

Porównanie planerów ruchu OMPL, STOMP i CHOMP w planowaniu ruchu manipulatora UR5e w MoveIt

Repozytorium Github

Link do repozytorium

1 Cel projektu

Celem projektu było porównanie planerów ruchu OMPL, STOMP oraz CHOMP w planowaniu ruchu manipulatora UR5e w MoveIt.

2 Kryteria oceny

Planery porównano pod kątem:

- czasu planowania ścieżki,
- skuteczności planowania,
- długości ścieżki,
- zapotrzebowania na zasoby CPU oraz GPU.

2.1 Czas planowania ścieżki

Czas planowania ścieżki liczony jest automatycznie w MoveIt w przypadku znalezienia ścieżki. Wartość można odczytać poprzez `<plan>.planning_time`.

2.2 Skuteczność planowania

Skuteczność definiowana jest jako liczba udanych planowań podzielona przez liczbę wszystkich prób.

2.3 Zapotrzebowanie na zasoby CPU i GPU

Określono jako procentowe zużycie zasobów komputera podczas zadania planowania trajektorii.

2.4 Długość ścieżki

Długość ścieżki końcówki manipulatora obliczano na podstawie odległości euklidesowej:

$$\sum_{i=0}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad (1)$$

Natomiast długość ścieżki w przestrzeni złączy na podstawie średniej zmiany pozycji:

$$\sum_{i=0}^n \text{mean}(|\text{position}_i - \text{position}_{i-1}|) \quad (2)$$

gdzie n to liczba punktów trajektorii.

3 Wykorzystane planery

3.1 OMPL (Open Motion Planning Library)

Biblioteka skupiona na planowaniu trajektorii w przestrzeni konfiguracyjnej robota. Oparta na losowym próbkowaniu i budowie grafów dróg.

3.2 STOMP (Stochastic Trajectory Optimization for Motion Planning)

Optymalizacyjny planer, który nie buduje grafów ani nie próbuje próbować przestrzeni. Zaczyna od wstępnej trajektorii (np. liniowej) i stochastycznie ją modyfikuje, minimalizując koszty związane z kolizjami, długością i płynnością trajektorii.

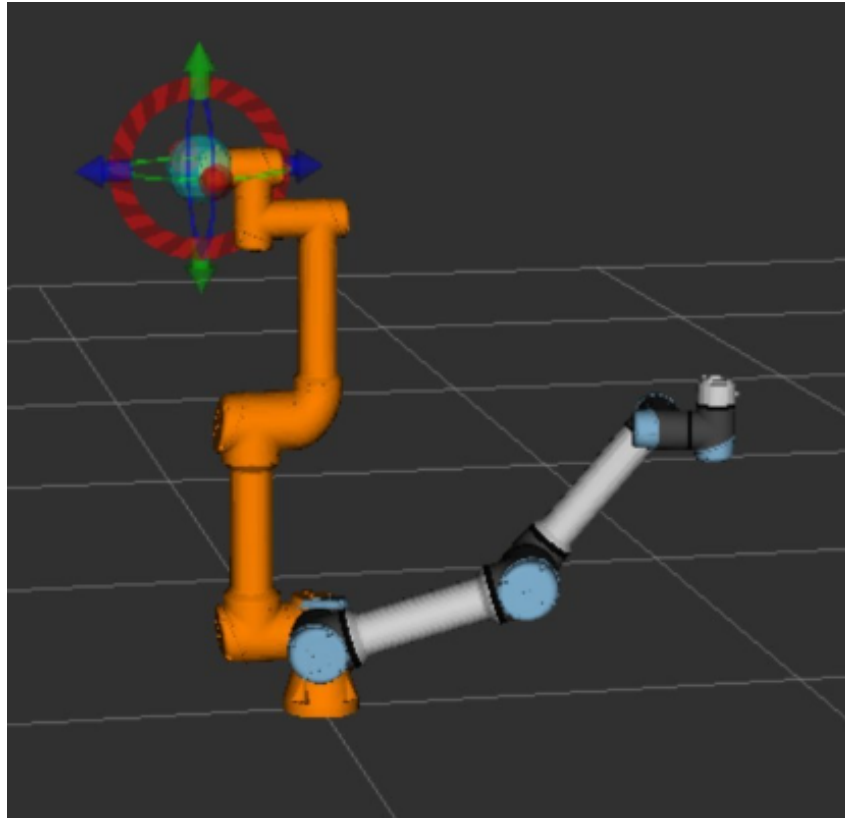
3.3 CHOMP (Covariant Hamiltonian Optimization for Motion Planning)

Deterministyczna metoda optymalizacji trajektorii. Rozpoczyna od wstępnej trajektorii i optymalizuje ją pod kątem płynności oraz unikania kolizji, minimalizując funkcję kosztu.

4 Porównanie

W celu wiarygodnego porównania, pozycja początkowa i końcowa robota, rodzaj robota oraz pozycja i parametry przeszkód były identyczne dla wszystkich testów. Każde planowanie uruchomiono 100 razy. W tabelach przedstawiono wartości minimalne, maksymalne oraz średnie.

4.1 Planowanie bez przeszkód



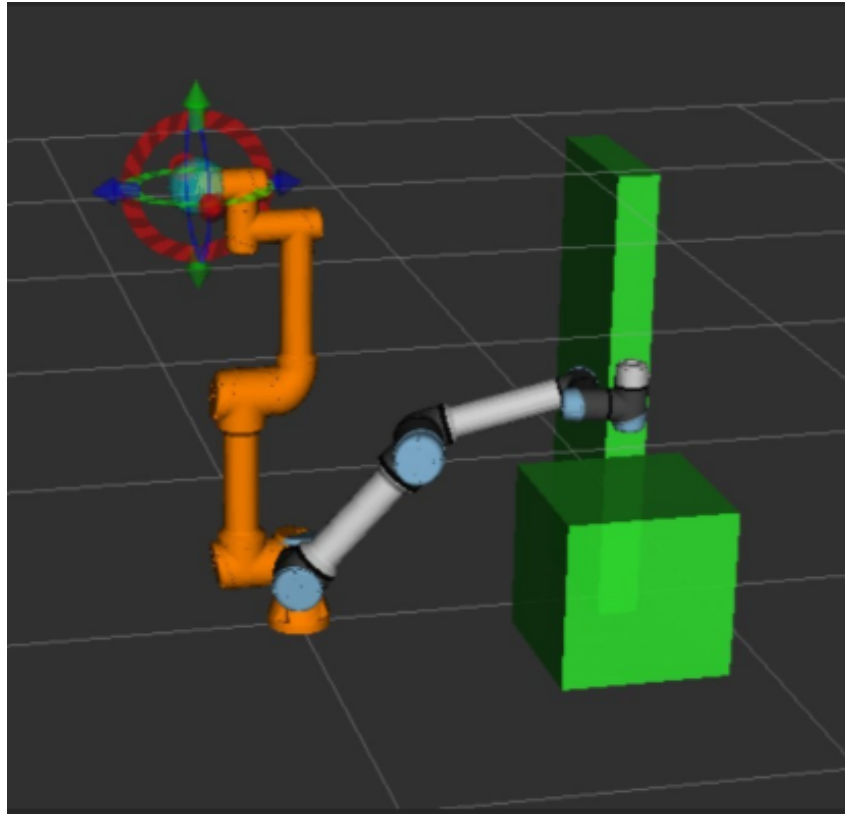
Rysunek 1: Przykładowa trajektoria planowania bez przeszkód

Tabela 1: Wyniki planowania bez przeszkód

Parametr	OMPL	STOMP	CHOMP
Czas planowania [s] min	0.0127	0.0129	0.0134
Czas planowania [s] max	0.0337	0.0455	0.0327
Czas planowania [s] mean	0.0218	0.0232	0.0234
Ilość punktów trajektorii min	58	58	77
Ilość punktów trajektorii max	285	293	266
Ilość punktów trajektorii mean	175.16	183.36	177.96
Droga końcówki [m] min	0.8623	0.8560	0.5685
Droga końcówki [m] max	2.6641	2.2626	2.2849
Droga końcówki [m] mean	1.1602	1.1238	1.0955
Droga złączy [rad] min	0.7857	0.7856	1.5058
Droga złączy [rad] max	3.9243	4.3639	3.9566
Droga złączy [rad] mean	2.1460	2.2667	2.6219
Zużycie CPU [%] min	2.93	1.20	1.03
Zużycie CPU [%] max	9.81	6.28	6.19
Zużycie CPU [%] mean	6.49	3.82	3.94
Skuteczność [%]	45	59	47

W planowaniu bez przeszkód OMPL charakteryzował się najkrótszym czasem planowania i najkrótszymi drogami złączy. STOMP uzyskał najwyższą skuteczność. CHOMP generował najkrótsze trajektorie końcówki i najmniejsze zużycie CPU, jednak przy niższej skuteczności niż STOMP. Planery te z paroma wyjątkami, jak na przykład wysokie w porównaniu do innych zużycie CPU w planerze OMPL były sobie mniej więcej statystycznie równe.

4.2 Planowanie z przeszkodami wariant 1



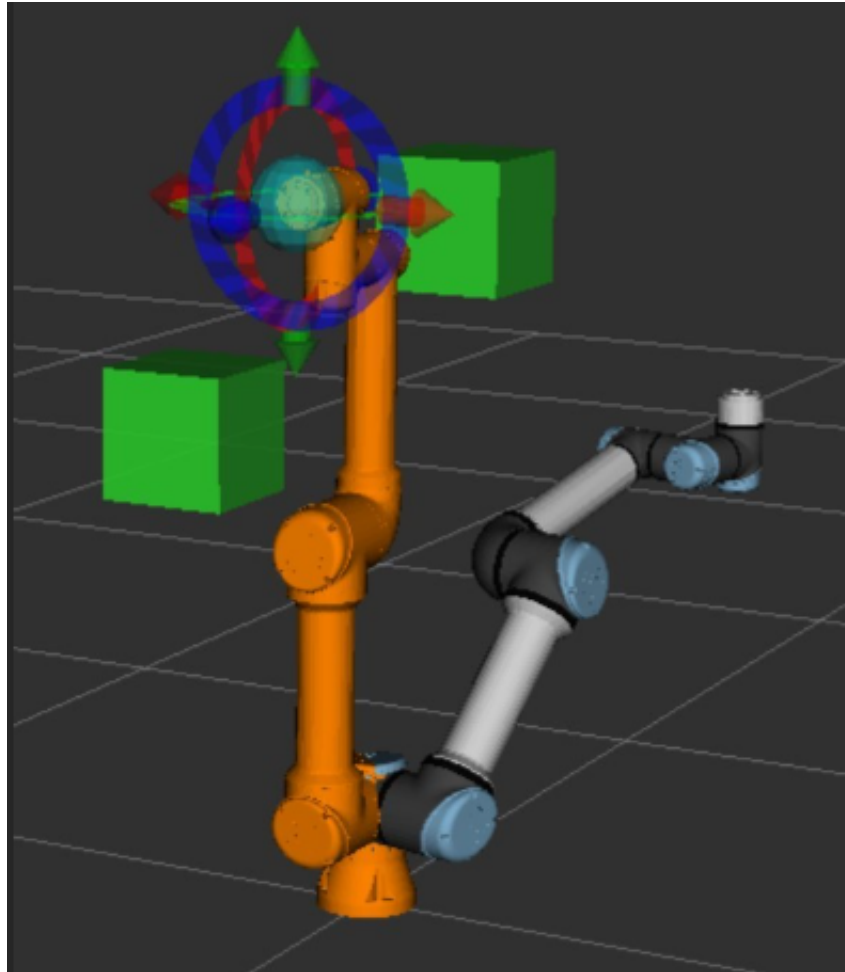
Rysunek 2: Przykładowa trajektoria planowania z przeszkodami wariant 1

Tabela 2: Wyniki planowania z przeszkodami wersja 1

Parametr	OMPL	STOMP	CHOMP
Czas planowania [s] min	0.0127	0.0145	0.0124
Czas planowania [s] max	0.0750	0.0469	0.1133
Czas planowania [s] mean	0.0255	0.0252	0.0253
Ilość punktów trajektorii min	58	58	77
Ilość punktów trajektorii max	303	392	407
Ilość punktów trajektorii mean	200.86	193.08	175.96
Droga końcówki [m] min	0.8560	0.8560	0.5685
Droga końcówki [m] max	2.1679	2.0774	2.2917
Droga końcówki [m] mean	1.1804	1.1482	1.0359
Droga złączy [rad] min	0.7855	0.7856	0.9830
Droga złączy [rad] max	4.1094	4.5820	4.6163
Droga złączy [rad] mean	2.4114	2.3184	2.5630
Zużycie CPU [%] min	2.02	1.20	0.87
Zużycie CPU [%] max	18.09	16.78	6.67
Zużycie CPU [%] mean	14.62	12.86	3.92
Skuteczność [%]	50	50	52

W wariancie z przeszkodami (wersja 1) CHOMP uzyskał najwyższą skuteczność i najkrótszą średnią długość trajektorii końcówki przy najniższym zużyciu CPU. STOMP charakteryzował się najmniejszą średnią drogą w przestrzeni złączy i najkrótszym średnim czasem planowania. OMPL oferował stabilny i krótki czas planowania i najkrótsze drogi złączy, przy wyższym zużyciu CPU.

4.3 Planowanie z przeszkodami wariant 2



Rysunek 3: Przykładowa trajektoria planowania z przeszkodami wariant 2

Tabela 3: Wyniki planowania z przeszkodami wersja 2

Parametr	OMPL	STOMP	CHOMP
Czas planowania [s] min	0.0139	0.0136	0.0154
Czas planowania [s] max	0.0530	0.0469	0.0714
Czas planowania [s] mean	0.0286	0.0300	0.0293
Ilość punktów trajektorii min	58	58	93
Ilość punktów trajektorii max	303	392	654
Ilość punktów trajektorii mean	209.95	221.78	217.3256
Droga końcówki [m] min	0.8560	0.8560	0.7528
Droga końcówki [m] max	3.1545	2.0774	3.5736
Droga końcówki [m] mean	1.3729	1.6208	1.5229
Droga złączy [rad] min	0.7854	0.7856	1.9199
Droga złączy [rad] max	4.1094	4.5820	8.6651
Droga złączy [rad] mean	2.6569	2.8133	3.0800
Zużycie CPU [%] min	2.02	1.20	0.87
Zużycie CPU [%] max	19.57	16.78	6.67
Zużycie CPU [%] mean	16.25	12.86	3.92
Skuteczność [%]	44	49	43

W wariancie z przeszkodami (wersja 2) najwyższą skuteczność osiągnął algorytm STOMP. Najniższe zużycie CPU odnotowano dla CHOMP. Najkrótszą drogę złączy, najmniejszą liczbę punktów trajektorii oraz najniższy średni czas planowania uzyskano przy wykorzystaniu OMPL.

5 Podsumowanie

Nie da się jednoznacznie wskazać najlepszego planera. Wybór algorytmu zależy od konkretnego zadania, charakterystyki środowiska oraz oczekiwań względem trajektorii i parametrów planowania.

Planer **CHOMP** charakteryzował się generowaniem najkrótszych średnich trajektorii w przestrzeni roboczej oraz w przestrzeni złączy. Uzyskiwał również najniższe średnie zużycie CPU, a w pierwszym wariancie środowiska z przeszkodami osiągnął najwyższą skuteczność. W środowisku z przeszkodami (wersja 2) CHOMP generował stabilne trajektorie przy minimalnym wykorzystaniu zasobów, choć skuteczność była nieco niższa względem STOMP.

Planer **STOMP** w środowisku bez przeszkód uzyskał najwyższą skuteczność. W obecności przeszkód również osiągał wysoką stabilność, zachowując kompromis między długością trajektorii złączy i końcówki. Cechował się także umiarkowanym zużyciem CPU oraz krótkimi czasami planowania.

Planer **OMPL** wyróżniał się średnio najkrótszym czasem planowania w większości testów, zwłaszcza w środowisku bez przeszkód. Generował także najkrótsze trajektorie w przestrzeni złączy. Jego skuteczność była jednak nieco niższa w środowiskach z przeszkodami, a zużycie CPU — wyższe w porównaniu do pozostałych algorytmów.

Podsumowując, OMPL oferuje najlepsze czasy planowania, STOMP zapewnia wysoką skuteczność i równowagę parametrów, natomiast CHOMP minimalizuje długość trajektorii i zużycie zasobów. Dobór planera powinien być uzależniony od priorytetów konkretnego

zastosowania.