

ŘEŠENÍ INVERZNÍ KINEMATIKY PLANÁRNÍHO MANIPULÁTORU MLP NEURONOVOU SÍŤÍ

Vladimír Hlaváč

ČVUT v Praze, Fakulta strojní, hlavac@fs.cvut.cz

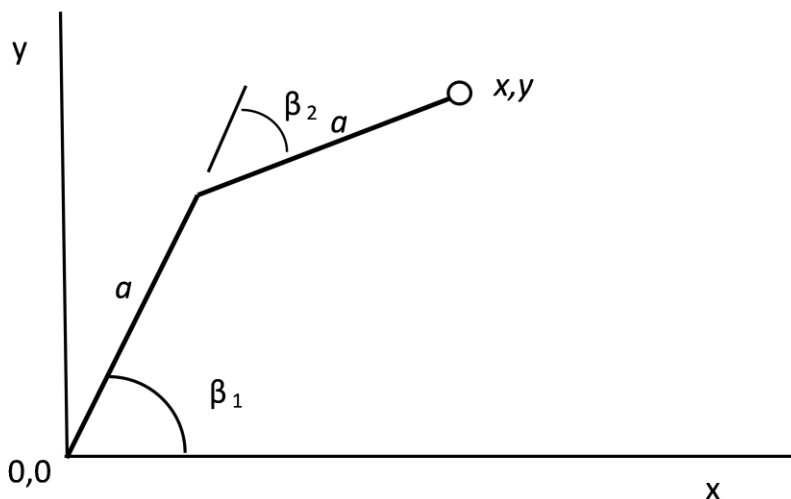
Abstrakt: Vícevrstvá neuronová síť typu MLP (multi-layer perceptron) je schopná interpolovat jakoukoli běžnou spojitou funkci, například kombinaci různých goniometrických funkcí a jejich inverzí. Následující článek popisuje její užití pro řešení úlohy inverzní kinematiky manipulátoru. Zatímco řešenou úlohu planárního manipulátoru lze řešit přímo, pro běžné robotické rameno pracující v prostoru vycházejí již vzorce poměrně komplikovanější a neuronová síť může představovat spolehlivější řešení.

Klíčová slova: Neuronová síť, MLP NN, planární manipulátor, inverzní kinematika.

1. Úvod

Problematika robotických manipulátorů je popsána například v doktorské prezentaci ing. Švejdy ke státní zkoušce [1]. Samotná disertační práce je dohledatelná na <https://portal.zcu.cz/portal/studium/>, zvolit “Kvalifikační práce” a vyhledat podle příjmení a typu práce (disertační). Kratší a přehlednější je disertační práce ing. Čejky [2].

Řešení s použitím neuronové sítě je popsáno v [3], [4]. Články jsou v podstatě totožné. Přináší popis úlohy a grafy z trénování neuronové sítě. V následujícím textu naleznete naproti tomu popis, jak postupovat a vizualizaci dosažených poloh po trénování neuronové sítě.



Obr. 1 Schematické znázornění úlohy. Poloha koncového bodu ramene je dána nastavením úhlů (kloubů) mezi jednotlivými segmenty a jejich délkou. Úhly se samozřejmě počítají jako kladné, pokud jsou proti směru hodinových ručiček; tak, jak je vyznačen úhel β_2 , má zápornou hodnotu.

2. Kinematika planárního robotického manipulátoru

Jak je znázorněno na obr. 1, polohu koncového bodu lze snadno spočítat:

$$x = a \cos \beta_1 + a \cos(\beta_1 + \beta_2) \quad (1)$$

$$y = a \sin \beta_1 + a \sin(\beta_1 + \beta_2) \quad (1b)$$

Tento výpočet se označuje jako přímá kinematika robotického manipulátoru. Většinou potřebujeme řešit opačnou úlohu, k zadaným souřadnicím navrhnout nastavení úhlů v kloubech manipulátoru, kterými jich lze dosáhnout. Tato úloha se označuje jako inverzní kinematika. Pro planární manipulátor je výpočet poměrně jednoduchý. Například můžeme nejprve převést požadovanou polohu koncového bodu do polárních souřadnic:

$$r = \sqrt{x^2 + y^2} \quad (2)$$

$$\varphi = \arctan(x/y) \quad (2b)$$

K dosažení tohoto bodu bude muset být délka ramena manipulátoru rovna r . Abychom jí nastavili, musíme nastavit úhel mezi oběma segmenty. Následující výpočet předpokládá, že délky obou segmentů budou stejné. Pak platí:

$$r = 2a \cos \beta_0 \quad (3)$$

$$\beta_0 = \arccos(r/2a) \quad (3b)$$

Pokud známe úhel mezi oběma segmenty β_0 a požadovaný úhel koncového bodu oproti počátku v polárních souřadnicích, můžeme snadno dopočítat oba nastavované úhly:

$$\beta_1 = \varphi + \beta_0 \quad (4)$$

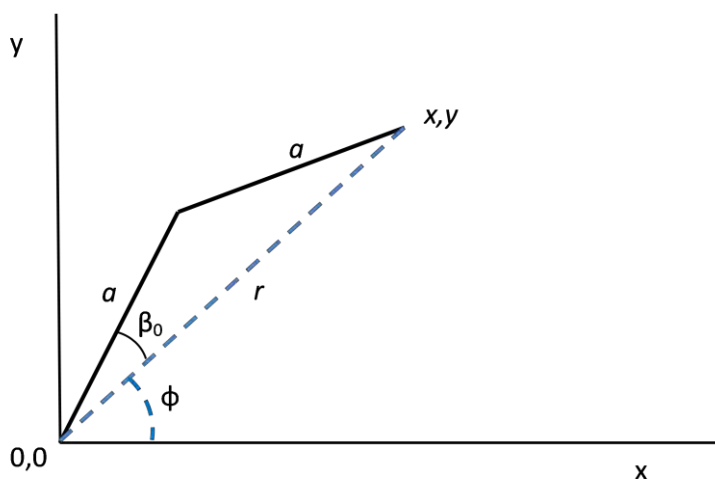
$$\beta_2 = -2\beta_0 \quad (4b)$$

Úloha má samozřejmě dvě řešení, podle toho, zda je rameno vyhnuto doprava, či doleva. Symetrické řešení je:

$$\beta_1 = \varphi - \beta_0 \quad (5)$$

$$\beta_2 = 2\beta_0 \quad (5b)$$

Úlohu lze řešit i když jsou ramena různě dlouhá, jen bude postup trochu komplikovanější.



Obr. 2 Transformace do polárních souřadnic.

3. Řešení neuronovou sítí

3.1 Příprava dat a trénování sítě

Data můžeme připravit přímo v Matlabu podle (1). U manipulátoru známe rozsah úhlů β i délku segmentů. Zvolíme vhodný krok a zadáme příslušné proměnné:

```
b1=(-0.2:0.1:2.4)';
b2=(0:0.1:2.6)';
```

Úhel, který svírají mezi sebou segmenty ramena manipulátoru, je generován vždy buď jako záporný, nebo jen jako kladný, abychom se vyhnuli duplicitnímu řešení kinematiky (pro výpočet inverzní funkce musí být funkce jednoznačná).

Budeme zkoušet různé počty neuronů ve skryté vrstvě. Celkový počet parametrů neuronové sítě lze spočítat:

$$n_v = (n_i + 1)n_h + (n_h + 1)n_o \quad (6)$$

n_v – počet stavitelných parametrů

n_i – počet vstupů

n_h – počet neuronů ve skryté vrstvě

n_o – počet výstupních neuronů

Například pro dva vstupy, dva výstupy a deset neuronů ve skryté vrstvě vychází 52 parametrů (+1 reprezentuje bias, posunutí pracovního bodu neuronu). Obecně platí, že pro trénování neuronové sítě by mělo být desetkrát tolik vzorků, tedy například 520.

Pro znázornění v grafu by ale vycházelo příliš mnoho bodů, proto pro vykreslení grafu připravíme jiné proměnné s mnohem menším počtem bodů:

```
bv1=(-0.2:0.2:2.4)';
bv2=(0:0.2:2.6)';
```

Pro obě dvojice proměnných b_1 , b_2 a bv_1 , bv_2 připravíme k úhlům i souřadnice x, y , zde jen b_1 :

```
x=L1*cos(b1)+L2*cos(b1+b2);
y=L1*sin(b1)+L2*sin(b1+b2);
```

Proměnné L_1 , L_2 reprezentují délky segmentů a musíme je nastavit předem.

Pole musíme spojit, knihovna `nftool` Matlabu pořadí pole vstupů a pole výstupů:

```
beta=[b1,b2];
xy=[x,y];
```

Alternativně můžeme vektor s úhly vygenerovat náhodně. Je to dokonce jednodušší, stačí zadat počet a pak přičíst minimální hodnotu (bývá záporná) a ponásobit rozsahem:

```
br1 = (rand(500,1)-0.2)*2.6;
br2 = rand(500,1)*2.6;
```

Kdyby rozsah obou úhlů byl stejný, šlo by pole hodnot vygenerovat v jednom kroku. Dále postupujeme stejně, vygenerujeme dosažené souřadnice, sloučíme do dvou polí a natrénujeme neuronovou sít.

Pro trénování sítě využijeme knihovnu `nftool`. Požadujeme, aby vstupem sítě byly rektangulární souřadnice, a výstupem úhly v kloubech manipulátoru, kterými jich lze dosáhnout. Takto musíme proměnné pro trénování zadat.

V dalších krocích zvolíme počet neuronů ve skryté vrstvě (osvědčil se mi velmi vysoký počet, například 25), a trénovací metodu. Nejlepší výsledky dává Bayesian regularization.

Na konci si necháme vygenerovat funkci pro Matlab, která pracuje s polem (matrix only function). Doporučuji přejmenovat (zde `nn2reg`) a uložit.

3.2 Zobrazení výsledků

Pro připravené testovací pole si necháme nově vygenerovanou neuronovou sítí spočítat úhly:

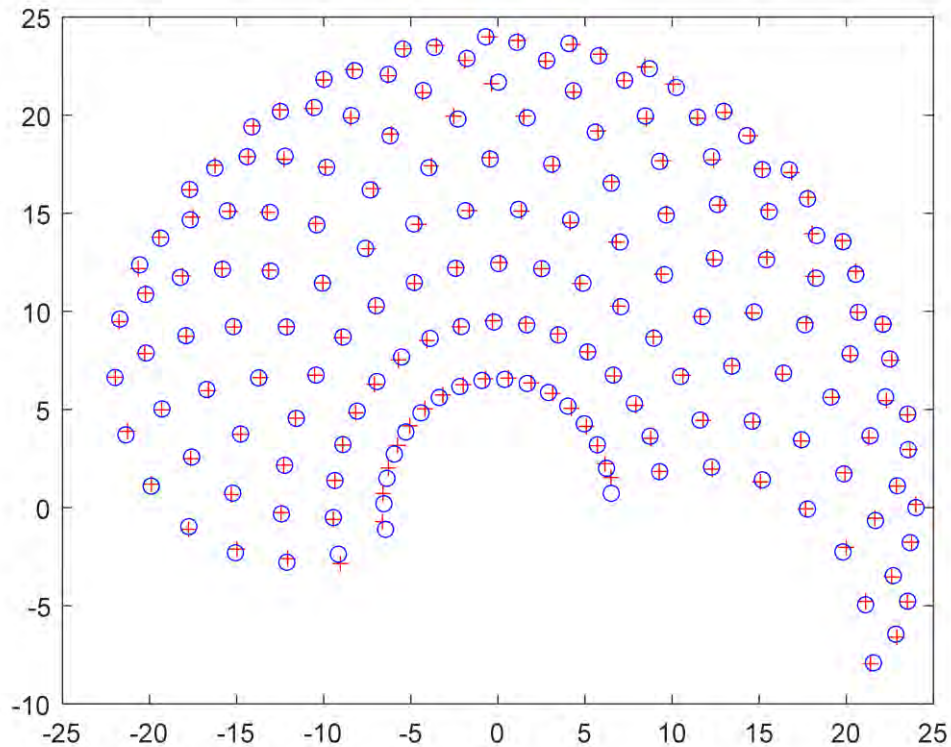
```
b = nn2reg(xy2);
```

Úhly opět dosadíme do přímé kinematiky

```
x=L1 * cos(b(:,1))+ L2*cos(b(:,1)+b(:,2));
y=L1 * sin(b(:,1))+ L2*sin(b(:,1)+b(:,2));
```

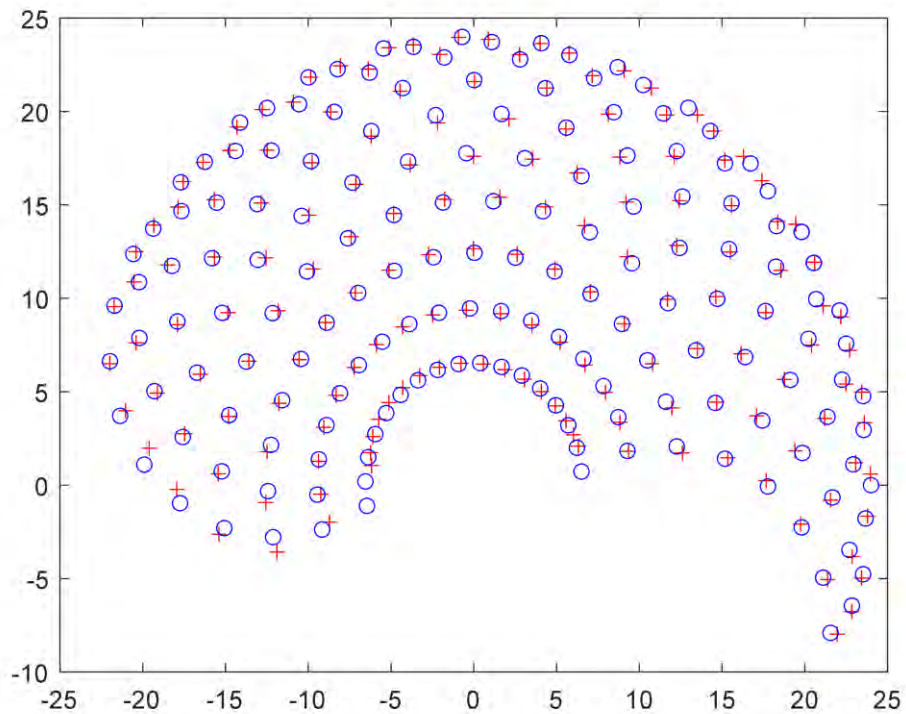
Zobrazíme v grafu. Zde jsou například kolečka jako cíle (správná hodnota) a křížky jako dosažená hodnota:

```
plot(x2,y2,'bo')
plot(x,y,'r+')
```



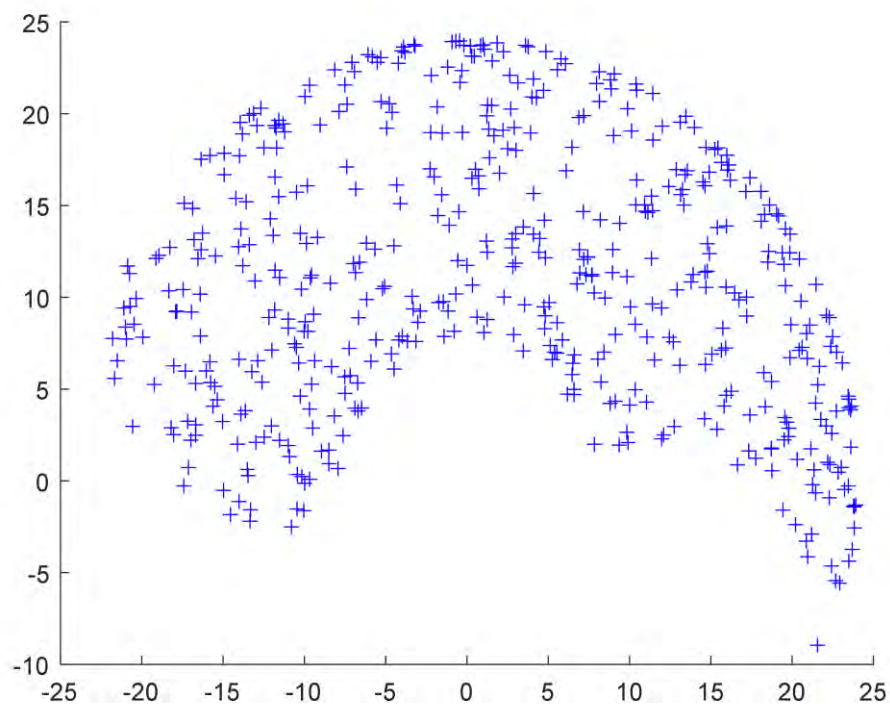
Obr. 3 Výsledek trénování. Modrá kolečka jsou zadaná data x,y , která se zadají jako vstup natrénované neuronové sítě, získané úhly se pomocí (přímé) kinematiky (1) převedou na souřadnice a ty se zobrazí červenými křížky. Délky segmentů byly 14 a 10.

Neuronová síť, jejíž výsledek je na obr. 3, měla 25 neuronů ve skryté vrstvě. Výsledek téhož experimentu, ale pro 15 neuronů:



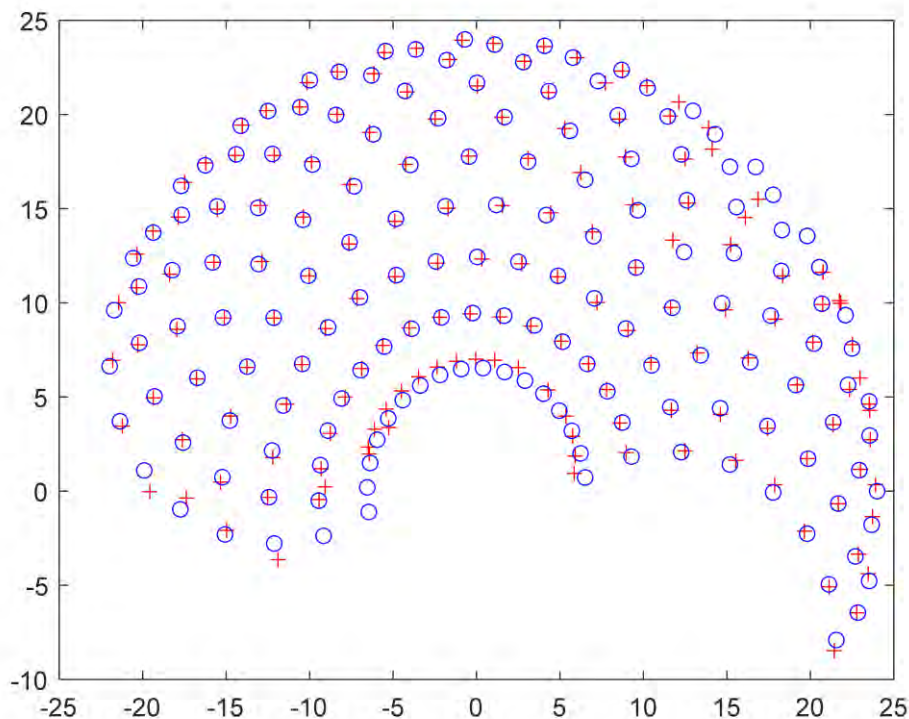
Obr. 4 Výsledek trénování. Stejná trénovací data, ale jen 15 neuronů ve skryté vrstvě.

Pokus byl zopakován s náhodně vygenerovanými daty (náhodně generované body jsou na následujícím obrázku):



Obr. 5 Pro tyto body bylo vytvořeno zadání pro trénování neuronové sítě.

Není ovšem důvod zobrazovat výsledek s náhodnými daty, takže i pro vizualizaci výsledku použijeme pravidelná data:



Obr. 6 Výsledek trénování při použití náhodně generovaných dat (natrénovaná síť byla použita pro stejné body, jako na obr. 3 a obr. 4 a výsledky vyneseny do grafu). Největší rozdíl je v pokrytí okrajových oblastí, kde je možné, že se vzorky pro trénování ani nevygenerovaly (zatímco pravidelně generovaná data obsahují i hraniční body na mezi dosahu manipulátoru).

4. Závěr

Vícevrstvá neuronová síť poskytuje jednoduché řešení inverzní kinematiky manipulátoru. Tuto metodu lze rozšířit nejen na 3D manipulátory, ale také na redundantní manipulátory. Možné řešení je popsáno v [5].

Literatura

- [1] M. Švejda, „Kinematika robotických architektur,“ 2011. [Online]. Available: http://home.zcu.cz/~msvejda/_publications/2011/rigo.pdf.
- [2] J. Čejka, „Rychlé heuristické metody numerického řešení úlohy inverzní kinematiky,“ 2019. [Online]. Available: https://dspace.tul.cz/bitstream/handle/15240/153383/Disertace_Cejka.pdf.
- [3] J. Shah, S.S. Rattan a B.C. Nakra, „Kinematic Analysis of 2-DOF Planer Robot Using Artificial Neural Network,“ *International Journal of Mechanical and Mechatronics Engineering*, sv. 5, č. 9, pp. 1720-1723, 2011.
- [4] J. Shah, S.S. Rattan a B.C. Nakra, „Kinematic Analysis of a Planer Robot Using Artificial Neural Network,“ *International Journal of Robotics and Automation (IJRA)*, sv. 1, č. 3, pp. 145-151, 2012.
- [5] V. Hlavac, „Neural Network for the identification of a functional dependence using data preselection,“ *Neural Network World*, sv. 2, pp. 109–124, 2021. Available: <http://nnw.cz/obsahy21.html>



Selected article from

Tento dokument byl publikován ve sborníku

**Nové metody a postupy v oblasti přístrojové
techniky, automatického řízení a informatiky 2021
New Methods and Practices in the Instrumentation,
Automatic Control and Informatics 2021
15. 9. – 17. 9. 2021, Žatec**

ISBN 978-80-01-06889-2

Web page of the original document:

<http://iat.fs.cvut.cz/nmp/2021.pdf>

Obsah čísla/individual articles:

<http://iat.fs.cvut.cz/nmp/2021/>

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6