Instytut Sterowania i Systemów Informatycznych
Uniwersytet Zielonogórski

# Inteligentne metody sterowania

Laboratorium

## Aproksymacja za pomocą sieci neuronowych

1. Zapoznać się z następującymi funkcjami przybornika `nnet` pakietu MATLAB: `nntool`, `nftool`, `newff`, `train` i `sim`.

2. Poziom wody w pewnym morzu zależy od pływu. Dokonano pomarów poziomu w interwałach 1 godzinnych. Uzyskane wyniki zestawiono w tabeli 1. Przybliżyć dane

Tabela 1. Zmierzone poziomy wody.

| $t$ [h] | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h(t)$ [m] | 1.0 | 1.32 | 1.6 | 1.54 | 1.41 | 1.01 | 0.6 | 0.42 | 0.2 | 0.51 | 0.8 |

pomiarowe za pomocą jednokierunkowej wielowarstwowej sieci neuronowej. Dokonać wyboru struktury neuronowego aproksymatora (liczba neuronów, typ funkcji aktywacji). Przetestować jakość aproksymacji stosując interwał czasu równy 6 minut. Wyrysować na wspólnym wykresie węzły aproksymacji oraz wyjście sieci neuronowej. Wyliczyć błąd aproksymacji w postaci sumy kwadratów błędów.

3. Dla układu zawieszenia magnetycznego wyznaczono charakterystykę statyczną. Pobudzając układ skokiem jednostkowym o różnej aplitudzie $u$ zarejestrowano wychylenie bolca w kierunku płytki elektromagnesu po zaniku przebiegów przejściowych $d$. Uzyskane wyniki przedstawia tabela 2.

Tabela 2. Węzły charakterystyki statycznej.

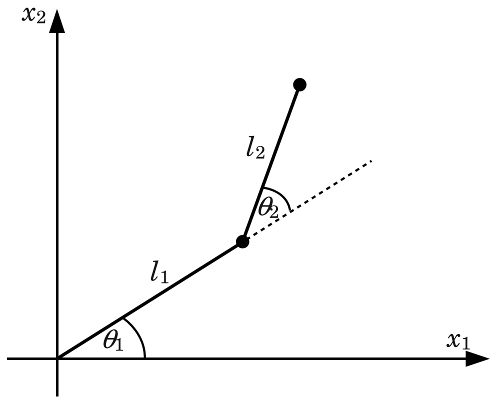
| $u$ [V] | 0 | 0.5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $d$ [mm] | 0 | 0 | 0.1 | 0.5 | 0.9 | 2 | 3.2 | 4.2 | 4.5 | 4.9 | 5 | 5 |

Za pomocą jednokierunkowej wielowarstwowej sieci neuronowej dokonać aproksymacji punktów pomiarowych.

4. Rozważmy zadanie kinematyki prostej manipulatora jednoprzegubowego zaprezentowanego na rys. 1:

$$\begin{aligned} x_1 &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2), \\ x_2 &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2), \end{aligned} \tag{1}$$

gdzie $\theta_1$, $\theta_2$ oznaczają kąty wychylenia przegubów, $l_1$ i $l_2$ są długościami ramion.




Rys. 1. Schemat manipulatora.

Skojarzone z (1) zadanie kinematyki odwrotnej ma postać:

$$\begin{aligned} \theta_1 &= \arctan\left(\frac{x_2}{x_1}\right) - \arctan\left(\frac{l_2\sin(\theta_2)}{l_1 + l_2\cos(\theta_2)}\right), \\ \theta_2 &= \arccos\left(\frac{x_1^2 + x_2^2 - l_1^2 - l_2^2}{2l_1 l_2}\right). \end{aligned} \tag{2}$$

Zaimplementować zadania kinematyki prostej i odwrotnej w postaci skryptów pakietu MATLAB następujące:

```
[x1,x2]=kin(th1,th2,l1,l2)
[th1,th2]=kin_od(x1,x2,l1,l2)
```

Rozwiązać zadanie kinematyki odwrotnej (2) za pomocą jednokierunkowej wielowarstwowej sieci neuronowej. Wygenerować zestaw 100 punktów uczących jako spiralę o środku w punkcie $(0.5, 0.5)$ ograniczoną kwadratem o wierzchołkach $(0, 0)$, $(0, 1)$, $(1, 0)$ i $(1, 1)$ (rys. 2). Do wyznaczenia próbek wejściowych użyć następującego kodu

```
phi=[1:0.05:50];
[x1,x2]=pol2cart(phi,0.01*phi);
x1=x1+0.5;x2=x2+0.5;
```

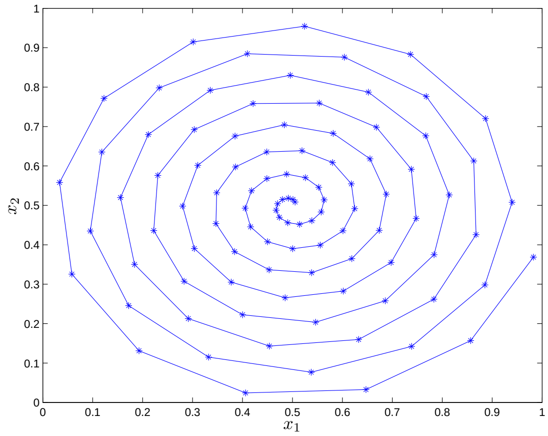
Założyć długości ramion $l_1 = l_2 = 1$. Manipular umieszczony jest w punkcie $(0, 0)$.

Następnie przy użyciu funkcji `kin_od` wygenerwować wyjściowe wzorce uczące. W tym przypadku sieć neuronowa będzie posiadała dwa wejścia ($x_1$ i $x_2$) i dwa wyjścia ($\theta_1$ i $\theta_2$). Po nauczeniu sieci neuronowej przetestować jakość działania modelu stosując różne trajektorie zadane:

(a) `x1=[0:0.1:1]; x2=[0:0.1:1];`,

(b) `x1=[0:0.1:1]; x2=x1.^2;`,

(c) `x1=[0:0.1:1, ones(1,10), 1:-0.1:0, zeros(1,10)];`
`x2=[zeros(1,10), 0:0.1:1, ones(1,10), 1:-0.1:0];`.




Rys. 2. Próbki uczące.

Porównać kąty uzyskane z modelu neuronowego $\hat{\theta}_1$, $\hat{\theta}_2$ z wygenerowanymi za pomocą modelu (2). Wykreślić odpowiednie wykresy.

Za pomocą modelu (1), na podstawie $\hat{\theta}_1$, $\hat{\theta}_2$ wyznaczyć aproksymowane współrzędne $\hat{x}_1$, $\hat{x}_2$ i porównać je z zadanymi trajektoriami. Wykreślić odpowiednie wykresy.

5. Za pomocą sieci neuronowej dokonać aproksymacji zadania kinematyki prostej (1).