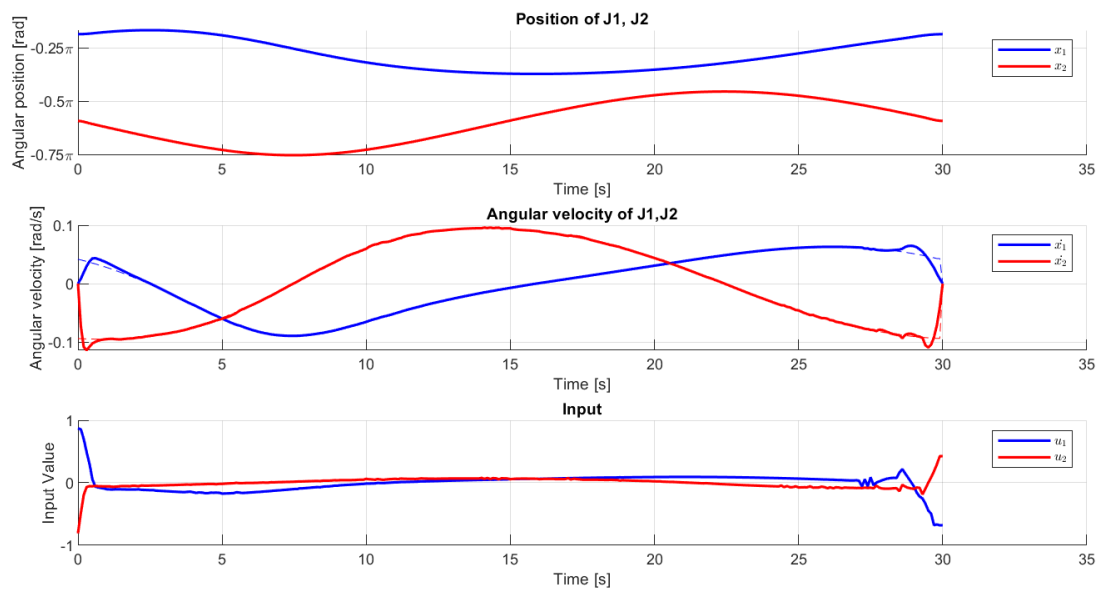
Rysunek 7.3c Wykres błędów

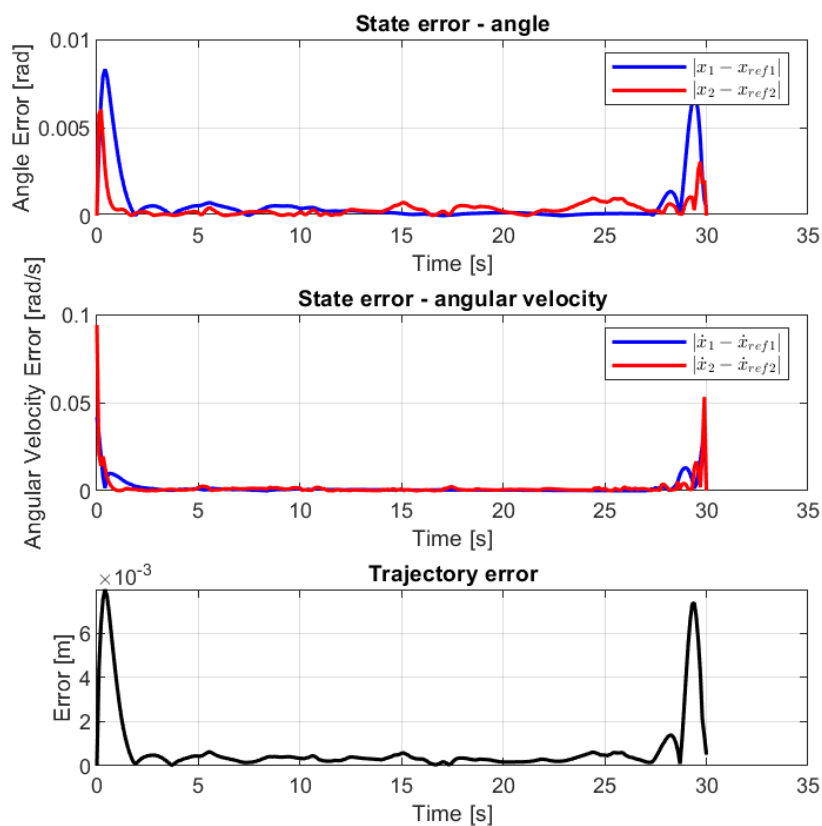


Rysunek 7.3d Wykres funkcji przełączających

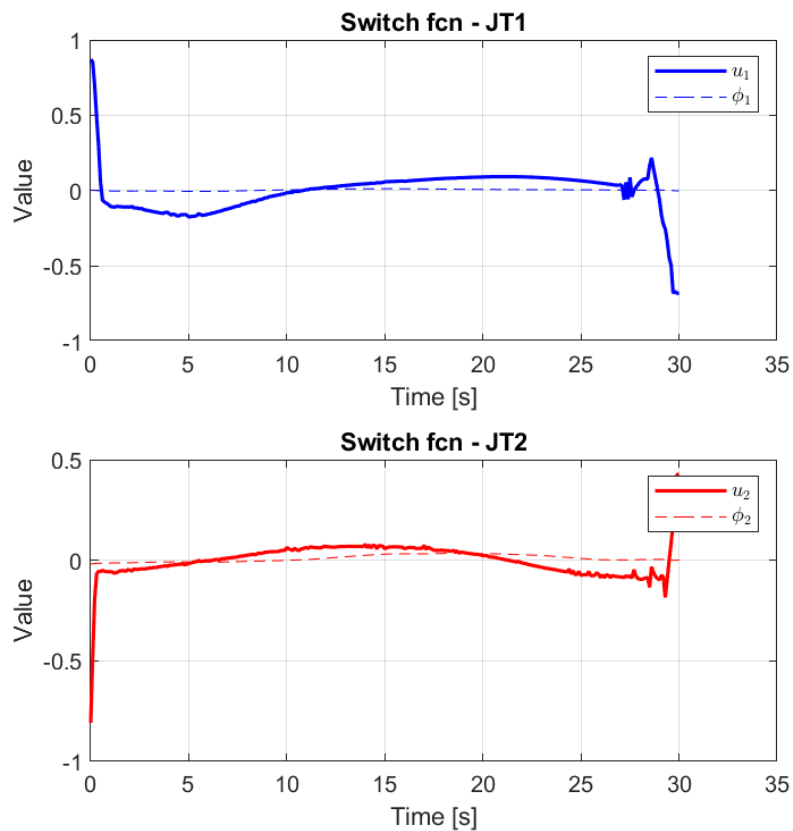
Kształt – koło:

Horyzont=3

*Rysunek 7.4 Wykres położenia**Rysunek 7.4b Wykres zmiennych stanu*



Rysunek 7.4c Wykres błędu



Rysunek 7.4d Wykres funkcji przełączających

7. Wnioski

W ramach ćwiczeń laboratoryjnych udało się zaimplementować oprogramowanie pozwalające na wyznaczenie sterowania optymalnego dla modelu matematycznego robota SCARA oraz przeprowadzić symulacje w celu weryfikacji otrzymanych wyników.

Analizując poszczególne wykresy można dojść do wniosku, iż postawione zadanie zostało poprawnie wykonane – następuje minimalizacja zadanej funkcji celu. W przypadku wykresy błędu zauważalne są znaczące wzrosty wartości bezwzględne, potencjalnie może to być spowodowane doбором silników o tej samej mocy, gdzie w rzeczywistości pierwszy silnik powinien mieć większą moc, ponieważ musi wprawić w ruch większą masę. Porównując kształt charakterystyk położenia (szczególnie widoczne dla trajektorii o kształcie kwadratu) widać pewną zależność – dla większego horyzontu sterowania rogi trajektorii są zaokrąglone. Dzieje się tak ponieważ dążymy do najlepszej trajektorii. Algorytm widząc, że kolejne punkty znajdują się „dalej” od rogu trajektorii skróci go uzyskując optymalniejsze sterowanie zarówno pod względem czasu jak i sterowania kosztem dokładności odwzorowania trajektorii.