POLISH ACADEMY OF SCIENCES - COMMITTEE OF MATERIALS SCIENCE
SILESIAN UNIVERSITY OF TECHNOLOGY OF GLIWICE
INSTITUTE OF ENGINEERING MATERIALS AND BIOMATERIALS
ASSOCIATION OF ALUMNI OF SILESIAN UNIVERSITY OF TECHNOLOGY
Conference Proceedings
12th INTERNATIONAL SCIENTIFIC CONFERENCE
ACHIEVEMENTS IN MECHANICAL & MATERIALS ENGINEERING

Zastosowanie algorytmu Q-learning do programowania robotów przemysłowych

G.G. Kost

Katedra Automatyzacji Procesów Technologicznych i Zintegrowanych Systemów Wytwarzania, Politechnika Śląska w Gliwicach,
44-100 Gliwice, ul. Konarskiego 18a, Poland

Przedstawiono metodę automatycznego planowania trajektorii robota w oparciu o algorytm Q-learning. Do oceny poprawności uzyskiwanych rozwiązań wykorzystano ergodyczny proces decyzyjny Markowa. Omówiono warunki w jakich zachodzi proces definiowania trajektorii robota w systemie technologicznym i podstawy zastosowanej metody uczenia ze wzmocnieniem.

1. WPROWADZENIE

Programowanie robota przemysłowego jest jednym z najtrudniejszych w realizacji zadań w zautomatyzowanych systemach wytwarzania. Jest ono szczególnie trudne z logicznego punktu widzenia, gdyż jest zdeterminowane wymaganiami procesu technologicznego. Składa się z dwóch zadań: podstawowego jakim jest programowanie ruchu i pomocniczego obejmującego realizację warunków synchronizacji robota z procesem technologicznym. Rozwiązanie zadania planowania bezkolizyjnych trajektorii robota wydaje się zagadnieniem pierwszoplanowym, dlatego celowym wydaje się poszukiwania metod, które by ten proces mogły zintensyfikować.

2. TRAJEKTORIA ROBOTA

Trajektoria robota jest zdefiniowana miejscem docelowym jakie robot powinien osiągnąć i sposobem jakim powinien ją zrealizować [3,2]. Konieczność unikania kolizji wymusza realizację tego zadania metodą kolejnych kroków, tzw. kroków elementarnych. Oznacza to, że każda trajektoria robota jest „sumą" następujących po sobie, uporządkowanych kroków elementarnych t_i, po zrealizowaniu których robot osiąga zaplanowany stan s_i. Stany s_i osiągane są drogą realizacji przyjętego sposobu działania δ_i. Uwzględniając powyższe czynniki, możemy określić trajektorię robota jako:

$$\Gamma = \prod_{i=1}^{n} t_i$$

gdzie: $t_i = (s \wedge \delta)_i$, $\wedge$ - operator koniunkcji, n – liczba kroków pozwalających na ominięcie przeszkody.

Jak zatem widać, zadanie planowania trajektorii robota sprowadza się do wyboru odpowiedniego ciągu kroków elementarnych t_i, osiąganych w nich stanów elementarnych s_i oraz sposobów δ_i przy pomocy których może być to zrealizowane.

3. ZAŁOŻENIA

Przyjmijmy, że zadanie planowania trajektorii robota realizowane jest w środowisku, w którym poszczególne obiekty nie zmieniają swojego położenia. Zatem, potencjalną liczbę wszystkich stanów s_i jakie robot może osiągnąć i kroków t_i jakie może zrealizować możemy uznać za skończoną.

Krokiem elementarnym t_i trajektorii robota nazywać będziemy jego przemieszczenie wynikające ze zmiany współrzędnych ruchu wyznaczonych w przestrzeni zewnętrznej robota (przestrzeń zadania robota [2]). Krok elementarny wyrażony jest w układzie współrzędnych zewnętrznych robota miejscem docelowym wynikającym, które nazywać będziemy pozycją wspomagającą P_{wi}.

Akcją robota a_i nazywamy jego elementarne działanie, określające sposób zmiany parametrów geometrycznych ustalonych dla jednego kroku elementarnego t_i. Akcja jest zdefiniowana pewną funkcją opisującą sposób przejścia δ_i tak, by osiągnąć kolejny stan konfiguracji przestrzennej s_i na drodze t_i. Osiągnięcie stanu s_i *w* wyniku zrealizowania funkcji δ_i oznaczać będziemy jako $\delta(s_i)$. Zatem:

$$a_i = t_i \wedge \delta(s_i)$$

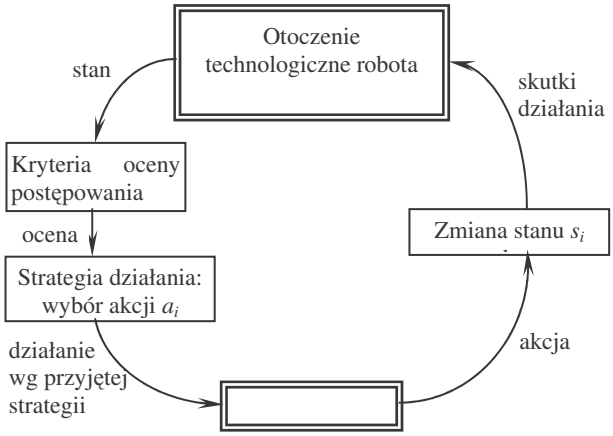
Zadaniem robota nazywać będziemy ciąg kolejnych akcji tworzący logiczną całość określony odpowiednimi parametrami początku i końca. Relacje te możemy to zapisać jako:

$$\Gamma_{opt} = \min_i \sum_{i=1}^{n} a_i = \min_i \sum_{i=1}^{n} \left[t \wedge \delta(s_i) \right]_i$$

przy spełnieniu warunku, że tak zdefiniowany ciąg akcji elementarnych został utworzony przy możliwie najmniejszej ich liczbie *n*. Warunek ten będziemy nazywać prostym kryterium optymalności.

4. ALGORYTM Q-leraning

W warunkach stosunkowo dużej różnorodności potencjalnie możliwych dróg robota, cenną umiejętnością byłoby wielokrotne wykonanie zadania „przez analogię do tego samego lub podobnego", zrealizowanego wcześniej i ocenionego jako dobre. Tak postawiony problem można identyfikować z procesem uczenia się. Podstawowym kryterium jakości uczenia się jest odpowiedni wybór metody. Z pośród kilku stosowanych metod maszynowego uczenia się [1,2] postanowiono zastosować algorytm Q-learning. Zakłada on, że działania „ucznia" odbywają się w dyskretnych przedziałach czasowych i są zdeterminowane jedynie bieżącymi warunkami występującymi w jego otoczeniu [1,2]. Oznacza to, że zrealizowanie każdego kolejnej akcji a_i robota może uwzględniać tylko warunki jakie „ujawniły" się po zrealizowaniu poprzedniego kroku, a wybór kolejnych akcji a_i wynika tylko z tego co dostępne jest w danej chwili i stanu w jakim robot się aktualnie znajduje. Podstawą takiego rozumowania jest tzw. dyskretny, ergodyczny i stochastyczny proces Markowa [1,2,3]. Przyjmijmy zatem, że w statycznym środowisku zasady zachowania się robota w przestrzeni nie ulegają zmianie z upływem czasu, a szanse na osiągnięcie zmiany położenia są limitowane wyłącznie warunkiem dostępności przestrzeni (brakiem kolizji z otoczeniem). Proces uczenia się możemy stymulować oceną jaką przyznajemy za wykonane zadanie, a tę możemy wiązać z jakością jego wykonania. Mechanizm takiego postępowania nazywamy maszynowym uczeniem się ze wzmocnieniem, w którym wartość wzmocnienia odpowiada wysokości uzyskanej oceny, co pokazano na rys. 1.



Rys. 1. Mechanizm uczenia maszynowego

5. PROCES DECYZYJNY MARKOWA

Decyzja o tym „co dalej" (krok 2) może zostać podjęta w oparciu o tzw. dyskretny proces decyzyjny Markowa, który możemy zinterpretować jako czwórkę $<S,A,\xi,\delta>$, gdzie [3]:

- S – to skończony zbiór stanów $S=(s_i)$ – w jakich może znaleźć się robot w po zrealizowaniu kolejnego kroku elementarnego (współrzędne położenia i orientacji robota);
- A – to skończony zbiór akcji $A=(a_i)$ – który definiuje sposoby jakimi możliwe jest osiągnięcie kolejnego położenia (stanu); A może być zbiorem wszystkich możliwych do zrealizowania dróg (trajektorii-kroków elementarnych t_i) zmierzających do osiągnięcia zamierzonego celu czyli kolejnego stanu s_i;
- γ - wartość funkcji wzmocnienia $\gamma=\xi(s,a)$ wyrażona prawdopodobieństwem osiągnięcia celu; dla uproszczenia możemy przyjąć uśrednioną wartość funkcji wzmocnienia określoną wg zasady [2,3]:

$$\gamma = E\left[\sum_{it=0}^{n-1} \omega_{P_{wi}} \cdot r(P_{wi})\right]$$

gdzie: $\omega_{P_{wi}}$ – waga określająca ważność uzyskanej oceny po osiągnięciu kolejnego zadanego (wybranego stochastycznie) punktu wspomagającego P_{wi}, a $r(P_{wi})$ – przyjęta wartość oceny zachowania poprawnego wyrażona jako wartość prawdopodobieństwa osiągnięcia celu w położeniu P_{wi}.

- δ - funkcja przejścia pomiędzy kolejnymi stanami $\delta = P_r[\delta(s_i)]$ – określająca sposób osiągnięcia zadanego (wybranego) celu, określona prawdopodobieństwem jego osiągnięcia.

6. SZCZEGÓŁOWY ALGORYTM UCZENIA SIĘ ZE WZMOCNIENIEM

Uwzględniając poczynione założenia i dyskretny, stochastyczny proces Markowa możemy uszczegółowić określony podstawowy algorytm samouczenia robota poszukującego trajektorii w przestrzeni. Ponieważ, każda zdobyta nagroda wartości $\omega_{P_{wi}}$ jest wynikiem zrealizowanego przejścia o wartości kolejnego i-tego kroku elementarnego równego t_i zatem ustalony wcześniej iteracyjny algorytm poszukiwania trajektorii robota oparty na procesie decyzyjnym Markowa możemy teraz szczegółowo ustalić w następującej postaci:

- Krok 1: Zdefiniuj zadanie Γ podając położenie docelowe robota P_k (położenie początkowe P_1 jest położeniem bieżącym),
- Krok 2: Określ $S(s_i)$ – potencjalnie możliwy do zrealizowanie zbiór stanów robota w jego przestrzeni roboczej,
- Krok 3: Zdefiniuj $\gamma=\xi(s,a)$ – sposób oceniania podjętych działań robota, czyli funkcję wzmocnienia (nagrodę), oraz sposób jej parametryzacji, czyli wartości wag ω_{Pwi},
- Krok 4: Ustal strategię działania $A(a_i)$, czyli poszukiwania kolejnych kroków t_i;
- Krok 5: Wyznacz długość b_i kroku elementarnego t_i,
- Krok 6: Określ prawdopodobieństwo sukcesu związanego z osiągnięciem dostępnych w zbiorze S stanów s_i;

Wykonaj kolejno:

- Krok 7: Wykorzystując wielkość kroku elementarnej b_i wyznacz kolejną docelową pozycję wspomagającą P_{wi} i przyjmij: $P'_1=P_1$ i $P'_k=P_k=P_{wi}$,
- Krok 8: Dla każdej ze współrzędnych układu zadania określ długość drogi do przebycia:
- Krok 9: Dla każdego i z przedziału $i \in [1,k]$ wykonaj działania:
  a) ustal zbiór kolejnych stanów s_i czyli punktów wspierający dla kolejnego kroku i wyznaczając współrzędne możliwych do wykonania kroków elementarnych t_i,
  b) ustal wartość uzyskanej nagrody γ dla każdego zrealizowanego kroku elementarnego,
  c) wybierz ten z kroków t_i dla którego wartość nagrody była największa, jeżeli było więcej takich kroków t_i, wybierz ten dla którego większa liczba współrzędnych zapewnia spełnienie warunku „zbliżenia" się do celu, czyli zmniejszy odległość do celu trajektorii P_k – jeżeli nie ma takiego przerwij działanie;
  d) zapamiętaj: wartość i, uzyskane współrzędne $P_{wi}(x_i, y_i)$, wartość uzyskanej nagrody γ,
  e) jeżeli $i<k$ to przejdź do punktu a), jeżeli $i=k$ to droga robota zostaje ustalona jako ciąg kolejnych kroków elementarnych: $\Gamma_n = \underset{n}{\wedge} \underset{i\in(1,k)}{\wedge} \left\{\left(x_1^n, x_2^n,..., x_k^n\right), \left(y_1^n, y_2^n,..., y_k^n\right)\right\}$. Przejdź do kroku 9;
- Krok 10: Jeżeli przy $i=k$ dla znalezionego punktu wspomagającego zachodzi zależność, że: $P_{wi} - P_k > \varepsilon$, gdzie ε – przyjęta dokładność planowania trajektorii, to przejdź do kroku 12.
- Krok 11: Przyjmij: $P_1=P_{wi}$ oraz $i=i+1$ i przejdź do kroku 7;
- Krok 12: Zachodzi warunek, że $P_{wi} - P_k \le \varepsilon$. KONIEC.

Oczywiście tak uzyskana droga jest łamaną, która wymaga odpowiedniej obróbki „wygładzającej", wykonanej zgodnie zobowiązującymi w robotyce zasadami.

7. WNIOSKI

Przedstawiona metoda planowania bezkolizyjnych trajektorii robota pozwala na poszukiwanie i wyznaczenie możliwych do zrealizowania przez robot dróg w jego przestrzeni bezkolizyjnej. Pełne zaadoptowanie omówionej metody do procesu komputerowego programowania robota wymaga:

- określenia strategii postępowania pozwalającej wyznaczyć optymalny tor ruchu robota,
- znalezienia odpowiedniej reprezentacji uzyskanej drogi w postaci gładkiej krzywej mogącej stanowić definicję jego trajektorii (np. wielomian Bernsteina-Baziera [2]), która pozwalałaby również na wykorzystanie jej w algorytmie sterowania robotem.

LITERATURA

1. Cichosz P.: Systemy uczące się. WN-T, Warszawa 2000.
2. Dulemba I.: Metody i algorytmy planowania ruchu robotów mobilnych i manipulacyjnych. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001.
3. Kost G.: Collision-Free Robot's Trajectory Planning Based Upon Self-Learning Algorithm Assumption. in Computer Integrated Manufacturing. Advanced Design and Management, Edts: B.Skołud, D. Krenczyk. WN-T, Warszawa 2003, pp.287-291.