

ARTIFICIAL NEURAL NETWORK APPLICATION TO WALK OF A FOUR LEGGED ROBOT

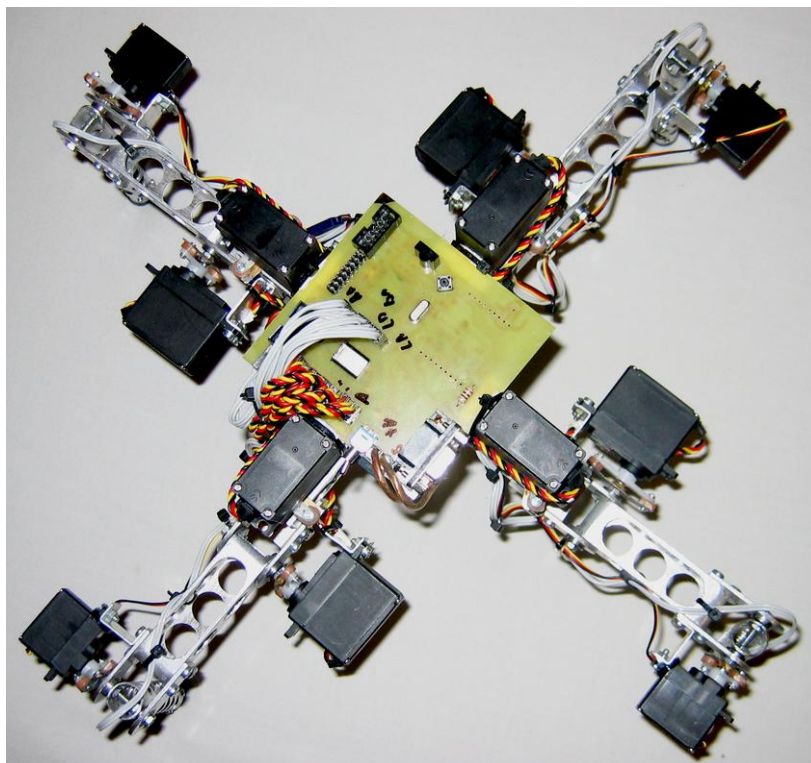
M. Bezdíček*, R. Grepl*, M. Švehlák*, J. Chmelíček*

Summary: *This paper presents simple method to extrapolate walk of a four legged robot. Movement of the robot is actuated by twelve servo-drives. This technique of walking is based on a few stable positions of robot body. To reach smooth movement, is sequence of positions extrapolated by artificial neural network. ANN is trained directly on values of servo angles. The servo angles are corresponding to the stable positions. Created neural network is able to extrapolate any posture for continuously changing input parameter. The input parameter is information about required position of the movement. Outputs of network are values of servo angles. Speed of the movement is dependent on the input parameter alternation rate. Introduced technique of walking is useful for plain surface moving.*

1. Úvod

V rámci řešení výzkumných projektů Laboratoře mechatroniky a robotiky se dlouhodobě zabýváme stavbou čtyřnohého kráčejícího robotu. Pro možnost rychlého ověření některých simulačních výpočtů a řídicích algoritmů jsme přikročili ke stavbě malého experimentálního robotu. Další popis je možno nalézt v příspěvcích [1,2,3] na této konferenci. Jde o čtyřnohého kráčivého robota, poháněného modelářskými servopohony, kde každá noha má tři stupně volnosti (obr. 1.1). Celkově se jedná o problém synchronního řízení dvanácti servopohonů. Pro vyrobeného robota bylo třeba vytvořit jednoduchý elegantní způsob chůze, na kterém by bylo možné dokázat, že robot je schopen se samostatně pohybovat, a který lze použít při pohybu ve snadno schůdném prostředí. Například, pohybuje-li se robot v prostředí kde není neočekáváno, že by musel překračovat nějaké překážky (kde je může obejít) postačí, bude-li se pohyb robotu skládat z předem pevně definovaných sekvencí pozic. S tímto přístupem pochopitelně není možné obsáhnout situaci, kdy se bude robot pohybovat v nerovném terénu a kdy je nutné vyhodnocovat jeho stabilitu. Touto problematikou se však tento příspěvek nezabývá. V článku je popsán návrh jednoduché chůze, která má minimální výpočetní nároky, je snadno implementovatelná přímo do mikrokontrolérů použitých na robotu a je univerzálně použitelná pro různé typy čtyřnohých robotů.

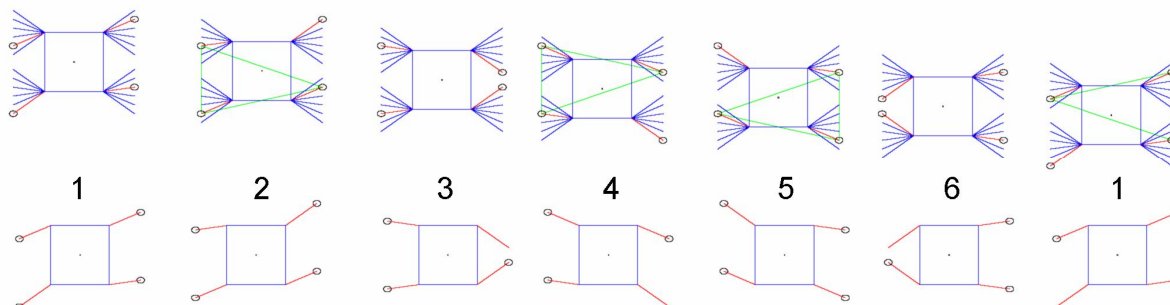
¹ * Ing. Milan Bezdíček, Ing. Robert Grepl, Ing. Jaroslav Chmelíček, Michal Švehlák:
Laboratoř mechatroniky a robotiky, ÚT AV ČR a ÚTMB FSI VUT v Brně, Technická 2,
616 69 Brno, e-mail: milannone@email.cz



Obr. 1.1 Fotografie čtyřnohého robotu

2. Základní skladba pohybů

Každý pohyb robotu se skládá ze sekvence předem pevně definovaných stabilních pozic. Pro přímočarou chůzi bylo zvoleno těchto výchozích pozic šest (obr. 2.1). Volba pozic vychází z úvahy, že robot musí být během chůze vždy ve stabilní poloze. Nebezpečí tedy hrozí pouze v případě, že robot stojí na třech nohách a čtvrtou má zvednutou. Posouzení stability v tomto případě vychází z předpokladu, že těžiště robotu musí být stále uvnitř trojúhelníka stability (obr. 2.1), který je definován koncovými body noh, na kterých robot právě stojí. Pro zvýšení stability (posunutí těžiště robotu co nejvíce do geometrického těžiště trojúhelníka) se během pohybu posouvá i celé tělo. Model přímočaré chůze je využitelný jak pro chůzi vpřed, tak i pro chůzi reverzní. Díky symetrii robotu je chůze použitelná také pro pohyb robotu do stran. Stejným způsobem jako při vytváření přímé chůze, byly vytvořeny i sekvence pozic pro rotaci robotu kolem jeho svislé osy a pro jeho vstávání.

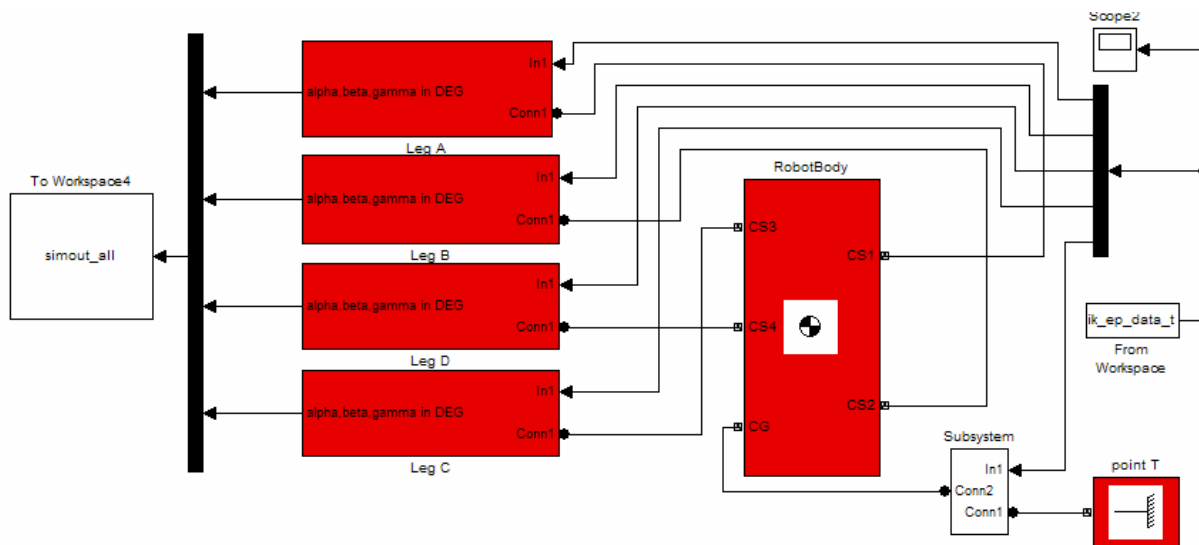


Obr. 2.1 Sekvence stabilních pozic pro přímou chůzi v jednom směru

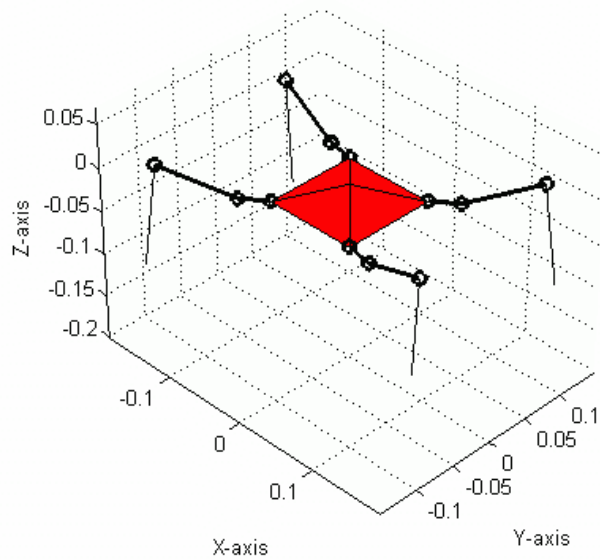
Algoritmus pro generování pozic těla v průběhu dané sekvence pohybu je vytvořen v Matlabu. Algoritmus je parametrický, proto je experimentování s různými druhy pohybů pro různé konstrukce čtyřnohých robotů velmi snadné. Výstupem algoritmu jsou souřadnice koncových bodů noh a poloha těla v globálním souřadném systému.

3. Řešení inverzní kinematiky

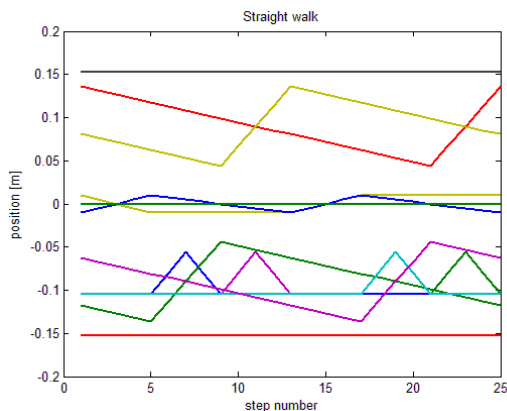
Pro vygenerovaná data (souřadnice koncových bodů noh a poloha těla) je třeba provést přepočítání těchto dat na úhly natočení servopohonů. Jedná se o úlohu inverzní kinematiky. K řešení je použit kinematický model v SimMechanics (obr. 3.1), vstupem modelu jsou pozice noh a poloha těla robotu. Výstupem jsou pak úhly natočení všech dvanácti servopohonů. Výhodou tohoto přístupu je, že není nutné sestavovat analytický IK model mechanismu. Model v SimMechanics je samozřejmě také parametrický a proto je jakákoli změna geometrie těla robotu velice snadná. Neboť se jedná pouze o úlohu kinematickou (ne dynamickou) je výpočet velice rychlý. Během simulace lze sledovat vizualizaci mechanismu (obr. 3.2).



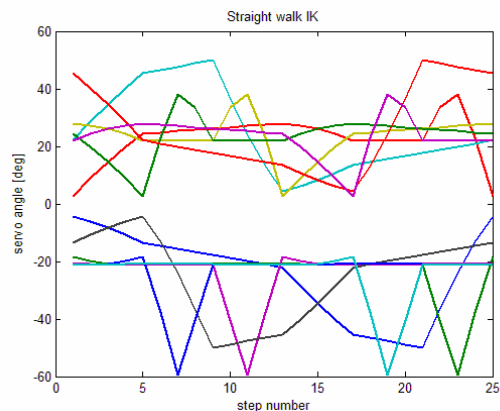
Obr. 3.1 Kinematický model v SimMechanics



Obr. 3.2 Vizualizace mechanismu v SimMechanics



Obr. 3.3 Souřadnice koncového bodu nohy pro přímočarou chůzi

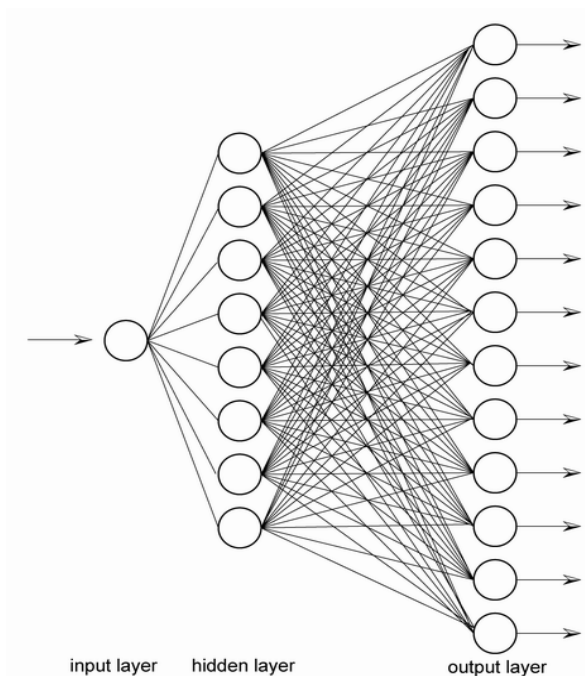


Obr. 3.4 Hodnoty natočení serv – výsledek IK transformace

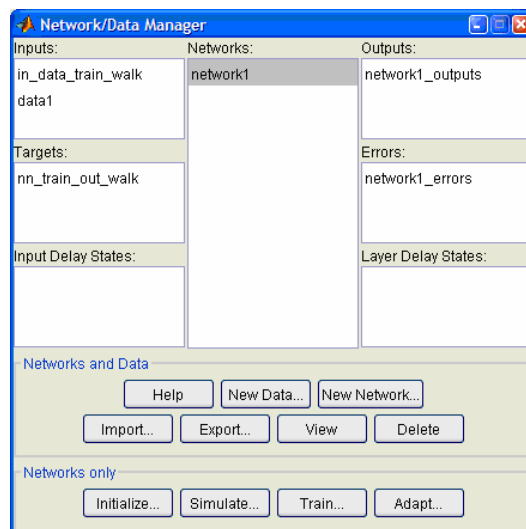
Na obrázcích 3.3 a 3.4 jsou znázorněny průběhy hodnot vstupujících do kinematického modelu (obr. 3.3) a hodnoty z modelu vystupující (obr. 3.4).

4. Umělá neuronová síť

Hodnoty získané z IK modelu jsou použity pro trénování umělé neuronové sítě. Je zřejmé, že v dané aplikaci musí mít síť dvanáct výstupních neuronů (dvanáct servopohonů). Experimentovat lze tedy pouze s počty vrstev a počty neuronů ve skrytých vrstvách. Během vývoje tohoto typu pohybu bylo vyzkoušeno několik druhů neuronových sítí s různým počtem vrstev, neuronů ve vrstvách a s různými topologiemi. Na závěr byla vybrána úplná, dopředná dvouvrstvá síť perceptronového typu, která má v první skryté vrstvě osm a ve výstupní dvanáct neuronů (obr. 4.1). K samotnému vytvoření a trénování umělé neuronové sítě je použit Neural Network Manager (obr. 4.2), který je součástí Matlabu.

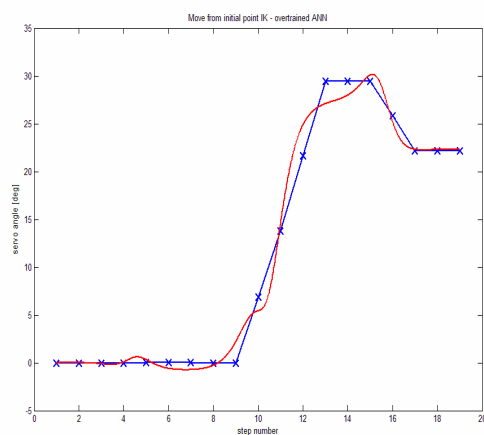


Obr. 4.1 Schéma použité neuronové sítě 8-12

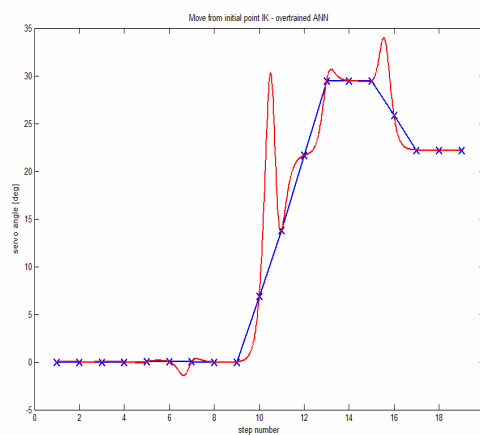


Obr. 4.2 Utilita Neural Network Manager v prostředí Matlabu

Při trénování sítě je třeba dbát především o to, aby nedošlo k „přetrénování“. K přetrénování dojde při přílišném snížení výstupní chyby neuronové sítě, tím síť ztratí schopnost generalizace. Jinými slovy, pokud naučíme neuronovou síť velmi přesně na daný vzor, pak tento bude rekonstruovat velmi přesně, ale ztratí schopnost rozpoznávat vzory i jen lehce odlišné (obr. 4.3, obr. 4.4).

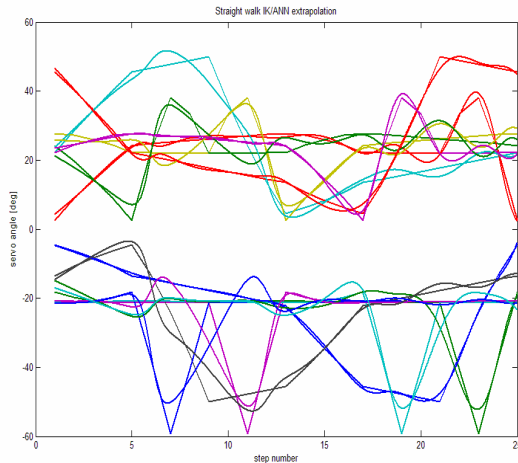


Obr. 4.3 Chování „dobře natrénované“ sítě

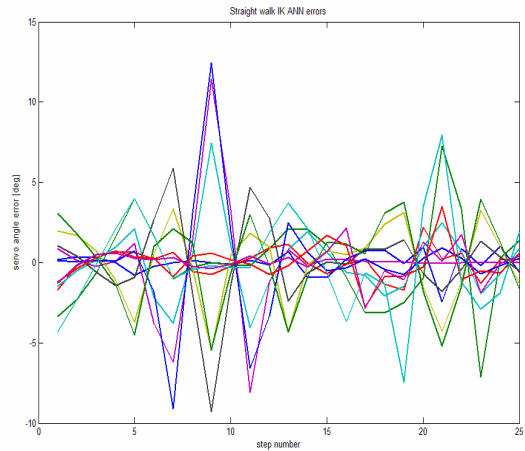


Obr. 4.4 Chování „přetrénované“ sítě

Vstupem neuronové sítě je v našem případě pouze jedna spojitá hodnota, která reprezentuje žádanou pozici robotu. Této vstupní hodnotě odpovídá dvanáct úhlů natočení na výstupu sítě. Na obrázku 4.5 jsou průběhy trénovacích hodnot a průběhy aproximované neuronovou sítí. Jak je vidět, tak odchylky mají pozitivní vliv na plynulost pohybů. Odchylky výsledků neuronové sítě a trénovacích dat jsou na obrázku 4.6.



Obr. 4.5 Porovnání hodnot získaných ze simulace s hodnotami použitými pro učení (pro přímou chůzi)

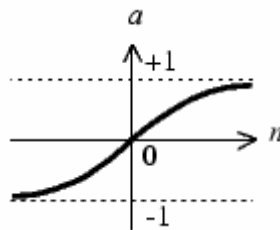


Obr. 4.6 Odchylky výsledků sítě a trénovacích dat

5. Implementace ANN do mikrokontroléru

Neuronová síť je navržena s co možná nejmenším počtem neuronů, se kterým je schopna extrapolovat požadované vzory. Snaha o co nejmenší počet neuronů vychází z předpokladu, že výpočet ANN bude probíhat přímo v mikrokontroléru robota a nikoli na osobním počítači. Jde o to, aby se mohl robot pohybovat autonomně a pouze v případě, že se vyskytne překážka, kterou nebude schopen obejít převezme řízení nadřazená úroveň, v našem případě osobní počítač. Výpočet ANN je nutné provádět s plovoucí aritmetikou, konkrétně se jedná o sčítání, násobení a výpočet sigmoidální funkce. Zde je použita funkce tansig, je definována rovnicí (1) a její průběh je na obrázku 5.1.

$$a = \frac{2}{(1 + e^{-2n})} - 1 \tag{1}$$



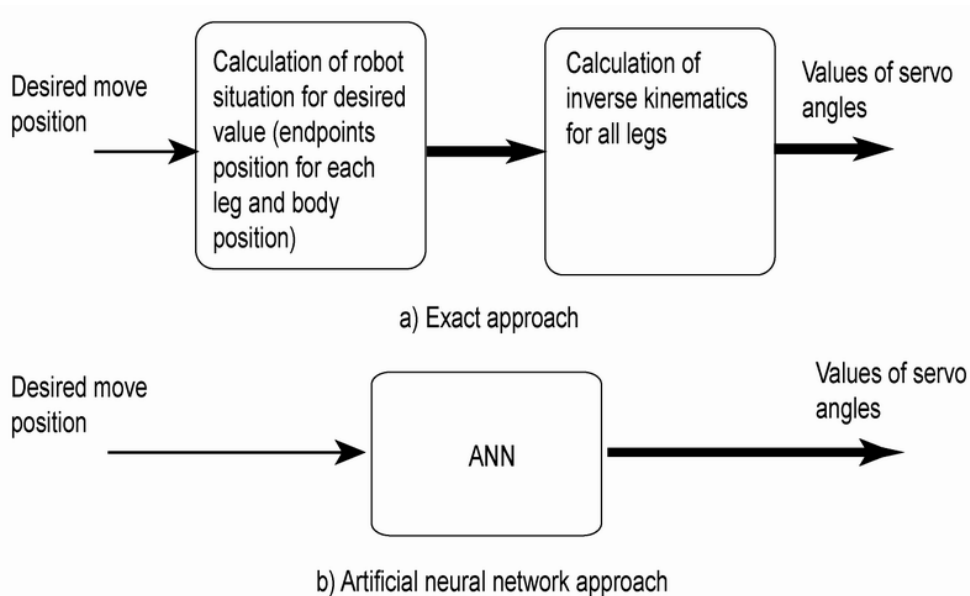
Obr. 5.1 Sigmoidální funkce „tansig“

Data natrénované sítě (hodnoty vah a prahů) jsou z Matlabu zkopírována do μC , kde je také naprogramován algoritmus neuronové sítě. Zdrojový kód je přeložen překladačem pro konkrétní mikrokontrolér. Tím je mikrokontrolér o firmy Atmel ATMEGA128, který pracuje na frekvenci 7.37 MHz. Řídící impulsy servopohonů mají periodu 20ms a z toho plyne, že pro zajištění plynulého pohybu je třeba provést celý výpočet právě za dobu kratší

než je oněch 20ms. Nejprve bylo pro výpočet použito funkce „exp()“, ale při tom trval výpočet sítě 72ms. Proto bylo přistoupeno k vytvoření funkce tansig tabulkou. Výhodou funkce tansig je, že je symetrická (obr. 5.1) a proto je možné tabulku zkrátit na polovinu. I s tímto zjednodušujícím předpokladem začala tabulková funkce dávat použitelné výsledky, až od tabulky obsahující pěti tisíc hodnot (to odpovídá objemu dat: $5000 \times 4B = 20kB$). Námí použitý mikrokontrolér má 128kB FLASH paměti programu, proto lze tuto oběť za rychlost výpočtu přijmout. Tabulkou zadaná funkce tansig se provede přibližně pětikrát rychleji než klasický výpočet. To znamená, že síť 8-12 se vypočte přibližně za 16ms, což je vyhovující.

6. Závěr

V prvních experimentech byla simulace neuronové sítě provedena pouze v Matlabu a výsledky v podobě tabulky nahrány do mikrokontroléru, velké nároky na paměť a neuspokojivé výsledky nás vedly k přímé implementaci ANN do mikrokontroléru. V této chvíli jsou pomocí umělých neuronových sítí realizovány pohyby přímé chůze, otáčení robota kolem jeho svislé osy, vstávání a samozřejmě přechody mezi jednotlivými pohyby. Neuronové sítě je použito i pro indikaci stavu nabití akumulátorů na těle robota. ANN v této aplikaci v sobě zahrnuje informace jak o postavení koncových bodů noh odpovídajících příslušným pozicím, tak i IK transformaci těchto hodnot (obr. 6.1). Tento příspěvek je nesnaží o nalezení řešení jež by pokrylo celou problematiku pohybu robota, ale jedná se pružnou metodu, kterou lze snadno rozvíjet i komplikovaný mechanismus.



Obr. 6.1 Mechanismus pro generování pozice robota

7. Poděkování

Práce vznikla za podpory pilotního projektu ÚT AV ČR č. 52020 „Realizace základních řídicích členů kráčivého robotu“ a projektu MŠMT MSM 262100024 „Výzkum a vývoj mechatronických soustav“.

8. Literatura

- [1] GREPL, R.,: *Control of experimental walking robot using simulating model*, Inženýrská mechanika 2004, Svatka, Česká republika, 2004
- [2] ŠVEHLÁK, M., : *Design of small laboratory quadruped robot*, Inženýrská mechanika 2004, Svatka, Česká republika, 2004
- [3] CHMELÍČEK, J., : *Collision states detection for quadruped robot motion*, Inženýrská mechanika 2004, Svatka, Česká republika, 2004
- [4] HUMUSOFT s.r.o., *Autorizovaný zástupce firmy The MathWorks, Inc.*, <http://www.humusoft.cz/matlab/simmech.htm>, 2003