XVI Konferencja Naukowo-Techniczna
TKI2022
TECHNIKI KOMPUTEROWE W INŻYNIERII
18–21 października 2022

Zastosowanie głębokiego uczenie przez wzmacnianie w wyznaczaniu optymalnej trajektorii manipulatora o pięciu stopniach swobody

Artur Stefańczyk[1], Patryk Bałazy[1], Paweł Knap[1], Tymoteusz Turlej[1]
[1]Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
email: astefanczyk@student.agh.edu.pl, balazy@student.agh.edu.pl
pknap@student.agh.edu.pl, turlej@agh.edu.pl

STRESZCZENIE: Wyznaczanie optymalnej trajektorii manipulatora jest matematycznie złożone ze względu na czasochłonne rozwiązywanie kinematyki odwrotnej, której złożoność rośnie wraz ze wzrostem stopni swobody układu. W tym artykule przedstawiono rezultaty treningu agenta z funkcją nagrody zaprojektowaną specjalnie do kontroli manipulatora o pięciu stopniach swobody. W odróżnieniu od klasycznego podejścia, rozwiązanie przedstawione w tej publikacji oparte jest o sygnały wyjściowe modelu. Agent wyznacza przyszłe akcje na podstawie pozycji kątowej, prędkości kątowej w poszczególnych aktuatorach manipulatora oraz informacji zwrotnej w postaci funkcji nagrody. Funkcja nagrody zawiera wskaźniki całkowe jakości kontroli, co pozwala na znaczące przyśpieszenie procesu nauki sieci neuronowej. Agent uczony był, aby dotrzeć do zadanego punktu ze stałej oraz z losowej pozycji początkowej manipulatora w optymalnym czasie oraz przy wydajnych wymaganiach energetycznych. Dodatkowo proces uczenia przeprowadzony jest w środowisku zbudowanym na trójwymiarowym modelu CAD manipulatora, zamiast modelu matematycznego.

SŁOWA KLUCZOWE: sztuczna inteligencja, symulacje komputerowe, robotyka

1. Przedmiot i Zakres

Sztuczna Inteligencja, w szczególności Sztuczne Sieci Neuronowe są używane w wielu dziedzinach inżynierii. Są one niezwykle skuteczne w rozwiązywaniu wielowymiarowych, nieliniowych problemów optymalizacyjnych [1]. Te cechy pozwalają na skuteczne planowanie trajektorii układów mechatronicznych i robotycznych [2]. Jednym z najczęstszych problemów w systemach kontroli układów robotycznych jest wydajne wyznaczanie trajektorii manipulatorów o wielu stopniach swobody. Wraz ze wzrostem różnorodnych zastosowań wspomnianych manipulatorów, a zatem czynników które należy rozważyć przy planowaniu trajektorii (np. omijanie przeszkód) tradycyjne metody mogą okazać się niewystarczające i czasochłonne. Jedną z najbardziej obiecujących [3], a zaraz najbardziej innowacyjnych metod pozwalających na rozwiązanie opisanego problemu jest wykorzystanie uczenia przez wzmocnienie (RL). To rozwiązanie zastępuje skomplikowany system kontroli poprzez zastosowanie agenta, który wykonuje wybrane akcje na podstawie otrzymanych obserwacji z wirtualnego środowiska. W tym artykule prezentowane są wyniki nauczania przez wzmacnianie zastosowanego do wyznaczenia optymalnej trajektorii manipulatora 5DOF z agentem uczonym z zastosowaniem Deep Deterministic Policy Gradient(DDPG) i metody Aktor-Krytyk, symulowanego w wirtualnym środowisku.

Głębokie uczenie przez wzmacnianie jest relatywnie nowym sposobem na rozwiązywanie złożonych, wielowymiarowych problemów. Jego kluczowym elementem jest funkcja nagrody, która jest eksplorowana w procesie uczenia. Wybranie odpowiedniej funkcji nagrody jest kluczowe, gdyż wpływa ona bezpośrednio na późniejszą konwergencje całego treningu, nie tylko na konieczny czas, ale także na to czy w ogóle dojdzie do konwergencji.

Funkcja nagrody powinna odzwierciedlać jakość wybranych akcji. W tym wypadku jest ona bezpośrednio zależna od dystansu pomiędzy efektorem manipulatora, a docelowym punktem w globalnym układzie odniesienia. Udany trening powinien skutkować agentem, który może dotrzeć do danego punktu w przestrzeni poprzez wybranie serii akcji z dowolnego stanu początkowego. W tej publikacji akcje reprezentować będą wartość momentu zadanego do danego przegubu manipulatora.

Aby umożliwić przeprowadzenie treningu sieci, konieczne jest zebranie danych zwrotnych z wirtualnego środowiska, w którym przeprowadzona jest symulacja. Aby wyznaczyć optymalną trajektorie, jest konieczne aby znać kąt oraz prędkość kątową w każdym przegubie manipulatora; te wartości będą nazywane obserwacjami w dalszej części tej publikacji.

2. Opis badanego układu

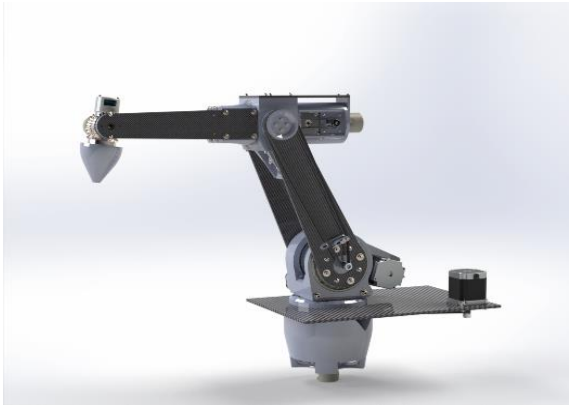
a) Metoda Aktor-Krytyk

Metoda Aktor-Krytyk jest podzbiorem większej grupy algorytmów nazywanych metodami REINFORCE. Głównym założeniem tych metod jest stworzenie dwóch sieci neuronowych. Jedna z nich jest siecią strategii, do której odnosi się jako sieć Aktora; decyduje ona o tym jakie akcje powinny zostać wykonane w danym stanie. Druga sieć neuronowa nazywana jest siecią Krytyka; jest to sieć wartości, która odpowiada za ocenianie decyzji podjętych

przez Aktora. Krytyk estymuje wartość stanu i porównuje ją ze zdobytą nagrodą r i następną wartością stanu.
Jako że oszacowanie przyszłej nagrody jest prawie niemożliwe w dalszej perspektywie, te podejście musi się ograniczać do najbliższej przyszłości. Ta metoda znana jest pod nazwą Bootstraping. W każdym kroku danego epizodu obie sieci poddane są uczeniu.

b) Symulacja Manipulatora

Symulacja manipulatora została stworzona przy użyciu Simscape'a oraz multi-body toolboxa ze środowiska MATLAB. Plug-in Inventora, MultibodyLink został użyty, aby przesłać niezbędne pliki o rozszerzeniu STEP. Trening sieci neuronowej także został przeprowadzony w środowisku MATLAB używając toolboxa Reinforcement Learning. Model manipulatora przedstawiono na rys. 1.



Rys. 1. Model manipulatora 5DOF

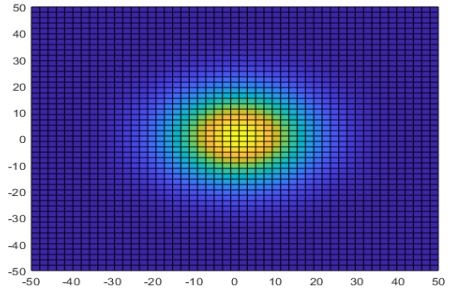
Wszystkie symulacje, modele i programy autorzy udostępnili w formie repozytorium GitHub:
https://github.com/SzachTech/5DOF_RL

3. Modelowanie numeryczne

Aby rozwiązać problem poruszony w tej publikacji konieczne było stworzenie wydajnej funkcji nagrody. Nasza funkcja nagrody jest sumą trzech komponentów, z czego dwa są nieliniowe. Funkcja nagrody zadana jest równaniem (1):

$$\xi(\theta) = \alpha_1 \cdot e^{(-\alpha_2 \cdot \theta^2)} - \alpha_3 \cdot N - \alpha_4 \cdot \Sigma u^2 \qquad (1)$$

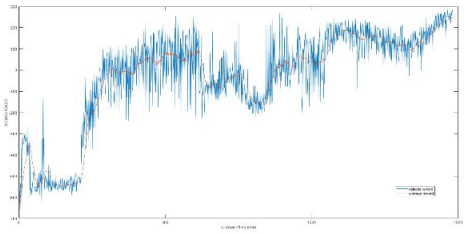
Gdzie wartości α_1, α_2, α_3, α_4 to metaparametry, które muszą zostać zoptymalizowane. W ramach tego badania parametr θ reprezentuje absolutny dystans pomiędzy efektorem końcowym manipulatora i punktem docelowym. Wartość N reprezentuje liczbę kroków przypadającą na dany epizod. Wartość N jest początkowo stała i może zmienić się tylko wtedy, gdy symulacja dojdzie do warunku końcowego. W tym przypadku, N będzie zależne od dystansu θ, jako że warunek końcowy przyjmie wartość prawdziwą, gdy efektor manipulatora dotrze do punktu docelowego z zadanym marginesem błędu. W związku z tym dla agenta korzystnym jest, aby dotrzeć do punktu docelowego, w jak najmniejszej ilości kroków, konsekwentnie w najszybszym czasie. Trzeci komponent jest konieczny, aby zminimalizować energię potrzebną do zrealizowania danego zadania. Wartość u reprezentuje wartości akcji wykonanych w poprzednim kroku. Ostatni komponent minimalizuje wysiłek aktuatorów i zapobiega drganiom. Przykładową reprezentację funkcji nagrody przedstawiono na rys. 2.



Rys. 2. Mapa ciepła przedstawia wartości funkcji nagrody, gdy punkt docelowy jest w pozycji (0, 0).

4. Wyniki symulacji

Początkowo, agent został nauczony aby znajdować optymalną trajektorie z ustalonego stanu początkowego do stacjonarnego punktu docelowego. Krok uczenia Aktora został ustalony na 10^{-4}, a dla Krytyka 10^{-3}. Czas każdego epizodu został ustalony na 2 s ze stałą liczbą kroków na epizod równą 80. Warunek końcowy zachodził, gdy końcowy efektor manipulatora znajdował się w odległości 0,1 cala od punktu docelowego. Po 1482 epizodach, zachodzi konwergencja ze średnią nagrodą 254,174 w czasie 2789 s. Krzywa konwergencji jest widoczna na rys. 3.



Rys. 3. Krzywa konwergencji, gdzie średnia jest obliczana z 25 epizodów.

5. Podsumowanie

W publikacji przedstawiono możliwość zastosowania głębokiego uczenia przez wzmacnianie w planowaniu optymalnej trajektorii manipulatora o 5 stopniach swobody.

Przedstawiona metoda używa reprezentacji środowiska w postaci trójwymiarowego modelu CAD co znacznie przyśpieszyło proces uczenia. Takie podejście skraca modelowanie systemów dynamicznych i dostarcza bardziej konkretne wyniki. Model manipulatora może zostać zmieniony, a reszta programu może nadal zostać użyta do treningu nowego agenta.

Literatura

[1] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: Asurvey. Heliyon, 4(11):e00938, 2018.

[2] Ahmed A Hassan, Mohamed El-Habrouk, and Samir Deghedie. In-verse kinematics of redundant manipulators formulated as quadratic programming optimization problem solved using recurrent neural networks: A review. Robotica, 38(8):1495-1512, 2020.

[3] SWETHA Danthala, SEERAMSRINIVASA Rao, KASIPRASAD Mannepalli, and Dhantala Shilpa. Robotic manipulator control by using machine learning algorithms: A review. International Journal of Mechanical and Production Engineering Research and Development, 8(5):305-310, 2018.