

Systemy Inteligentnego Przetwarzania  
— Projekt —  
Wahadło odwrócone (sieć neuronowa)

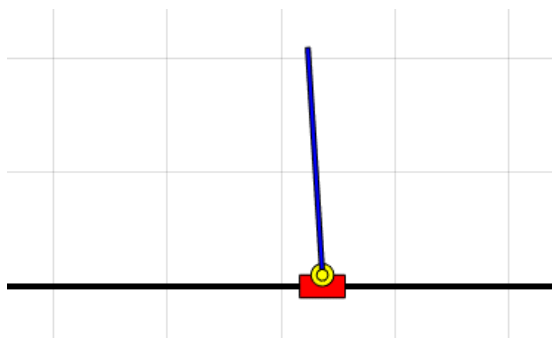
Adrian Frydmański, Dawid Gracek

17 grudnia 2017

# 1 Wstęp teoretyczny

Odwrócone wahadło niewiele różni się od swojego „zwykłego” odpowiednika. Jest swobodnie wiszącym prętem przymocowanym do wózka. Wózek z kolei ma możliwość poruszania się wzdłuż osi (w jednym wymiarze, acz nie stoi na przeszkodzie, żeby rozszerzyć problem wahadła do dwóch wymiarów). Układ ten:

- posiada dwa punkty równowagi: stabilny, kiedy wahadło spoczywa w położeniu dolnym i niestabilny, kiedy wahadło skierowane jest pionowo ku górze,
- jest tzw. obiektem niedosterowanym, ponieważ wielkości sterowanych możemy wyróżnić więcej niż jest wejść w układzie.



Rysunek 1: Model wahadła

Jedyną wielkością, która wpływa na stan układu jest siła przyłożona do wózka, którego przemieszczanie się wprawia w ruch wahadło. Taki układ regulacji może mieć kilka celów:

- stabilizacja wahadła w położeniu górnym,
- regulacja położenia wózka w odniesieniu do całego stanowiska,
- realizacja algorytmów umożliwiających rozbijanie wahadła z pozycji dolnej i doprowadzenie go to pozycji górnej.

Ów projekt zakłada realizację 2. pierwszych celów poprzez generowanie odpowiednich nastaw regulatora kontrolującego ruch wózka i wprowadzenie go do położenia górnego ze stanu początkowego, gdy wahadło jest nachylone względem ziemi pod kątem mniejszym, niż  $90^\circ$ . Nastawy mają być generowane przez wyuczoną sieć neuronową.

Matematyczny model wahadła jest opisany na stronie [jtjt.pl](http://jtjt.pl)<sup>1</sup> i stamtąd właśnie zaczerpnięte są obliczenia, z których korzysta symulacja.

<sup>1</sup><http://jtjt.pl/www/pages/odwrocone-wahadlo/LMIP.pdf>

## 2 Implementacja

Środowisko, w jakim została przeprowadzona symulacja i tworzenie sieci neuronowej, to Matlab R2017a. Dane testowe do uczenia sieci zostały pozyskane przez funkcję<sup>2</sup> do strojenia regulatora liniowo kwadratowego w zależności od „parametrów środowiska”:

- masa wózka,
- masa wahadła,
- długość od mocowania do środka ciężkości wahadła,
- współczynnik tarcia wózka.

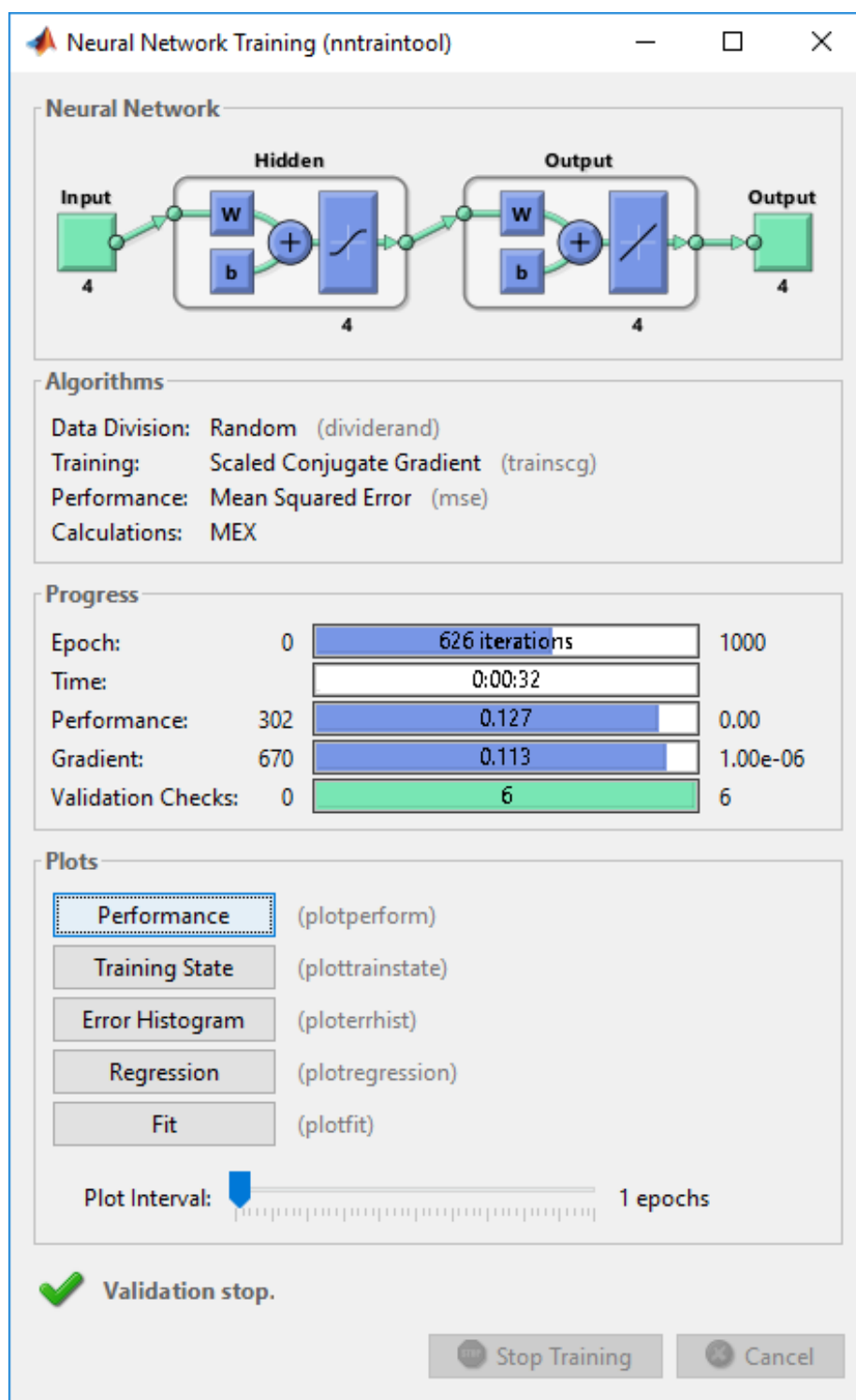
W głównej funkcji, `lqr_training`, zostało wygenerowane 100.000 losowych zestawów danych i znaleziono dla każdego z nich odpowiednie nastawy regulatora.<sup>3</sup> Następnie w narzędziu `nntool` została wygenerowana sieć neuronowa, która na owych zestawach nauczyła się dobierać parametry regulatora. Sprawdzono, że dla 4 neuronów — tylu, co wyjść — sieć działa wystarczająco dobrze. Sieć posiada jedną warstwę ukrytą. Przyjęto następujące wielkości podzbiorów zbioru 100.000 zestawów testowych:

- 70% — zbiór uczący
- 15% — zbiór walidacyjny
- 15% — zbiór testowy

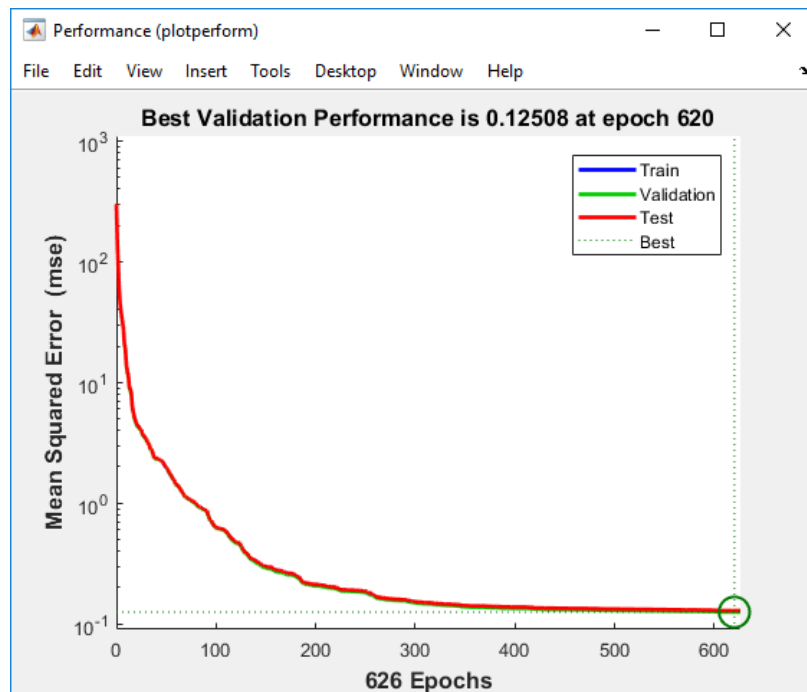
---

<sup>2</sup>Model i funkcja dostępne na <https://github.com/Jarczyslaw/Inverted-Pendulum>

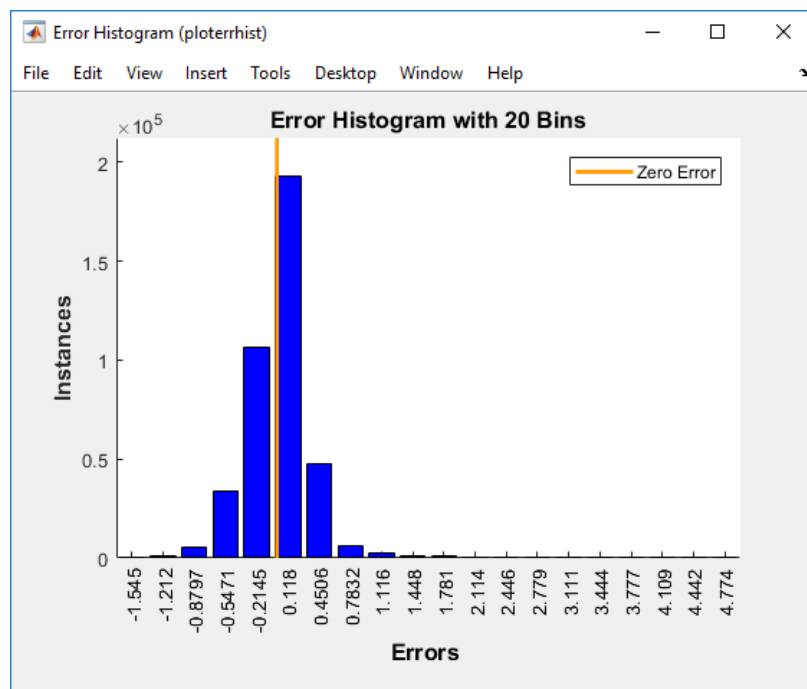
<sup>3</sup>Cały kod dostępny w repozytorium <https://github.com/Adrian94F/SIP>



Rysunek 2: Wygenerowana sieć i proces uczenia

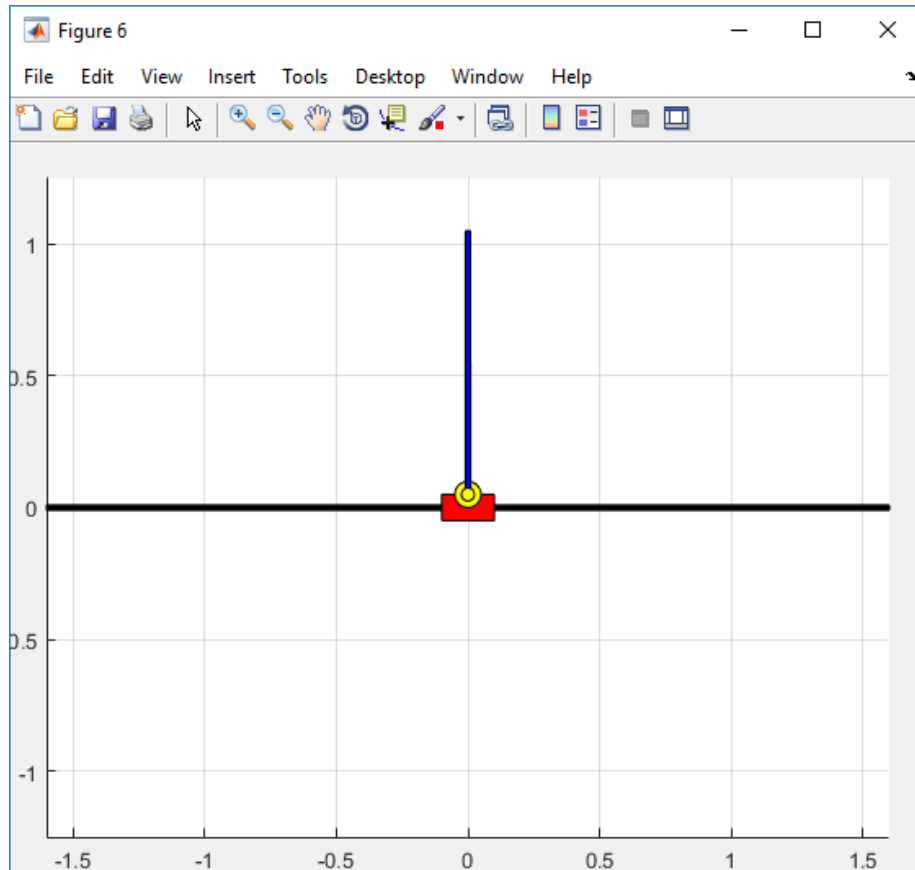


Rysunek 3: Wykres błędów od epok treningowych

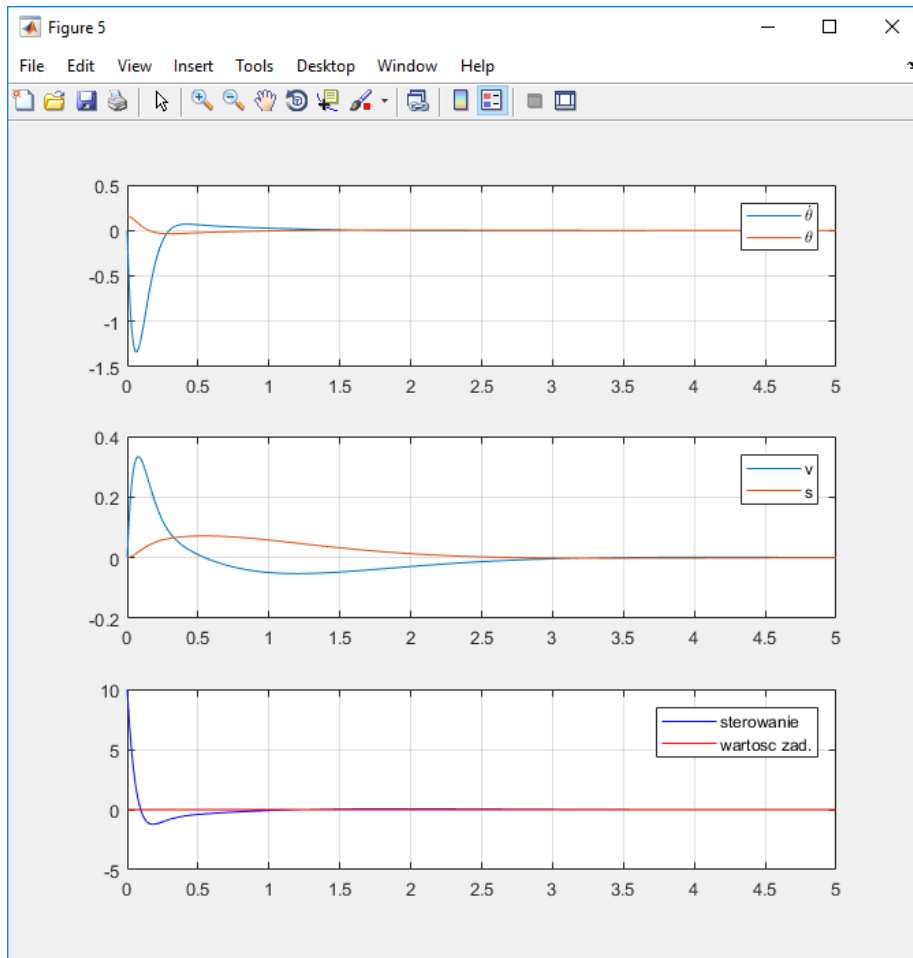


Rysunek 4: Histogram błędów

Po wygenerowaniu sieci został ponownie wylosowany zbiór zmiennych środowiska, na podstawie których sieć znalazła odpowiednie nastawy regulatorów. Została uruchomiona symulacja i przebiegła ona pomyślnie.



Rysunek 5: Osiągnięty stan równowagi podczas symulacji



Rysunek 6: Wyniki symulacji (od góry): kąt i pochodna kąta pomiędzy waha-  
dłem, a pionem, prędkość i pozycja wózka, przyłożona siła do wózka i zadana  
wartość zadana

### 3 Podsumowanie

Projekt pokazał, że sieć neuronową można zaprząć do przeróżnych zadań, nawet do dostosowywania nastaw regulatora w wahadle odwróconym. Jest to możliwe do zrobienia i proste w implementacji w takim środowisku, jak Matlab, aczkolwiek naszym zdaniem zbędne, gdyż funkcja strojąca regulator, dzięki której uzyskaliśmy dane do nauki sieci, działa wystarczająco szybko, a wynik działania sieci neuronowej zawsze będzie obciążony pewnym błędem w stosunku do wyniku działania owej funkcji.