



Algorytmy wysokiej dokładności śledzenia trajektorii robota przemysłowego

mgr inż. Wojciech Zwonarz

promotor: prof. dr hab. inż. Andrzej Turnau

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

05.12.2019

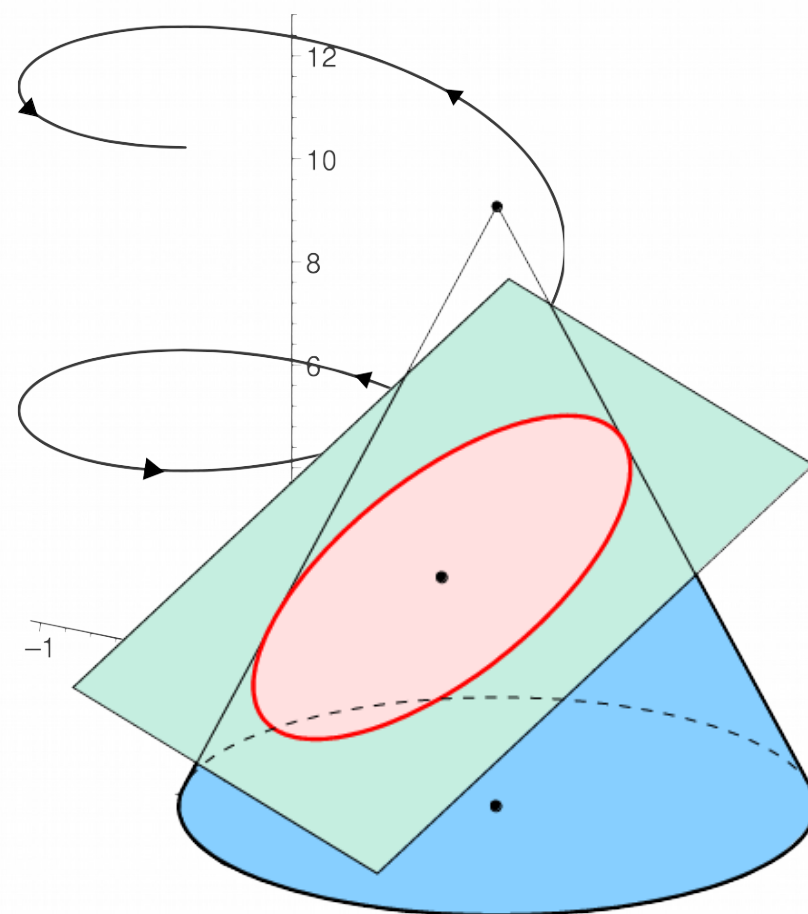
Agenda

- Trajektorie w robotyce
- Interpolacje Bezierra dla zmniejszenia ilości obliczeń
- Kinematyka i dynamika robotów
- Typowe algorytmy regulacji
- Sterowanie predykcyjne dla śledzenia trajektorii
- Funkcje kary za zbliżanie się do przeszkody i ograniczeń oraz za niedokładne śledzenie
- Podsumowanie

Trajektorie ruchu

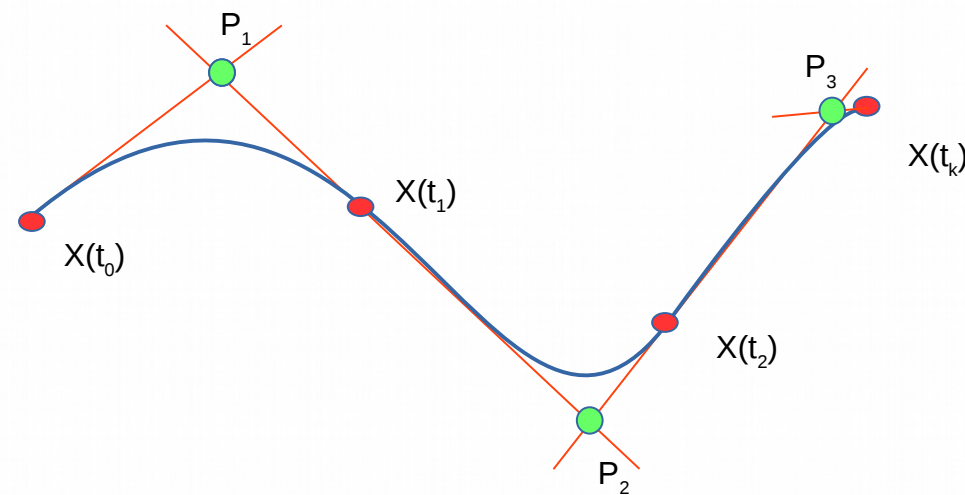
W pracy opisano trzy elementarne trajektorie:

- Helisa – linia śrubowa
- Elipsa
- Prosta



Interpolacja krzywymi Bezierra

Interpolacja typu spline krzywymi Bezierra drugiego rzędu okazała się być najdokładniejszym sposobem odwzorowania trajektorii w prostej postaci wielomianowej.



Krzywe Bezierra drugiego rzędu

Krzywe B-sklejane są to „sklejone” wielomiany niskiego stopnia, zwykle nie większego niż trzeci. Są one regularne w rzędzie zależnym od stopnia wielomianu.

Podstawowe wzory opisujące aproksymowaną drogę efektora mają podaną postać.

$$p(t) = \sum_{i=0}^{m-n-1} p_i N_i^n(t), \quad t \in \langle u_n, u_{m-1} \rangle$$

$$N_i^0(t) = \begin{cases} 1 & \text{dla } t \in [u_i, u_{i+1}] \\ 0 & \text{dla } t \notin [u_i, u_{i+1}] \end{cases}$$

$$N_i^n(t) = \frac{t - u_i}{u_{i+n} - u_i} N_i^{n-1}(t) + \frac{u_{i+n+1} - t}{u_{i+n+1} - u_{i+1}} N_{i+1}^{n-1}(t), \quad \text{dla } n > 0$$

$$p(t) = [P_1 \ P_2 \ P_3] \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^2 \\ t \\ 1 \end{bmatrix}$$

$$P_i = \left\{ \begin{array}{l} \frac{X(t_i) - X(t_{i-1}) - v(t_i)t_i + v(t_{i-1})t_{i-1}}{v(t_{i-1}) - v(t_i)} \quad \text{dla } v(t_{i-1}) \neq v(t_i) \\ \frac{X(t_i) - X(t_{i-1})}{2} \quad \text{dla } v(t_{i-1}) = v(t_i) \end{array} \right\}$$

Kinematyka i dynamika robotów

Macierz transformacji pomiędzy członami i oraz $i-1$.

Dynamika uzyskana przy pomocy równania Eulera-Lagrange'a. Na podstawie kinematyki uzyskanej przy pomocy notacji DH.

Wynikowe równanie jest nieliniowe względem zmiennych przegubowych.

Kinematyka:

$$A_i^{i-1} = RotZ(\Theta_i) TranZ(d_i) TranX(a_i) RotX(\alpha_i)$$

Dynamika:

$$L = E_k - E_p \quad \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau$$

$$E_K(q, \dot{q}) = \frac{1}{2} \dot{q}^T \sum_{i=1}^n \left[m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q) \right] \dot{q}$$

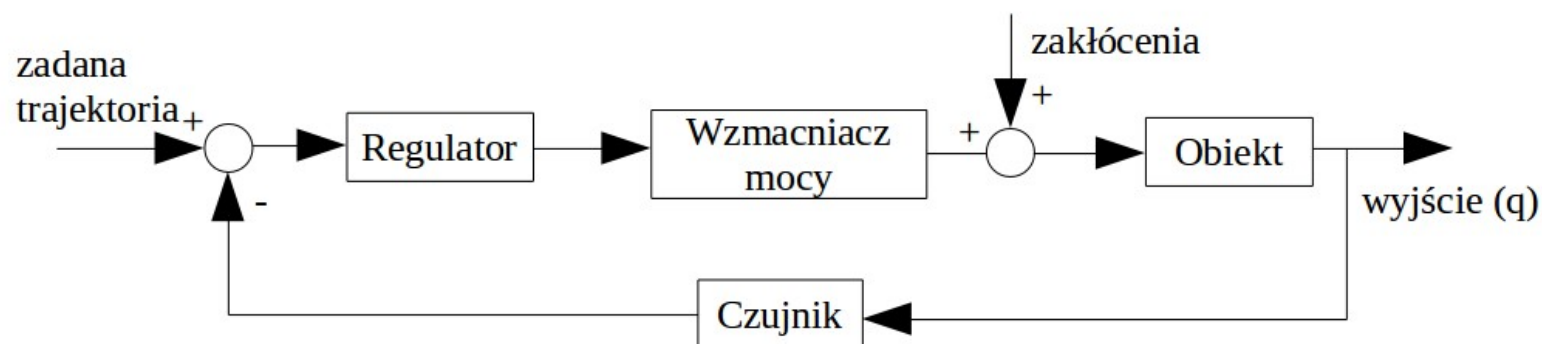
$$E_P(q) = \sum_{i=1}^n g^T r_{ci} m_i$$

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(q, \dot{q}) = \tau$$

Sterowniki robotów

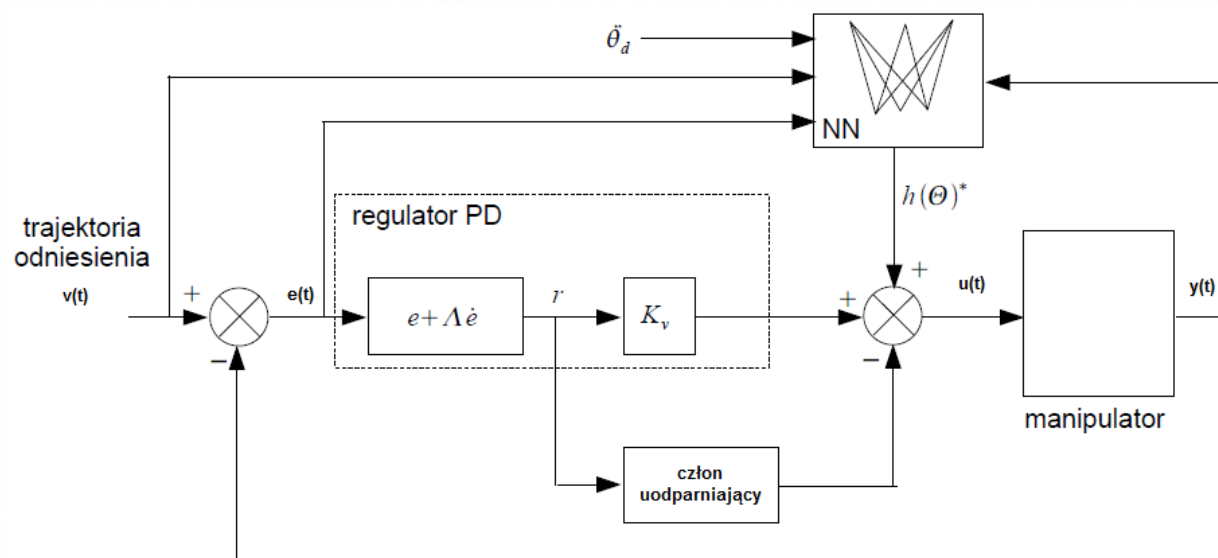
Typy regulatorów
omawiane w pracy

- Klasyczne
- Adaptacyjne
- Neuronowe
- Predykcyjne



Regulatory neuronowe

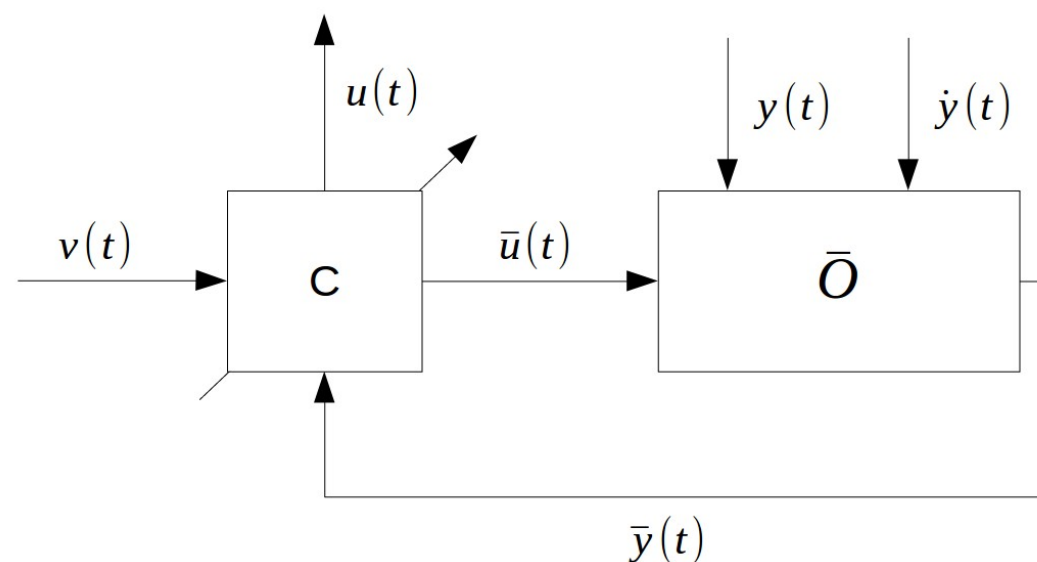
Regulatory wzbogacone o sieci neuronowe zwykle składają się z członu PD oraz sieci neuronowej ze sprzężeniem w przód, która pozwala na redukcję błędu od elementów nieliniowych dynamiki.



Regulatory predykcyjne

Regulator predykcyjny składa się z bloku predykcyjnego oraz bloku poszukiwania minimum funkcji kosztu sterowania.

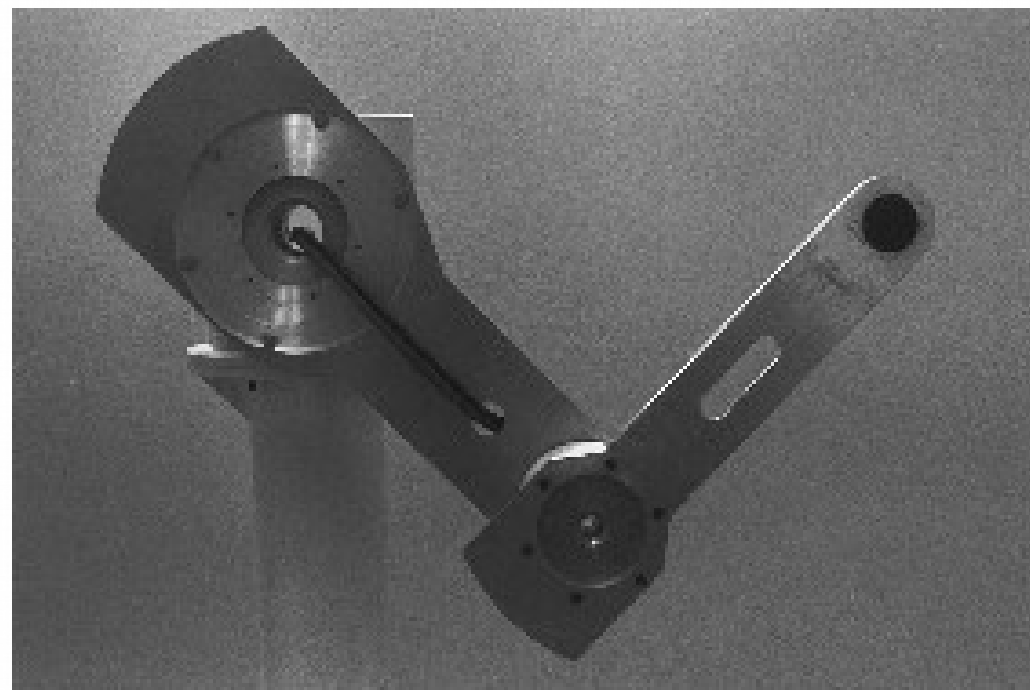
- $u(t)$ Wyznaczone sterowanie
- $v(t)$ Zadana trajektoria
- $\bar{u}(t)$ Próbné sterowanie
- $y(t)$ Wyjście obiektu
- $\bar{y}(t)$ Wyjście modelu obiektu
- $\dot{y}(t)$ Pochodna po czasie wyjścia obiektu



- C Regulator
- \bar{O} Model obiektu

Wybór najlepszego sterowania

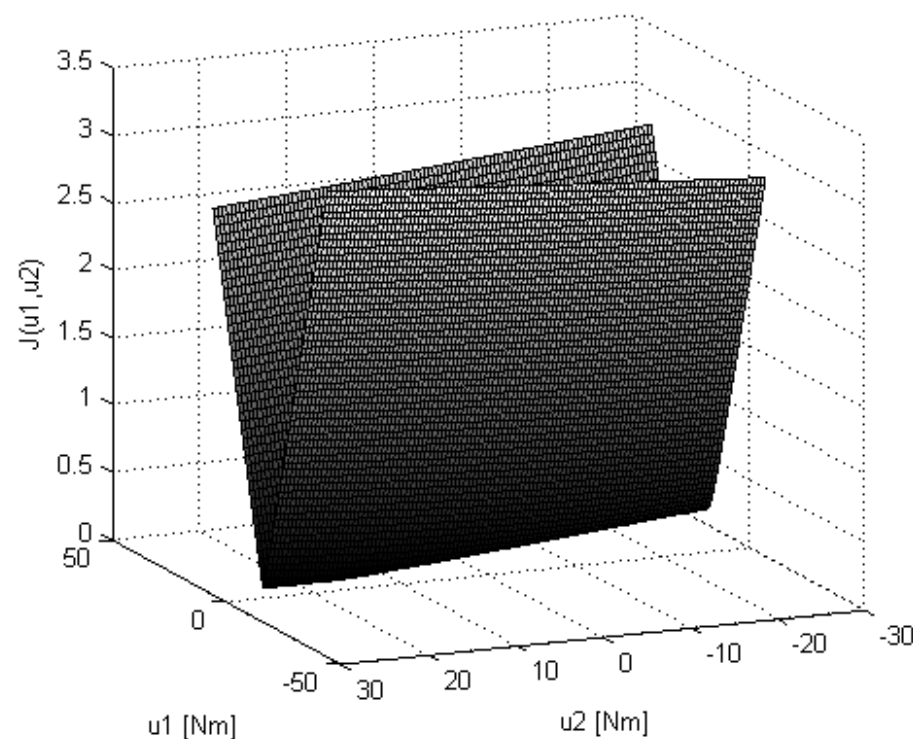
Robot dwuczłonowy jako
obiekt regulacji
predykcyjnej.



Funkcja kosztu

Koszt sterowania można opisać jako błąd na zadanym horyzoncie czasowym predykcji.

$$J(u_1, u_2, h) = \int_{t=0}^h |q(t) - v(t)|^2 dt$$

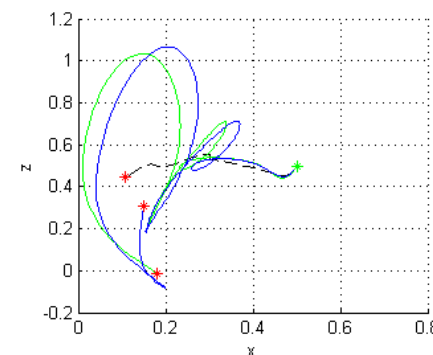
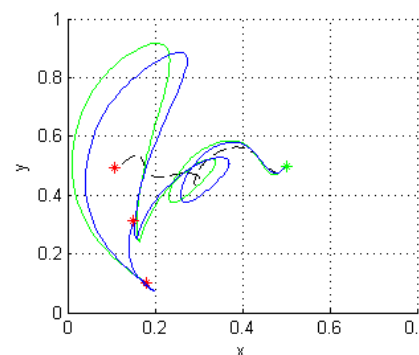
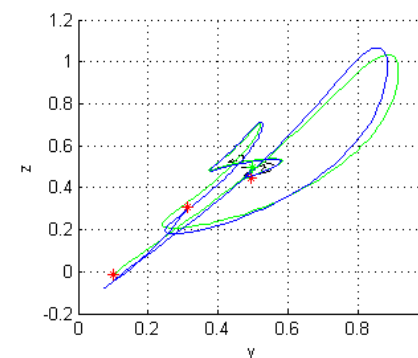
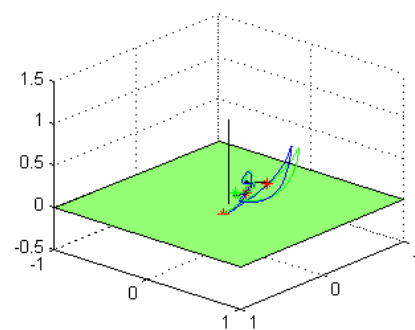


Predykcja numeryczna

Predykcja używa metod numerycznych.

Najprostsza predykcja:

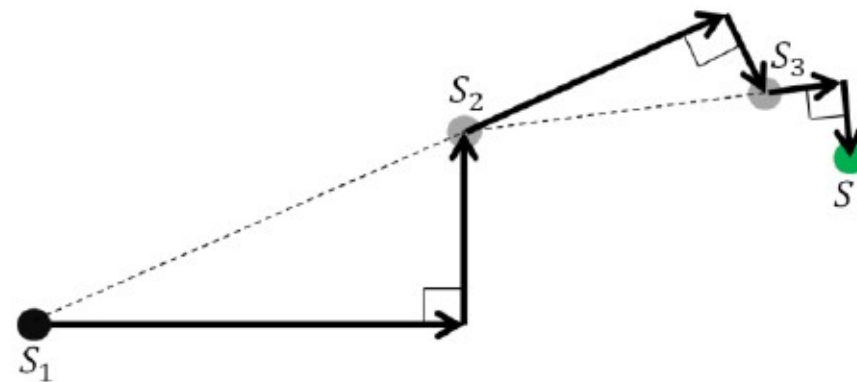
$$q_{i+1} = q_i + \dot{q}_i \cdot d_{sim}$$



Poszukiwanie minimum

Metoda Powella jest iteracyjną bezgradientową metodą poszukiwania minimum funkcji wielowymiarowej.

Jako metody poszukiwania na prostej, użyto złotego podziału. Jest ona prosta w implementacji oraz szybka w wykonaniu.



Sterowanie przy pomocy funkcji kary

Użycie funkcji kary pozwala na zwiększenie liczby źródeł błędu. Oznacza to, że podczas poszukiwania minimum możliwym jest jednoczesne podążanie za trajektorią oraz omijanie przeszkód.

$$f_p(t) = f_q(x(t)) + f_R(x(t)) + f_o(x(t))$$

f_p Funkcja kary

f_R Kara od ograniczeń robota

f_q Kara od położenia

f_o Kara od przeszkód

Kary od położenia końcowego i ograniczeń robota

Odległość pomiędzy położeniem końcówki robota w danym ruchu oraz docelowym położeniem w kroku.

$$f_{\bar{x}} = \Theta \left(\bar{x}(t_n) - x(t_n) \right)^2$$

Kara która jest nakładana na regulator za przekroczenie ograniczeń konstrukcyjnych robota. Przykład dla maksymalnej dopuszczalnej zmiany położenia w jednym kroku.

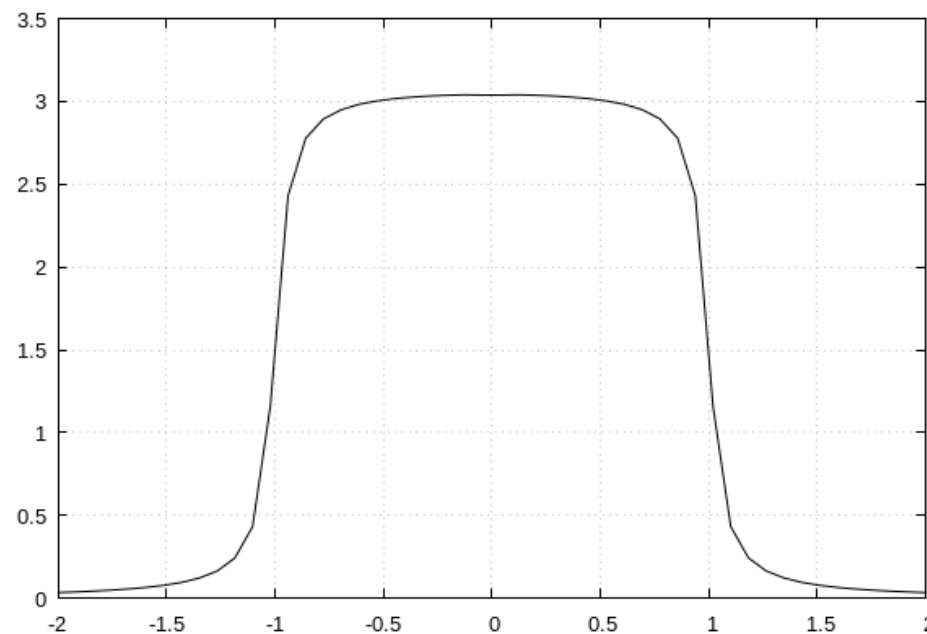
$$f_R = \xi \left(\frac{x(t_n) - x(t_{n-1})}{\rho_v} \right)^4$$

Kara od przeszkody

Kara za zbytne zbliżenie się do przeszkody o zadanej wielkości.

Przykład jednowymiarowy.

$$f_o^1(x(t_n)) = \arctan\left(\xi_n\left(x(t_n) - (\bar{x}_o - \bar{r}_o)\right)\right) + \arctan\left(\xi_n\left(x(t_n) - (\bar{x}_o + \bar{r}_o)\right)\right)$$



Kara dla wielowymiarowej przeszkody

Przykład jednowymiarowy można z łatwością rozszerzyć do N wymiarów:

$$f_o^N = \prod_{i=1}^N f_o^i(x(t))$$

Kara dla wielu przeszkód

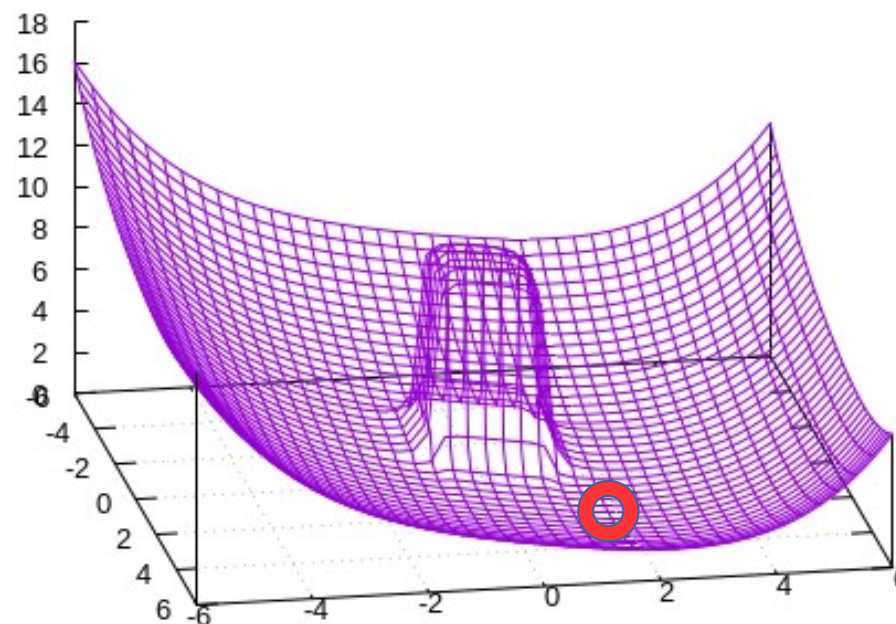
Dla M przeszkód w N wymiarowej przestrzeni funkcja kary przyjmuje postać:

$$f_o = \sum_{i=1}^M f_o^N(x(t))$$

Funkcja kary w przestrzeni 2D

Dla przestrzeni dwu-wymiarowej możliwym jest narysowanie wykresu funkcji kary.

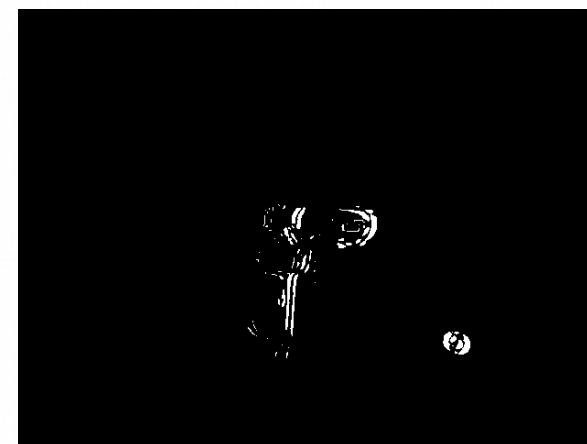
Funkcja kary z jedną przeszkodą w zasięgu ruchu manipulatora.



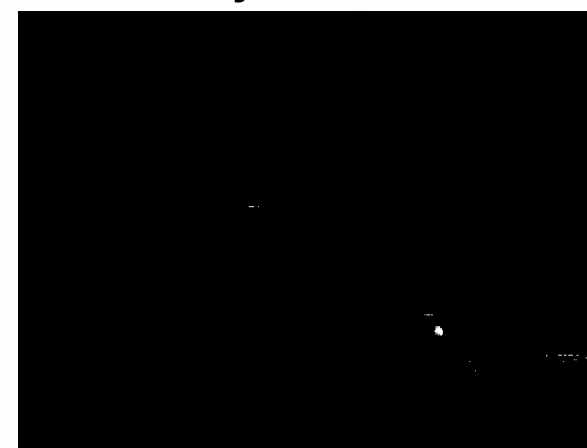
Detekcja przeszkód



Wykryta przeszkoda



Detekcja ruchu



Detekcja koloru

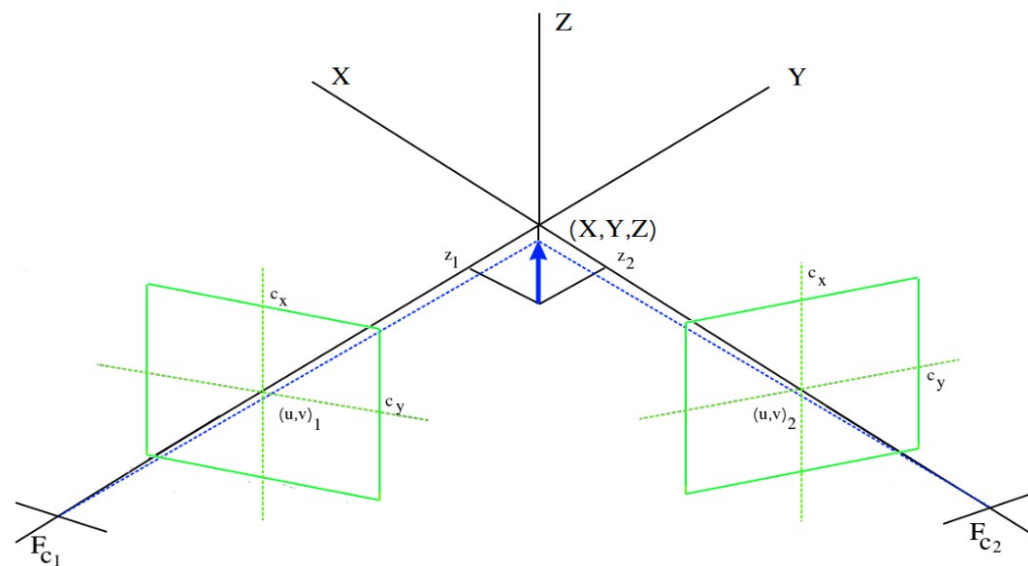
Obliczanie pozycji 3D

Wyznaczenie położenia przy pomocy prostego przekształcenia geometrycznego

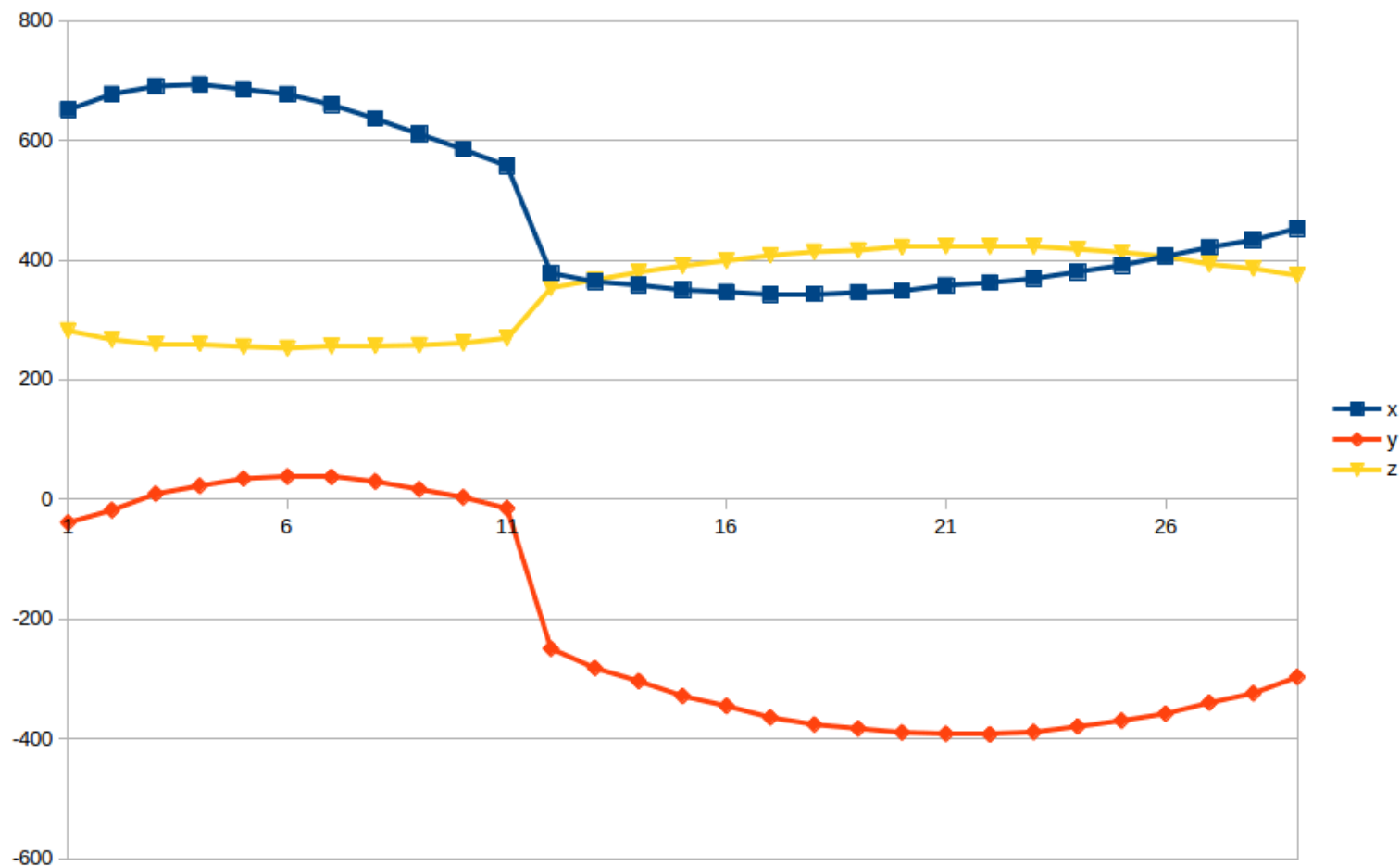
$$X_1 = F_{C_1} + \vec{a}_1 \frac{-(\vec{a}_1 \cdot \vec{a}_2)(\vec{a}_2 \cdot \vec{f}) + (\vec{a}_1 \cdot \vec{f})(\vec{a}_2 \cdot \vec{a}_2)}{(\vec{a}_1 \cdot \vec{a}_1)(\vec{a}_2 \cdot \vec{a}_2) - (\vec{a}_1 \cdot \vec{a}_2)(\vec{a}_1 \cdot \vec{a}_2)}$$

$$X_2 = F_{C_2} + \vec{a}_2 \frac{-(\vec{a}_1 \cdot \vec{a}_2)(\vec{a}_1 \cdot \vec{f}) + (\vec{a}_2 \cdot \vec{f})(\vec{a}_1 \cdot \vec{a}_1)}{(\vec{a}_1 \cdot \vec{a}_1)(\vec{a}_2 \cdot \vec{a}_2) - (\vec{a}_1 \cdot \vec{a}_2)(\vec{a}_1 \cdot \vec{a}_2)}$$

$$X_{est} = \frac{X_1 + X_2}{2}$$



Pomiary rzeczywiste z dwóch kamer

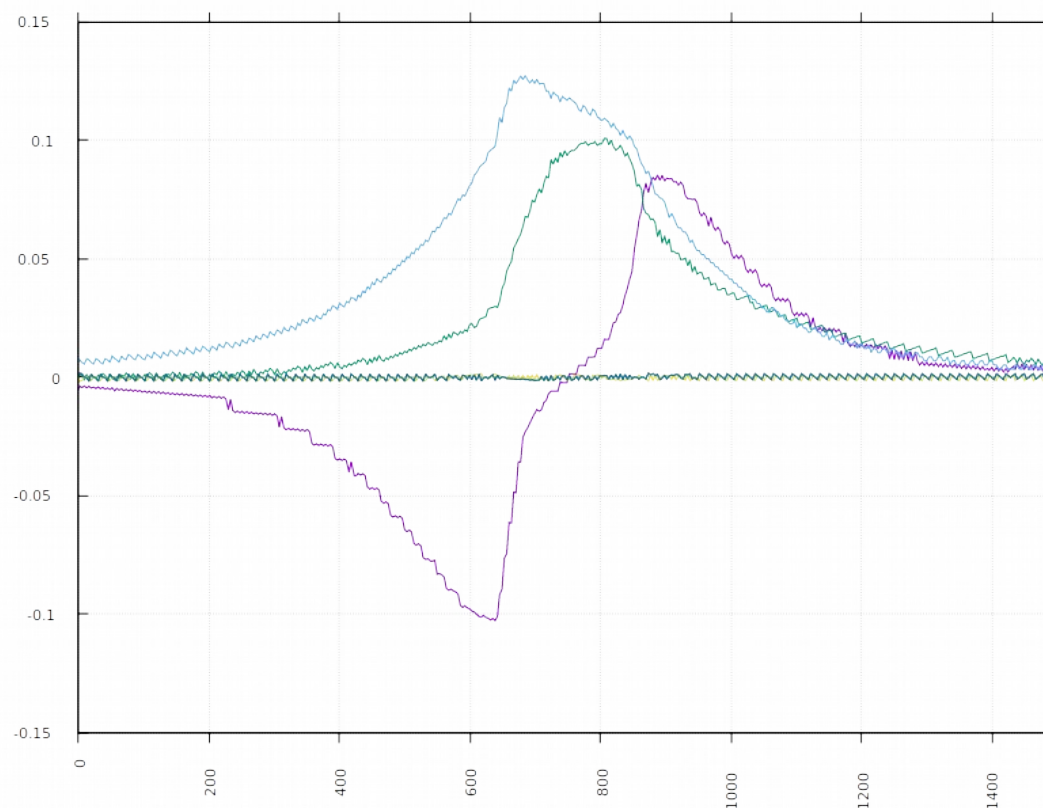


Błąd podczas manewru omijania

Funkcja błędu lepiej obrazuje faktyczne zachowanie robota.

W pierwszej fazie robot zwalniał aby nie uderzyć w przeszkodę.

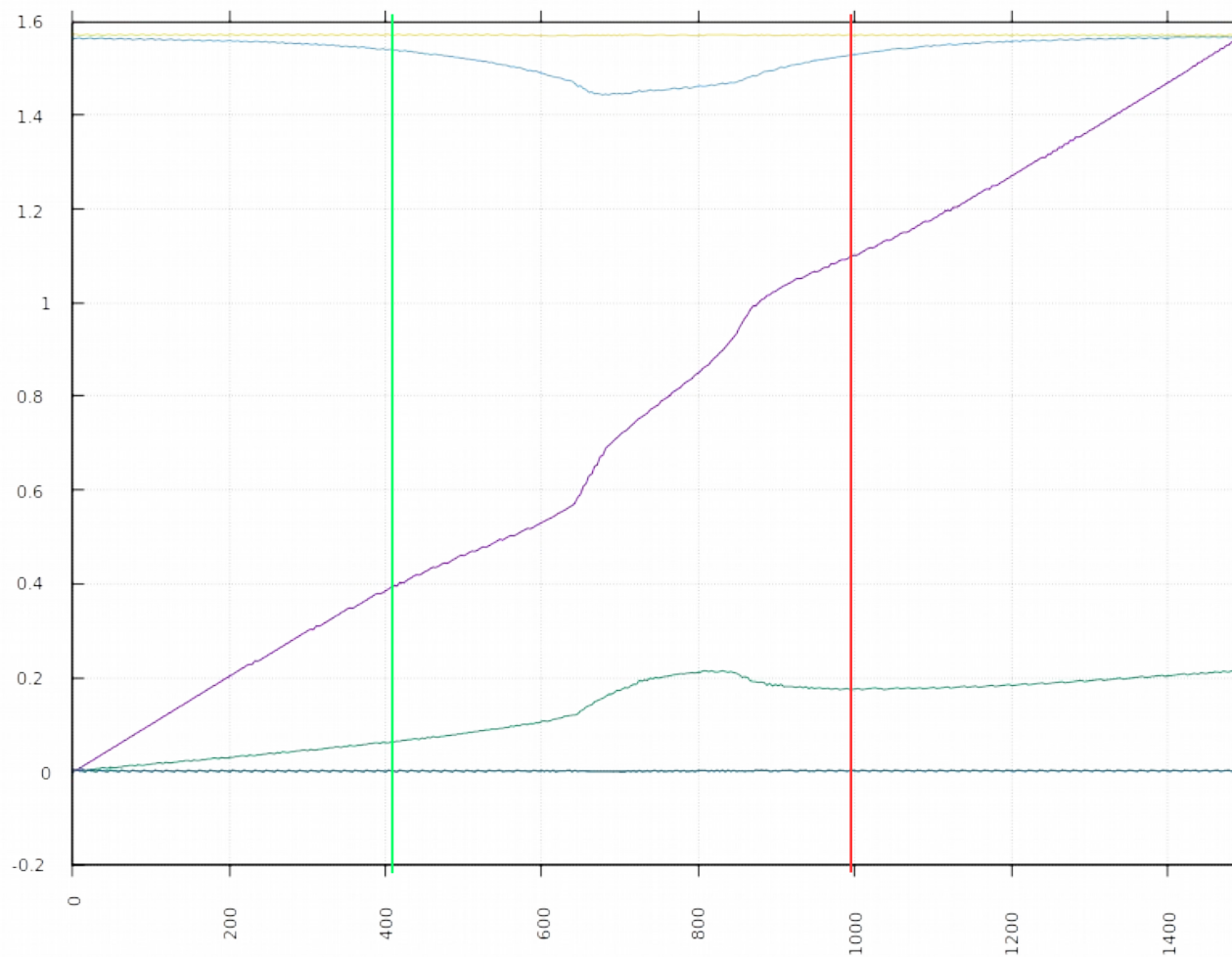
Gdy przeszkoda została ominięta robot, znów mógł przyspieszyć.



Manewr omijania

Wynik eksperymentu rzeczywistego omijania przeszkody znajdującej się na torze ruchu.

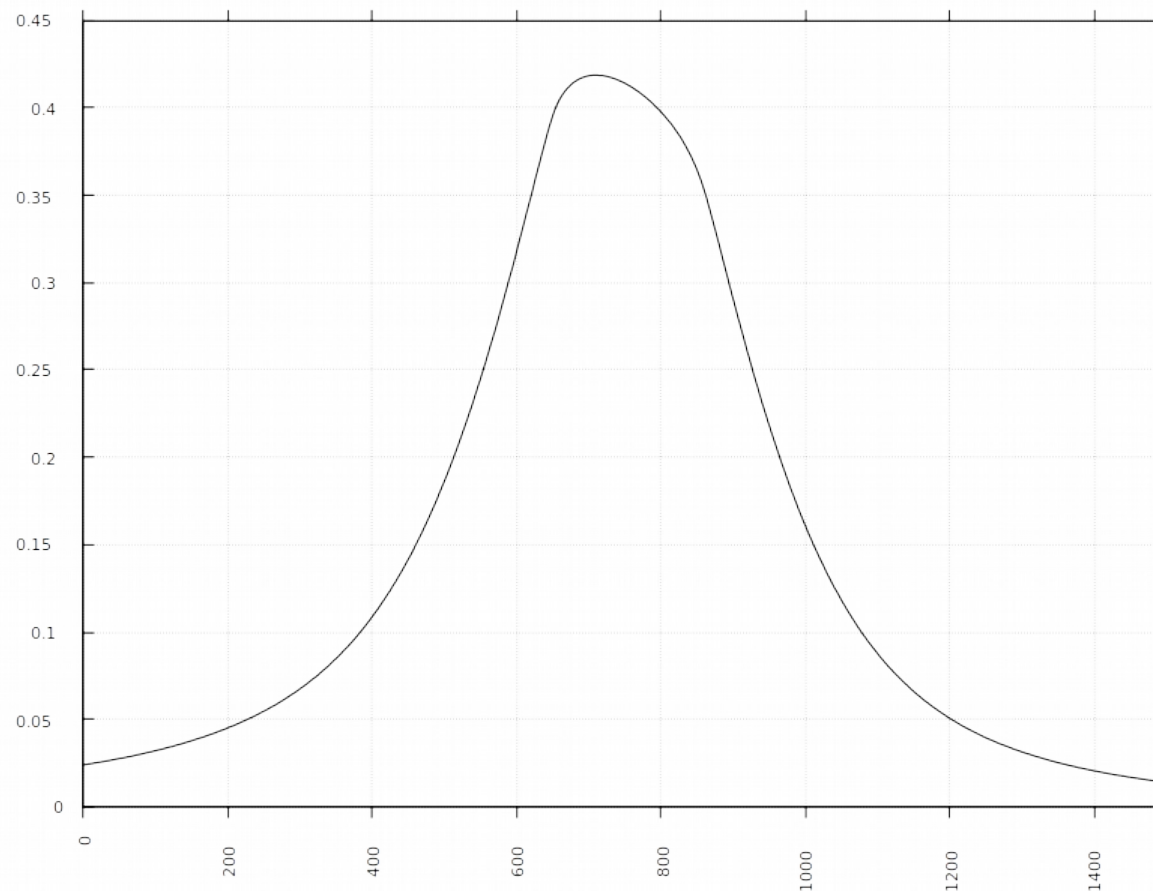
Regulator omija przeszkodę bez interwencji operatora.



Kara w trakcie trwania ruchu

Funkcja kary osiąga maksimum w chwili mijania przeszkody.

Po manewrze ominięcia funkcja kary maleje.



Podsumowanie

Oryginalnym wkładem autora jest:

- implementacja złożonej procedury polegającej na przejściu od planowanej trajektorii, z zastosowaniem punktów kontrolnych, przez równania kinematyki odwrotnej do przestrzeni przegubowej robota, a następnie konstrukcji trajektorii interpolacyjnej przy użyciu krzywych Beziera, co pozwala na jej użycie w funkcji kary,
- eksperymentalna generacja sterowań, prowadzona przy użyciu optymalizacji on-line, dla unikania przeszkód na ścieżce ruchu robota oraz dla uwzględnienia ograniczeń pola ruchu geometrią konstrukcji robota.