

Ćwiczenie nr 520:

Metody interpolacyjne planowania ruchu manipulatorów

1 Wstęp

Celem ćwiczenia jest zapoznanie studentów z metodami interpolacyjnymi planowania ruchu manipulatorów. Realizacja ćwiczenia polega na:

- zapoznaniu się z teorią metod (wydruki ekranów pomocy znajdują się w dodatku, a szczegóły teoretyczne w, dostępnej w laboratorium i bibliotece, literaturze),
- ustalenie metod badanych i ich przetestowanie,
- opracowanie wyników w formie sprawozdania (najlepiej: część podczas laboratorium, część po laboratorium)

Wykorzystywane mogą być metody:

- (1) dwie metody planowania toru ruchu wielomianami z kontrolowanymi własnościami (ciągłość, dopuszczalne skoki o przewidywalnej amplitudzie) przejść między segmentami
- (2) metoda Taylora (i jej modyfikacja) planowania toru ruchu z kontrolowanym odchyleniem od zadanej ścieżki prostoliniowej w przestrzeni zadaniowej
- (3) łączenie punktów węzłowych (w przestrzeni konfiguracyjnej) wielomianem Bernsteina-Beziera

2 Pytania do zastanowienia

Dla właściwego przygotowania (przeprowadzenia) ćwiczenia ważne jest odpowiedź na wszystkie (lub też część) pytań zamieszczonych poniżej. Niektóre z nich mają charakter ogólny (niezależne od metody interpolacyjnej), niektóre uwzględniają specyfikę metody planowania.

Pytania ogólne:

- co to jest interpolacja?
- czym różni się od aproksymacji, ekstrapolacji?
- co to jest przestrzeń konfiguracyjna, zadaniowa?
- w jakiej przestrzeni najłatwiej sterować, a w jakiej definiować zadanie (np. przeszkody), dlaczego?
- czym jest punkt w tych przestrzeniach?
- co to jest trajektoria, tor, ścieżka (efektora)?
- jaki tor ruchu jest łatwy do wykonania przez manipulator, a jaki nie?
- $SO(3)$, $SE(3)$ - własności,
- $SO(3)$, $SE(3)$ - do czego w robotyce wykorzystywane?
- parametryzacje $SO(3)$, $SE(3)$
- dlaczego zwykle nie wykorzystujemy bezpośrednio $SO(3)$, $SE(3)$, tylko ich parametryzacje?
- jakiego typu występują kłopoty (pomoc: zastanowić się nad wzajemną jednoznacznością odwzorowania punkt w parametryzacji - punkt w $SO/E(3)$)
- jakimi funkcjami można interpolować tor (zalety, wady)?
- ile parametrów powinny mieć funkcje interpolujące?

- czy lepiej interpolować wiele punktów jedną funkcją, czy kleić kawałkami (zalety, wady: np. złożoność obliczeniowa, łatwość modyfikacji gdy zmieniają się punkty węzłowe)?

Pytania dotyczące metody sklejaných wielomianów:

- jaki stopień powinien mieć wielomian by zapewnić tor o zadanych własnościach na brzegu (położenie, prędkość, przyspieszenie)
- ile ciągłych pochodnych toru jest zwykle potrzebnych w zadaniach robotycznych i dlaczego?
- jak kształtuje się błąd w śledzeniu ścieżki prostoliniowej dla metod interpolacyjnych? Jakże może to mieć skutki w środowisku kolizyjnym?

Pytania do metody Taylora:

- co to jest prosta w przestrzeni zadaniowej?
- jak wygląda prosta w różnych parametryzacjach $SE(3)$
- jakiego typu są współrzędne (czy są współmierne?)
- jak maksymalny błąd wzdłuż ścieżki zależy od jej długości? gdzie ten błąd najczęściej występuje? Czy jest pewna regularność?
- a gdy długość ścieżki jest ustalona, to jak błąd zależy od lokalizacji ścieżki w przestrzeni zadaniowej?

3 Przydatne komendy Linuxa:

`interpol.JF &` - wywołanie programu (nie zapomnieć o uruchomieniu w tle (znak ampersanda)), pozwala wykorzystywać konsolę podczas działania programu
`ls` - prezentuje na konsoli zawartość katalogu bieżącego
`rm xxx.a` - kasuje plik o nazwie `xxx.a`
`rm *.dat` - kasuje wszystkie pliki z rozszerzeniem `dat`
`dir` - prezentacja zawartości dyskiety znajdującej się w napędzie
`cp *.dat` - kopiowanie plików z rozszerzeniem `dat` na dyskietkę

Uwaga: Z nieznaných (mi) przyczyn zbyt wiele okien wykorzystywanych podczas wizualizacyjnej części ćwiczenia powoduje uruchomienie wielu procesów w wyniku czego następuje zwolnienie pracy komputera. Dlatego należy: 1) kasować okna graficzne po ich wykorzystaniu; 2) po kilkudziesięciu (np. 20) minutach użytkowania, opuścić program i powtórnie do niego wejść.

Literatura:

- (1) I. Dulęba, "Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych", Akademicka Oficyna Wyd. EXIT, W-wa, 2001.

Implementatorką programów jest mgr Justyna Foremniak (absolwentka AiR 2004)

Instrukcję opracował: dr hab. Ignacy Dulęba, prof. PWr

Wersja z dnia 20.02.2006

Wszystkie uwagi dotyczące (poprawy) działania programów wykorzystywanych w tym ćwiczeniu kierować na adres: `ignacy.duleba@pwr.wroc.pl` (ewentualnie dokładnie opisując dane wykorzystywane w testach)

4 Dodatek

W podrozdziale znajdują się wydruki ekranów pomocy dla wszystkich metod.

Interpolacja wielomianami stopnia trzeciego z ciągłym przyspieszeniem

Gwarantuje ciągłość pochodnej toru drugiego rzędu w przestrzeni konfiguracyjnej.

Wyznacza tor będący sklejeniem wielomianów stopnia trzeciego o ciągłych pochodnych do rzędu drugiego w N – konfiguracjach węzłowych.

Użytkownik w programie podaje:

- długość ramion robota L_1 i L_2 (1 – 2000 m)
- konfiguracje początkową $q_1(x, y)^T$ oraz końcową $q_N(x, y)^T$ (konfiguracje podajemy w stopniach)
- prędkości na końcach przedziału $\dot{q}_1(x, y)^T$ i $\dot{q}_N(x, y)^T$
- przyspieszenia na końcach przedziału $\ddot{q}_1(x, y)^T$ i $\ddot{q}_N(x, y)^T$
- czas realizacji toru T (z przedziału 1-20 s)
- błąd śledzenia trajektorii (przedział zależy od długości ramion robota)

W wyniku symulacji można wygenerować (do pliku *.ps lub na ekran) wykresy:

- położenia
- prędkości
- przyspieszenia
- odległości od ścieżki prostoliniowej w przestrzeni zadaniowej
- wynikową ścieżkę efektora

Interpolacja wielomianami stopnia trzeciego z ciągłą prędkością

Gwarantuje ciągłość pochodnej toru pierwszego rzędu w przestrzeni konfiguracyjnej.

Trzy algorytmy różniące się sposobem wyznaczenia prędkości w węzłowych pośrednich:

$$\dot{q}_i = \begin{cases} \frac{-q_{i+2} + 6q_{i+1} - 3q_i - 2q_{i-1}}{6h} & X1 \\ \frac{q_{i+1} + 4q_i - 5q_{i-1} - \dot{q}_{i-1}}{4h} - \frac{\dot{q}_{i-1}}{2} & X2 \\ \frac{-q_{i+2} + 9q_{i+1} + 9q_i - 17q_{i-1} - \dot{q}_{i-1}}{18h} - \frac{\dot{q}_{i-1}}{3} & X3 \end{cases} \quad i=2, \dots, N-2$$

$$\dot{q}_{N-1} = \begin{cases} \frac{2q_N + 3q_{N-1} - 6q_{N-2} + q_{N-3}}{6h} & X1 \\ \frac{q_N + 4q_{N-1} - 5q_{N-2} - \dot{q}_{N-2}}{4h} - \frac{\dot{q}_{N-2}}{2} & X2 \\ \frac{17q_N - 9q_{N-1} - 9q_{N-2} + q_{N-3} - \dot{q}_N}{18h} - \frac{\dot{q}_N}{3} & X3 \end{cases}$$

Wyznacza tor będący sklejeniem wielomianów stopnia trzeciego o ciągłych pochodnych do pierwszego rzędu.

Użytkownik w programie podaje:

- długość ramion robota L1 i L2 (1 – 2000 m)
- konfiguracje początkową $q_1(x, y)^T$ oraz końcową $q_N(x, y)^T$ (konfiguracje podajemy w stopniach)
- prędkości na końcach przedziału $\dot{q}_1(x, y)^T$ i $\dot{q}_N(x, y)^T$
- przyspieszenia na końcach przedziału $\ddot{q}_1(x, y)^T$ i $\ddot{q}_N(x, y)^T$
- czas realizacji toru T (z przedziału 1-20 s)
- błąd śledzenia trajektorii (przedział zależy od długości ramion robota)

W wyniku symulacji można wygenerować (do pliku *.ps lub na ekran) wykresy:

- położenia
- prędkości
- przyspieszenia
- odległości od ścieżki prostoliniowej w przestrzeni zadaniowej
- wynikową ścieżkę efektora

Metoda kontrolowanego odchylenia od ścieżki prostoliniowej. (algorytm Taylora)

Polega na takim wyznaczeniu ścieżki wynikowej w przestrzeni zadaniowej aby nie odbiegała ona od ścieżki zadanej na odległość większą niż założoną przez użytkownika (błąd śledzenia).

Jest to typowy algorytm działający na zasadzie „dziel i rządź”.

Użytkownik w programie podaje:

- długość ramion robota L1 i L2 (1 – 2000 m)
- konfiguracje początkową $q_1(x, y)^T$ oraz końcową $q_N(x, y)^T$, z których wyliczane są punkty brzegowe $x(0)$ i $x(N)$. (konfiguracje podajemy w stopniach)
- czas realizacji toru T (z przedziału 1-20 s)
- błąd śledzenia trajektorii (przedział zależy od długości ramion robota)

W wyniku symulacji można wygenerować (do pliku *.ps lub na ekran) wykresy:

- *położenia*
- *prędkości*
- *odległości od ścieżki prostoliniowej w przestrzeni zadaniowej*
- *wynikową ścieżkę efektora*

Zmodyfikowana metoda kontrolowanego odchylenia od ścieżki prostoliniowej. (zmodyfikowany algorytm Taylora)

Polega na takim sposobie dodawania punktów węzłowych, by błąd śledzenia osiągnął maksymalną dopuszczalną wartość.

Od zwykłego algorytmu Taylora różni się właśnie mniejszą liczbą punktów węzłowych, a co za tym idzie mniejszą liczbę skoków prędkości na wynikowym torze.

Użytkownik w programie podaje:

- długość ramion robota L_1 i L_2 (1 – 2000 m)
- konfiguracje początkową $q_1(x, y)^T$ oraz końcową $q_N(x, y)^T$, z których wyliczane są punkty brzegowe $x(0)$ i $x(N)$. (konfiguracje podajemy w stopniach)
- czas realizacji toru T (z przedziału 1-20 s)
- błąd śledzenia trajektorii (przedział zależy od długości ramion robota)

*W wyniku symulacji można wygenerować (do pliku *.ps lub na ekran) wykresy:*

- *położenia*
- *prędkości*
- *odległości od ścieżki prostoliniowej w przestrzeni zadaniowej*
- *wynikową ścieżkę efektora*

Zmodyfikowana metoda kontrolowanego odchylenia od ścieżki prostoliniowej. (zmodyfikowany algorytm Taylora)

Polega na takim sposobie dodawania punktów węzłowych, by błąd śledzenia Osiągnął maksymalną dopuszczalną wartość.

Od zwykłego algorytmu Taylora różni się właśnie mniejszą liczbą punktów węzłowych, a co za tym idzie mniejszą liczbę skoków prędkości na wynikowym torze.

Użytkownik w programie podaje:

- długość ramion robota L_1 i L_2 (1 – 2000 m)
- konfiguracje początkową $q_1(x, y)^T$ oraz końcową $q_N(x, y)^T$, z których wyliczane są punkty brzegowe $x(0)$ i $x(N)$. (konfiguracje podajemy w stopniach)
- czas realizacji toru T (z przedziału 1-20 s)
- błąd śledzenia trajektorii (przedział zależy od długości ramion robota)

*W wyniku symulacji można wygenerować (do pliku *.ps lub na ekran) wykresy:*

- *położenia*
- *prędkości*
- *odległości od ścieżki prostoliniowej w przestrzeni zadaniowej*
- *wynikową ścieżkę efektora*