

Inżynieria Mechatroniczna

Teoria Sterowania: Sterowanie inteligentne

Ziemowit Dworakowski
AGH w Krakowie

1

Kiedy klasyczne metody nie zdają egzaminu? *SD*

Zakładaliśmy do tej pory wykorzystywanie układów:

- Liniowych
- Stacjonarnych
- Bez dużego opóźnienia
- Prosty w identyfikacji

Jak podejść do sterowania, gdy obiekt sterowania jest silnie nieliniowy, nie daje się zlinearyzować wokół punktu pracy i/lub jest niestacjonarny?

2

Kiedy klasyczne metody nie zdają egzaminu? *SD*

Sztuczna inteligencja

Inteligentna identyfikacja układów (w tym budowa modeli układów nieliniowych oraz nadążanie za zmianami)

Inteligentne sterowanie (w tym adaptacyjne oraz pozwalające na kontrolę układów nieliniowych)

Sterowanie rozmyte (pozwalające na wykorzystanie doświadczenia eksperta)

Ze względu na ograniczony zakres czasu na wykładzie i laboratoriach zajmiemy się wyłącznie pokazaniem wybranych przykładów oraz przedstawieniem ogólnej zasady działania metod. Nie będziemy rozważać układów nieliniowych (skupimy się na problemach niestacjonarnych), ale ogólna zasada działania przedstawianych rozwiązań pozostaby ta sama również w przypadku pojawiania się nieliniowości w układzie sterowania.

3

Sterowanie inteligentne to samodzielny moduł, oceniany osobno.
W jego skład wchodzi:

4 instrukcje laboratoryjne:

- Ewolucyjna identyfikacja obiektu sterowania
- Ewolucyjna optymalizacja kontrolera PID
- Sterowanie predykcyjne z modelem opartym o sieć neuronową
- Sterowanie predykcyjne oraz optymalizacja ewolucyjna układów zmiennych w czasie

Oceniane jak na PSIWSUM/PSIWMUM: Im więcej zadań zrobionych na zajęciach, tym wyższa ocena. Instrukcje oraz materiały do zajęć do pobrania z <http://galaxy.agh.edu.pl/~zdu/Materials/CT/>

1 test

Na 1 zajęciach w ramach modułu. Możliwy do poprawy 2 razy.
Wymagany zakres wiedzy obejmuje niniejszy wykład.

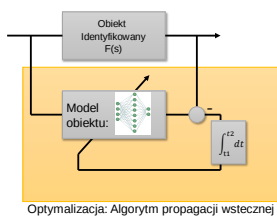
Ocena końcowa z modułu stanowi średnią ważoną ocen z instrukcji z wagą 1 oraz testu z wagą 2. Aby uzyskać zaliczenie modułu, wszystkie oceny muszą być pozytywne.

4

Inteligentne algorytmy identyfikacyjne

5

Identyfikacja neuronowa



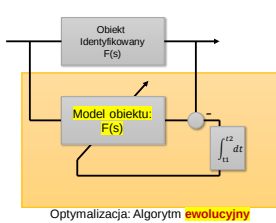
- + Możliwość identyfikacji nieliniowości
- + Obiekt identyfikowany nie musi być wyłączany z użytkowania
- + Szybka nauka
- + Możliwość łatwej adaptacji (cykliczne douczanie sieci)

- Reprezentacja nie jest intuicyjna
- Brak możliwości dalszego badania zidentyfikowanego obiektu (np. stabilności)

6

Identyfikacja ewolucyjna

SD



+ Zachowujemy intuicyjną reprezentację (w tym możliwość jej dalszych badań)

+ Obiekt identyfikowany nie musi być wyłączany z użytkowania

+ Możliwość łatwej adaptacji

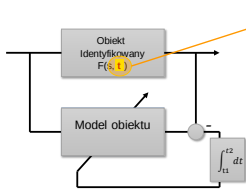
- Problem złożony obliczeniowo

- Konfiguracja AG jest trudna

7

Identyfikacja w problemach niestacjonarnych

SD



Co w przypadku, gdy obiekt zmienia się w czasie?

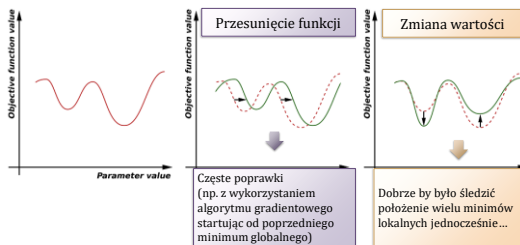
Reprezentacja powinna pozwalać na łatwą adaptację do nowych warunków (zmiany są zazwyczaj płynne, nie ma potrzeby identyfikowania „od zera”)

8

Optymalizacja adaptacyjna

SD

Dwa rodzaje zmian funkcji celu – zazwyczaj jednocześnie, w różnym stopniu:



9

Optymalizacja adaptacyjna

Wymagania:

- AG powinien utrzymać populację w wielu minimach jednocześnie  Spora populacja, niski nacisk selektywny, być może dodatkowe podtrzymanie różnorodności
- AG powinien nadążać za zmianami położenia minimów  Krok mutacji nigdy nie zbliża się do zera. Po początkowym przeszukianiu przestrzeni i uzyskaniu konwergencji krok pozostaje na stałym poziomie
- AG powinien zachowywać zdolności eksploracyjne przez cały czas trwania optymalizacji (zdolność do znajdowania nowych minimów)  Część populacji jest generowana losowo lub też mutowana z zastosowaniem relatywnie dużego kroku mutacji

10

Alternatywna konfiguracja: elastyczny genom

Jeśli osobnik znajduje się **w pobliżu** minimum lokalnego, tylko **mały** zakres mutacji umożliwi poprawę rezultatu

Jeśli osobnik znajduje się **daleko** od minimum lokalnego, tylko **duży** zakres mutacji umożliwi poprawę rezultatu

11

Alternatywna konfiguracja: elastyczny genom

Jeśli osobnik znajduje się **w pobliżu** minimum lokalnego, tylko **mały** zakres mutacji umożliwi poprawę rezultatu

Jeśli osobnik znajduje się **daleko** od minimum lokalnego, tylko **duży** zakres mutacji umożliwi poprawę rezultatu

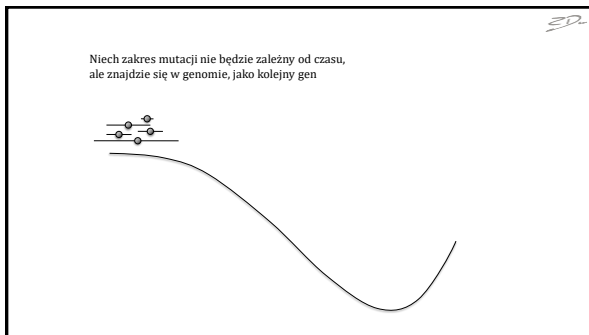


Odwróćmy wnioskowanie...

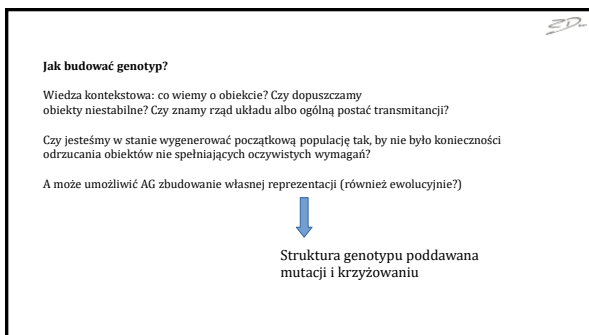
Skoro osobnika wygenerowano **małym** zakresem mutacji a poprawił rezultat, znajduje się **w pobliżu** minimum lokalnego

Skoro osobnika wygenerowano **dużym** zakresem mutacji a poprawił rezultat, znajduje się **daleko** od minimum lokalnego

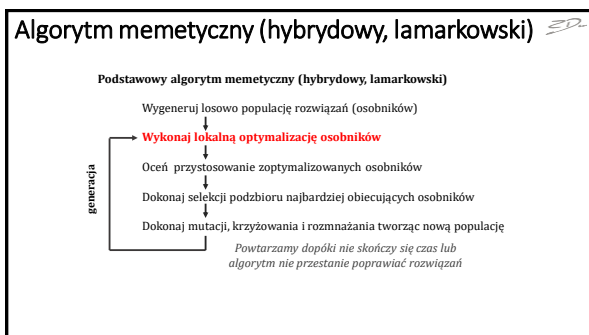
12



13



14



15

Algorytm memetyczny (hybrydowy, lamarkowski) *SD*

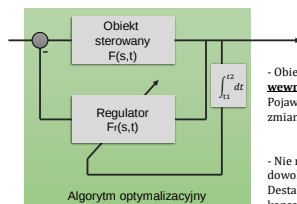
- + Wymagana mniejsza populacja
- + Nadążanie za zmianami funkcji celu staje się dużo prostsze
- Jeszcze większa ilość metaparametrów do konfiguracji
- Skomplikowany w implementacji
- Zazwyczaj wymaga więcej mocy obliczeniowej od klasycznego AG

16

Sterowanie adaptacyjne *SD*

17

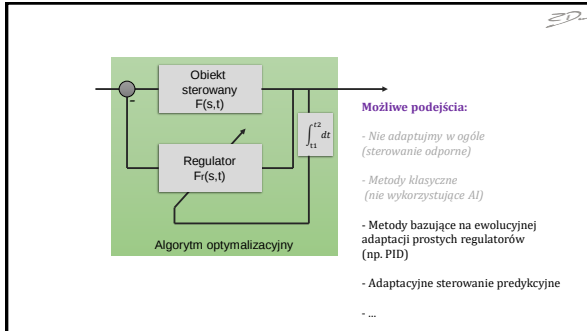
Dlaczego teraz nie będzie już tak łatwo... *SD*



- Obiekt sterowany znajduje się **wewnątrz** pętli optymalizacyjnej:
Pojawia się konieczność nadążania za jego zmianami dużo szybciej

- Nie możemy pozwolić sobie na testowanie dowolnych parametrów:
Destabilizacja obiektu będzie miała realne konsekwencje.

18



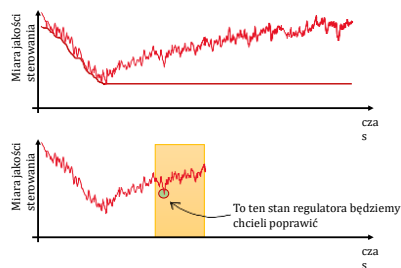
19

Ewolucja regulatorów

- Jako punkt wyjściowy potrzebny jest „w miarę dobry” regulator (np. PID, aby zapewnić poprawną regulację od samego początku)
- Obiekt musi być „łatwy w regulacji” – zmiany obiektu muszą być powolne aby zapewnić możliwość nadążenia za nimi
- Funkcja celu nie powinna posiadać wielu minimów lokalnych (przejście od dowolnego regulatora początkowego do położenia optymalnego powinno być możliwe na drodze ciągłych niewielkich poprawek)
- Obiekt musi być stabilny (tj. odłączenie regulatora nie powinno stanowić zagrożenia dla obiektu)

20

Rozwiązanie na bazie algorytmu 1+1



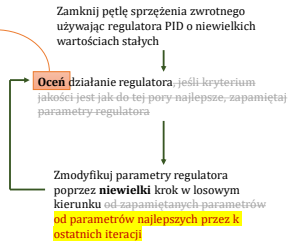
21

Rozwiązanie na bazie algorytmu 1+1

Idealnie: poprzez testowy, znany sygnał (zawsze ten sam)

Dobrze: poprzez porównanie sygnałów wejściowych do wyjściowych

Gorzej: poprzez uśrednienie działania systemu w długim czasie



22

Rozwiązanie na bazie algorytmu 1+1

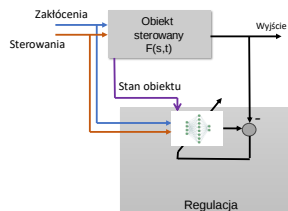
+ Algorytm 1+1 (w przeciwieństwie do pełnego AG) zapewnia na tyle szybką konwergencję, że w korzystnych warunkach może nadążyć za zmianami obiektu

ALE:

- Konieczność uśredniania wskazań (jeśli nie mamy możliwości pełnego monitorowania wejść) powoduje znaczne spowolnienie konwergencji
- Im szybciej obiekt się zmienia, tym bardziej trzeba skrócić okno z którego wybierane jest rozwiązanie do modyfikacji: możliwe dalsze spowolnienie konwergencji

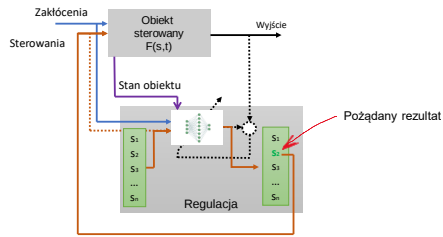
23

Neuronowe sterowanie predykcyjne



24

Neuronowe sterowanie predykcyjne



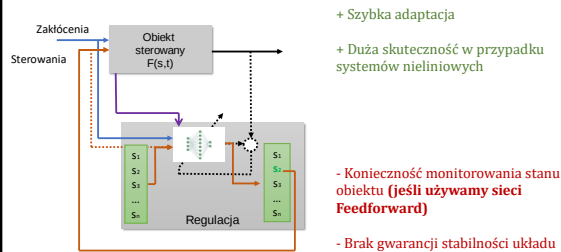
25

Neuronowe sterowanie predykcyjne

1. Uczymy sieć neuronową do przewidywania zachowania układu w zależności od obserwowanych sterowań, pozostałych wejść oraz stanu układu
2. Podajemy do sieci neuronowej różne możliwe sterowania wybierając ostatecznie to, które minimalizuje wybrane kryterium jakościowe dla danego stanu układu oraz pozostałych wejść
3. Używamy wybranego sterowania w rzeczywistym układzie
4. Co jakiś czas dotrenowujemy sieć na podstawie nowo zebranych danych

26

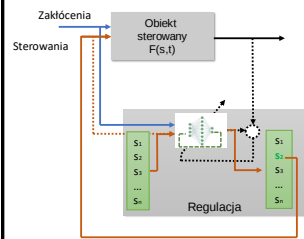
Neuronowe sterowanie predykcyjne



27

Neuronowe sterowanie predykcyjne

SD



+ Szybka adaptacja

+ Duża skuteczność w przypadku systemów nieliniowych

+ **Brak konieczności monitorowania stanu obiektu** (jeśli korzystamy z sieci rekurencyjnych)

- Konieczność monitorowania stanu obiektu (jeśli używamy sieci Feedforward)

- Brak gwarancji stabilności układu

28

Materiał do powtórki:

SD

- 1) Jak działa genetyczna identyfikacja układu sterowania?
- 2) Jak dostosować działanie algorytmu genetycznego do problemów zmiennych w czasie?
- 3) Jak działa adaptacyjne sterowanie z zastosowaniem metody PID (np. w oparciu o algorytm 1+1)?
- 4) Jak działa neuronowe sterowanie predykcyjne?
- 5) Co daje zastosowanie w sterowaniu predykcyjnym sieci rekurencyjnych?

29
