U.P.B. Sci. Bull., Series D, Vol. 70, No. 2,2008 ISSN 1454-2358

A METHOD TO FIND DYNAMIC PARAMETERS FOR A LEG OF A HEXAPOD WALKING ROBOT

Jean-Claude HABUMUREMYI [1], Ioan DOROFTEI[2], Patrick KOOL[3], Yvan BAUDOIN [4]

În această lucrare, este propusă o metodă de determinare a parametrilor dinamici ai unui sistem mecanic, utilizând tehnici NeuroFuzzy. În multe strategii de control adaptiv este necesară cunoaşterea precisă sau estimarea acestor parametri. În scopul determinării lor, se începe cu colectarea unor date, prin experiment sau simulare, folosind arhitecturi ANFIS (Adaptive Neuro-Fuzzy Interference Systems). Apoi, utilizăm principii bazate pe proprietățile ecuțiilor care descriu modelul dinamic al robotului, pentru a determina parametrii dinamici. Pentru a verifica metoda propusă, se fac simulări pe sisteme ale căror modele dinamice sunt cunoscute.

A method to find dynamic parameters of a system using Neuro-Fuzzy techniques is proposed in this paper. In many model-based and adaptive control strategies, the precise or estimated knowledge of these parameters is required. In order to determine these parameters, we first begin by building an input/output mapping using ANFIS (Adaptive Neuro-Fuzzy Interference Systems) architecture based on input/output data pairs collected from experiment or simulation. Then, we use principles based on the properties of equation which described the dynamic model of robots in order to derive the dynamic parameters. Simulations on the systems where the mathematical dynamic models are well known demonstrate that proposed method is quite effective.

Key words: walking robot, pantograph mechanism, dynamic parameters, neuro-fuzzy

1. Introduction

There are different approaches to estimate the dynamic parameters of the systems in general and robots in particular. One solution is to dismantle the system and determine experimentally the mass, position of the center of mass, the moments and the products of inertia of the links. This method is very complex and time consuming. Next to that, we do not have information about the friction. Another approach is to estimate these parameters from CAD (Computer-Aided Design) models. With this approach, we encounter the same problem as previous. The approach that has been much applied is the identification using measurements

[1] Dr., Depart. of Applied Mechanics, Royal Military Academy of Brussels, BELGIUM
[2] Prof., Theory of Mechanisms and Robotics Depart., Gh. Asachi Technical University of Iasi, ROMANIA
[3] Prof., Depart. of mechanical Engineering, Vrije Universiteit Brussel, BELGIUM
[4] Prof., Depart. of Applied Mechanics, Royal Military Academy of Brussels, BELGIUM

of motion and actuation data. In the past, this problem has been well-studied [9, 10, 11]. Various techniques based on MLE (Maximum Likelihood Parameter Estimation), Levenberg-Marquardt method, LSE (Linear Least Square Estimation), Kalman observers, pseudo-inverses, etc., have been developed. Since the dynamic behaviors of the systems may be complicated due to varying environmental changes, the identification of their parameters using the mentioned techniques could be difficult. Soft-computing approaches such Neuro-Fuzzy techniques are the alternative for solving such complex problems.

In this paper, a method to find dynamic parameters using Neuro-Fuzzy techniques is proposed.

2. Generalized model and some properties for robot dynamics

Generally, the dynamic model of robots with n degrees of freedom is formulated as follow,

$$\tau = A(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + F(\theta,\dot{\theta}) + G(\theta) - J_T^F(\theta)F_{RF} \quad (1)$$

where:

$\tau = [\tau_1, \tau_2, \cdots, \tau_{n-1}, \tau_n]^T$ is the forces/torques vector;

$\theta = [\theta_1, \theta_2, \cdots, \theta_{n-1}, \theta_n]^T$ is the position coordinates vector;

$A(\theta)$ is the inertia matrix;

$C(\theta,\dot{\theta})$ are the centrifugal and Coriolis vectors

$F(\theta,\dot{\theta})$ are the friction forces on the joints;

$G(\theta)$ is the vector of the gravitational forces/torques;

$J_F^T(\theta)$ is a matrix 3 x m, which is the transpose of a Jacobian matrix;

F_{RF} is the m x 1 vector of the reaction forces that the ground exerts on the robot feet (F_{RF} is null during the transfer phase).

Equation (1) can be written in a compact way as follow:

$$\tau = A(\theta)\ddot{\theta} + H(\theta,\dot{\theta}), \quad (2)$$

with $H(\theta,\dot{\theta})$ being defined as a whole without distinguishing the differences among the different terms. It contains centrifugal, Coriolis, gravitational forces, viscous friction, coulomb friction and reaction forces terms. Some properties, used in this paper, of the robotic dynamics with n d.o.f. are:

- *Inertia matrix,* $A(\theta)$
  - It is symmetrical, i.e. $A(\theta) = A^T(\theta)$.
  - It is positive definite and bounded below and above, i.e. $\exists 0 < \alpha \leq \beta < \infty$, such that

$$\alpha I_n \leq A(\theta) \leq \beta I_n \quad \forall \theta \in \Re^n, \tag{3}$$

where I_n is the *n x n* identity matrix.

  - Its inverse $A^{-1}(\theta)$ exists, and is also positive definite and bounded,

$$\frac{1}{\beta} I_n \leq A^{-1}(\theta) \leq \frac{1}{\alpha} I_n. \tag{4}$$

- *Centrifugal and Coriolis forces,* $C(\theta, \dot{\theta})$
  - It is bilinear in $\dot{\theta}$.
  - It may be written in several factorizations, such as

$$C(\theta,\dot{\theta})\dot{\theta} = C_0(\theta,\dot{\theta}) = C_1(\theta) C_2 \left[\dot{\theta}\dot{\theta}\right] = C_3(\theta)\left[\dot{\theta}\dot{\theta}\right] + C_4(\theta)\left[\dot{\theta}^2\right], \tag{5}$$

where $\left[\dot{\theta}\dot{\theta}\right] = \left[\dot{\theta}_1\dot{\theta}_2 \ \dot{\theta}_2\dot{\theta}_3 \cdots \dot{\theta}_{n-1}\dot{\theta}_n\right]^T$ and $\left[\dot{\theta}^2\right] = \left[\dot{\theta}_1^2 \ \dot{\theta}_2^2 \cdots \dot{\theta}_n^2\right]^T$.

  - Given two *n*-dimensional vectors *x* and *y*, we have $C(\theta, x)y = C(\theta, y)x$.
- *Friction,* $F(\theta, \dot{\theta})$: Friction terms are complex and are described only approximately by a deterministic model [1, 2, 3]. The friction is present between any pair of surfaces having relative motion. Despite its complexity, one of the most important characteristics is that it is energy dissipative, i.e. $\dot{\theta}^T F(\theta, \dot{\theta}) \leq 0$.
- *Gravitational force,* $G(\theta)$
  - It can be derived from the gravitational potential energy function $\mathcal{P}(\theta)$, i.e. $G(\theta) = \frac{\partial \mathcal{P}}{\partial \theta}$.

- It is also bounded, i.e. $\left\| G(\theta) \right\| \le \gamma(\theta)$

where γ is a scalar function. For revolute joints, the bound is a constant independent of θ whereas for prismatic joints, the bound may depend on θ.

3. Principle to find dynamic parameters using Neuro-Fuzzy techniques

Experimentally on the real system (robot) or on its simulator, we collect from sensors (encoder, tachogenerator, potentiometer, …) following data sets,

$$\left\{ \theta_1(k),\ \theta_2(k),\cdots\theta_{n-1}(k),\ \theta_n(k),\ \tau_1(k),\ \tau_2(k),\cdots\tau_{n-1}(k),\ \tau_n(k) \right\}. \quad (6)$$

To collect these data, the trajectories should be well chosen. They will determine the accuracy of the dynamic parameters found. A lot of work has been done on this subject [13]. From these data, we constitute $\dot{\theta}_j(k)$ and $\ddot{\theta}_j(k)$ $\left(j = 1 \text{ to } n \right)$.

If data sets are collected from a simulator and have no noise, we can estimate the speed and the acceleration of the joints as follow:

$$\dot{\hat{\theta}}(k) = \frac{\theta_j(k+1) - \theta_j(k-1)}{2T}, \quad (7)$$

$$\ddot{\hat{\theta}}(k) = \frac{\theta_j(k+1) - 2\theta_j(k) + \theta_j(k-1)}{T^2}, \quad (8)$$

where T is a sampling time.

In reality, collected data have noises and we cannot use the differentiators as above because they are excessively sensitive to even small errors (they behave as high pass-filter). To solve this problem we accept the assumption that the speeds and the accelerations of the joints change a little during five consecutive observations. This assumption is practically valid because with actual microcontroller, the time of sampling is less than 1 *ms*. We also assume that these observations are near on a parabola of second order. Knowing $\theta_j(k-2)$, $\theta_j(k-1)$, $\theta_j(k)$, $\theta_j(k+1)$, $\theta_j(k+2)$ at respectively time $(k-2)T$, $(k-1)T$, kT, $(k+1)T$, $(k+2)T$, we would like to find *a*, *b* and *c* such that the errors of these observations data to the parabola $\theta_j = ax^2 + bx + c$ are minimal in the sense of least square. Solving this problem, we have:

$$a = \frac{2\theta_j(k-2) - \theta_j(k-1) - 2\theta_j(k) - \theta_j(k+1) + 2\theta_j(k+2)}{14T^2}, \tag{9}$$

$$b = \frac{(20k+14)\theta_j(k-2) - (10k-7)\theta_j(k-1) - 20k\theta_j(k) - (10k+7)\theta_j(k+1) + (20k-14)\theta_j(k+2)}{-70T}. \tag{10}$$

Because $\dot{\theta}_j(k) = 2akT + b$, we obtain by calculation

$$\dot{\theta}_j(k) = \frac{2\theta_j(k-2) + \theta_j(k-1) - \theta_j(k+1) - 2\theta_j(k+2)}{-10T}. \tag{11}$$

We can apply the same procedure to have the acceleration. Doing that, we will get:

$$\ddot{\theta}_j = \frac{2\dot{\theta}_j(k-2) + \dot{\theta}_j(k-1) - \dot{\theta}_j(k+1) - 2\dot{\theta}_j(k+2)}{-10T} = \tag{12}$$
$$= \frac{4\theta_j(k-4) + 4\theta_j(k-3) + \theta_j(k-2) - 4\theta_j(k-1) - 10\theta_j(k) - 4\theta_j(k+1) + \theta_j(k+2) + 4\theta_j(k+3) + 4\theta_j(k+4)}{100T^2}$$

Now, with the extended collected data (which include the velocities and accelerations of the joints), we can use these sets for mapping,

$$\theta_1(k),\ \theta_2(k), \cdots, \theta_n(k),\ \dot{\theta}_1(k),\ \dot{\theta}_2(k),\ \dot{\theta}_n(k), \cdots, \ddot{\theta}_1(k),\ \ddot{\theta}_2(k), \cdots \ddot{\theta}_n(k),$$

to $\tau_j(k)$ (where *j = 1* to *n*) in parallel identification model as shown in Figure 1.

ANFIS architecture is used for that. It has been proved that Mandani controllers as well as Sugeno controllers are universal approximators [4, 5]. ANFIS is one of the first fused Neuro-Fuzzy proposed by Jang [6]. It implements a Sugeno FIS (Fuzzy Interference System) and it has five layers as shown in Figures 1 and 2, for 2 input and 3 membership functions.

The first layer has adaptive nodes and it is used for the fuzzification of the input variables. The output of this layer will depend on the selected membership function which can be a triangular, a Gaussian, a trapezoidal, a bell-shaped, etc. Those parameters which specify a membership function will be called *premise parameters*. The output of the second layer is the product of all incoming signals and represents the firing strength of a rule. The product (it is more appropriate) has been chosen but it can be any T-norm operator that performs fuzzy AND. The

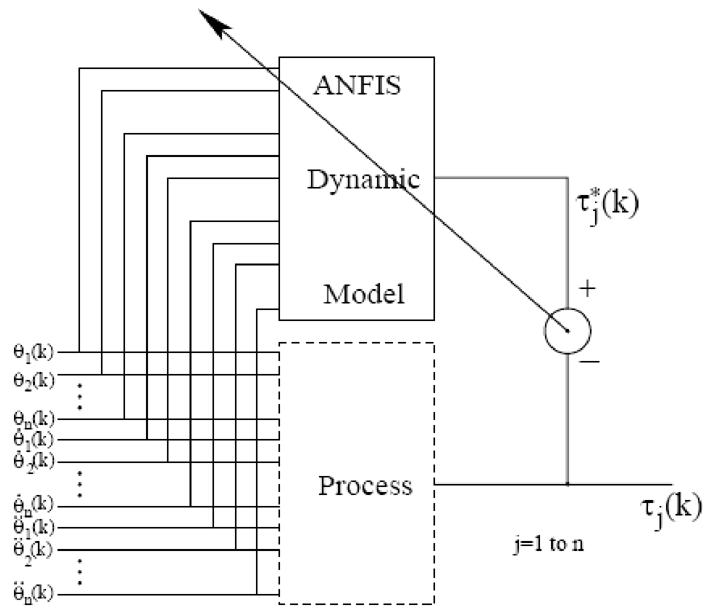

Fig. 1. Offline ANFIS parallel identification model

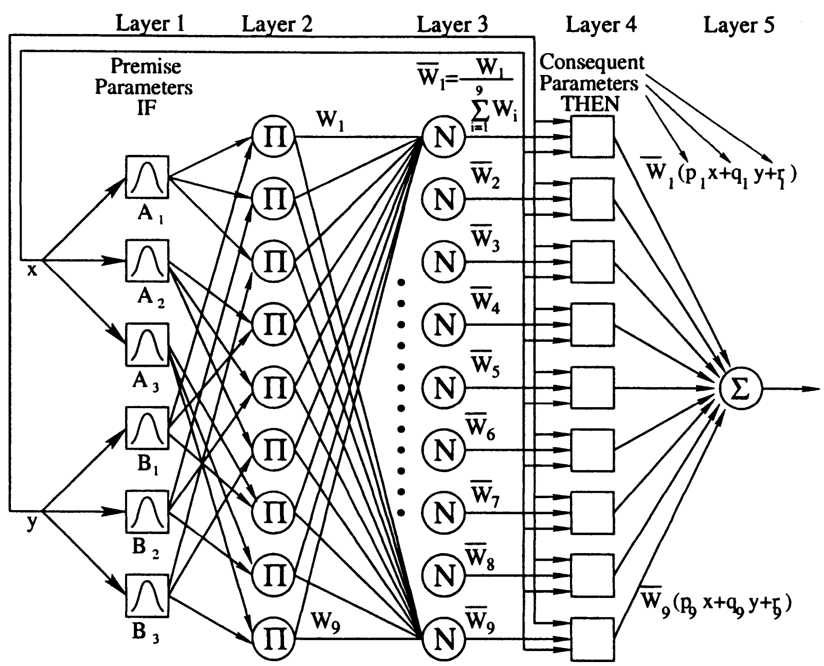

Fig. 2 ANFIS architecture

third layer normalizes the rule strengths. The i^{th} node is calculated as the ratio of the strength of the i^{th} rule to the sum of all firing strengths. The fourth layer has also adaptive nodes. The output of this layer for one nod will be the product between the normalized firing strength with a function. This function has parameters called *consequent parameters*. The fifth layer is the output. It computes the overall input as the summation of all incoming signals.

A step in the learning procedure (called hybrid learning) has got two parts [6]:

- The premise parameters are fixed and the input patterns are propagated up to fourth level of the layer. With the values of this level and the output patterns, optimal consequent parameters are calculated using an iterative least mean square procedure. The least mean square can be used because the output is a linear function of the consequent procedure.
- The patterns are propagated again with now, consequent parameters fixed. In this part back-propagation algorithm is used to modify the premise parameters.

After obtaining *n* ANFIS architectures (which express the mapping between data from the motion of the joints and the data of the torque of one joint), we carry out *n* + 1 experiments on them. We maintain the same position and speed of the joints but we change *n* + 1 times the accelerations. We have

$$\tau_{ij} = \sum_{k=1}^{n} A_{ik}\ddot{\theta}_{kj} + H\left(\theta,\dot{\theta}\right), \tag{13}$$

where $\ddot{\theta}_{kj}$ are the values of the torques and the accelerations on the joint *i* at the j^{th} experiment respectively. From the properties of the robotic dynamics, the inertia matrix is only dependent of the position (angular or linear position), i.e. by changing only the acceleration, the value of A_{ik} will not change and $H\left(\theta,\dot{\theta}\right)$ will remain the same. Subtracting relation (13) obtained on the same joint but at different experimentations, we have:

$$\tau_{i1} - \tau_{i(j+1)} = \sum_{k=1}^{n}\left(\ddot{\theta}_{k1} - \ddot{\theta}_{k(j+1)}\right)A_{ik} \tag{14}$$

or

$$\Delta\tau_i = \Delta\ddot{\theta}\ A_i\,, \tag{15}$$

where

$$\Delta\tau_i = \begin{pmatrix} \Delta\tau_{i1} - \Delta\tau_{i2} \\ \Delta\tau_{i1} - \Delta\tau_{i3} \\ \vdots \\ \Delta\tau_{i1} - \Delta\tau_{in} \\ \Delta\tau_{i1} - \Delta\tau_{i(n+1)} \end{pmatrix}, \ \Delta\ddot{\theta} = \begin{pmatrix} \ddot{\theta}_{11} - \ddot{\theta}_{12} & \ddot{\theta}_{21} - \ddot{\theta}_{22} & \cdots & \ddot{\theta}_{n1} - \ddot{\theta}_{n2} \\ \ddot{\theta}_{11} - \ddot{\theta}_{13} & \ddot{\theta}_{21} - \ddot{\theta}_{23} & \cdots & \ddot{\theta}_{n1} - \ddot{\theta}_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ \ddot{\theta}_{11} - \ddot{\theta}_{1n} & \ddot{\theta}_{21} - \ddot{\theta}_{2n} & \cdots & \ddot{\theta}_{n1} - \ddot{\theta}_{nn} \\ \ddot{\theta}_{11} - \ddot{\theta}_{1(n+1)} & \ddot{\theta}_{21} - \ddot{\theta}_{2(n+1)} & \cdots & \ddot{\theta}_{n1} - \ddot{\theta}_{n(n+1)} \end{pmatrix}$$

and $A_i = \begin{bmatrix} A_{i1} & A_{i2} & \cdots & A_{i(n-1)} & A_{in} \end{bmatrix}^T$.

We can then calculate A_i if $\Delta\ddot{\theta}$ is invertible as follow:

$$A_i = \Delta\ddot{\theta}^{-1}\Delta\tau_i . \tag{16}$$

Finally we deduced $H_i\left(\theta,\dot{\theta}\right)$ from the equation (13). We can use the property which stipulates that the inertia matrix is symmetrical to check the validity of the values of its elements obtained on different joints and experiments (is A_{ij} equal to A_{ji}?).

4. Application of the method

4.1. Identification of the dynamic model and parameters estimation of a two link planar arm

In order to illustrate the method quoted above, we have applied it to a two link planar arm, shown in Figure 3.

The parameters of its dynamic model are well-known [14]. They will enable us to establish comparisons with those obtained by means of the Neuro-Fuzzy model. By neglecting the friction torques and the tip contact forces, the model of the two link planar arm is as follows [14],

$$\begin{aligned} \tau_1 &= A_{11}\ddot{\theta}_1 + A_{12}\ddot{\theta}_2 + H_1 \\ \tau_2 &= A_{21}\ddot{\theta}_1 + A_{22}\ddot{\theta}_2 + H_2 \end{aligned}, \tag{17}$$

where

$$A_{11} = I_{l_1} + m_{l_1} l_1^2 + k_{r1}^2 I_{m_1} + I_{l_2} + m_{l_2}\left(a_1^2 + l_2^2 + 2a_1 l_2 \cos\theta_2\right) + I_{m_2} + m_{m_2} a_1^2 ;$$

$$A_{12} = A_{21} = I_{l_2} + m_{l_2}\left(l_2^2 + a_1 l_2 \cos\theta_2\right) + k_{r2} I_{m_2} ;$$

$$A_{22} = I_{l_2} + m_{l_2} l_2^2 + k_{r2} I_{m_2};$$

$$H_1 = -2m_{l_2} a_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin\theta_2 - m_{l_2} a_1 l_2 \dot{\theta}_2^2 \sin\theta_2 + \left(m_{l_1} l_1 + m_{m_2} a_1 + m_{l_2} a_1\right) g \cos\theta_1 + \\ + m_{l_2} l_2 g \cos\left(\theta_1 + \theta_2\right);$$

$$H_2 = m_{l_2} a_1 l_2 \dot{\theta}_1^2 \sin\theta_2 - m_{l_2} l_2 g \cos\left(\theta_1 + \theta_2\right);$$

l_1, l_2 are the distances of centers of mass for the two links from the respective joint axes;

m_{l_1}, m_{l_2} are the masses of the two links;

m_{m_1}, m_{m_2} are the masses of the rotors for the two joints motors;

I_{m_1}, I_{m_2} are the moments of inertia with respect to the axes of the two joints;

k_{r1}, k_{r2} are the reduction ratio of the gears;

I_{l_1}, I_{l_2} are the moments of inertia relative to the centers of mass for the two links.

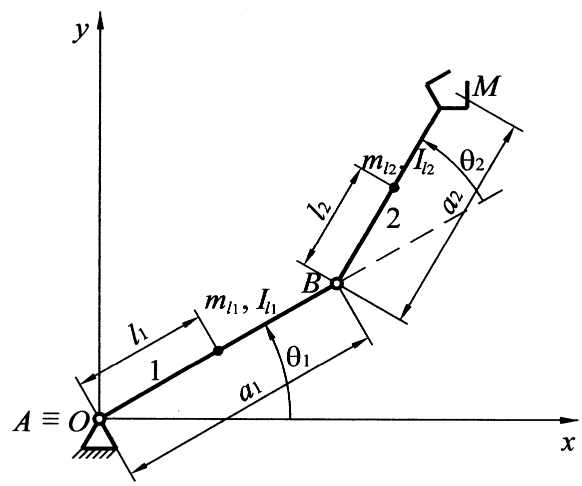

Fig. 3. Two link planar arm

In order to have numerical values, we use the data from [14]:

$a_1 = a_2 = 1m;\ l_1 = l_2 = 0.5m;\ m_{l_1} = m_{l_2} = 50\text{kg};\ I_{l_1} = I_{l_2} = 10\text{kg}\cdot\text{m}^2;$

$k_{r1} = k_{r2} = 100;\ m_{m_1} = m_{m_2} = 5\text{kg};\ I_{m_1} = I_{m_2} = 0.01\text{kg}\cdot\text{m}^2.$

From equation (17), we have collected data sets,

$$\left\{\theta_1(k),\ \dot{\theta}_1(k),\ \ddot{\theta}_1(k),\ \theta_2(k),\ \dot{\theta}_2(k),\ \ddot{\theta}_2(k),\ \tau_1(k)\right\}$$
$$\left\{\theta_1(k),\ \dot{\theta}_1(k),\ \ddot{\theta}_1(k),\ \theta_2(k),\ \dot{\theta}_2(k),\ \ddot{\theta}_2(k),\ \tau_2(k)\right\},$$

with $\theta_1(k)$ and $\theta_2(k)$ in the range of 0 to $\frac{\pi}{3}$ and k going from 1 to 4096 (the range and the number of data were taken to reduce the offline learning calculation time). We have made trials to determine the appropriate number of membership functions and the type of FIS by considering the final RMSE (Root Mean Square Error). We have found from different trials that two triangular membership functions (*N* and *P*) by input and first-order Sugeno FIS give a small error.

After nearly 409600 iterations for each data sets, we have obtained a RMSE of about 0.0879 for τ_1 and 0.0862 for τ_2.

From the ANFIS models of the two link planar arm, we have solved the equations (15), to estimate its dynamic parameters. Some results are shown in Table 1 (we have chosen for illustration the variation of θ_2 because the matrix *A* is only dependent on this parameter). The $\tilde{A}_{11}$, $\tilde{A}_{12}(1)$, $\tilde{A}_{12}(2)$, $\tilde{A}_{22}$ parameters are estimated using our method and A_{11}, A_{12}, A_{22} are computed using the equations from [14].

Table 1

Dynamic parameters of the two link planar arm

θ_2	$\tilde{A}_{11}$	A_{11}	$\tilde{A}_{12}(1)$	$\tilde{A}_{12}(2)$	A_{12}	$\tilde{A}_{22}$	A_{22}
0.1	249.8	249.76	48.408	48.303	48.375	122.53	122.5
0.2	249.03	249.01	48.022	47.98	48.002	122.52	122.5
0.3	247.78	247.78	47.39	47.38	47.383	122.51	122.5
0.4	246.05	246.06	46.525	46.524	46.527	122.5	122.5
0.5	243.88	243.89	45.435	45.432	45.44	122.49	122.5
0.6	241.27	241.28	44.131	44.122	44.133	122.48	122.5
0.7	238.26	238.25	42.622	42.61	42.621	122.47	122.5
0.8	234.85	234.85	40.916	40.91	40.918	122.46	122.5
0.9	231.06	231.09	39.022	39.048	39.04	122.45	122.5

4.2. Identification of the dynamic model and parameters estimation of the leg of AMRU5 robot

The pantograph mechanism has been used in many legged robots projects (TITAN III, TITAN IV, RIMHO, …). Opposed to other mechanisms, the pantograph mechanism exhibits the most potentiality for the following reasons [7, 8]:

- An exact-straight-line foot trajectory can be obtained by actuating only one linear actuator. A high efficiency is to be expected.
- The leg geometry can be made more compact by adjusting the magnification ratio.
- The input motions can be mechanically decoupled. The simplest approach to the foot trajectory control can be used.
- The closed-loop gives a good rigidity of the leg.

As disadvantage of this mechanism, the linear actuating systems are more difficult to design and to protect if normal electric motors are used. Also when using model-based controller, it is not easy to find the parameters of the leg due to the closed-loop. The first problem leads often to the choice of a 2D pantograph mechanism to make the design more compact. In this case, the GDA (Gravitational Decoupled Actuation) is not perfect. A 2D pantograph (Figure 4) has been used on the robots MECANT, BOADICEA and the robot AMRU5 (Figure 5) that we have designed.

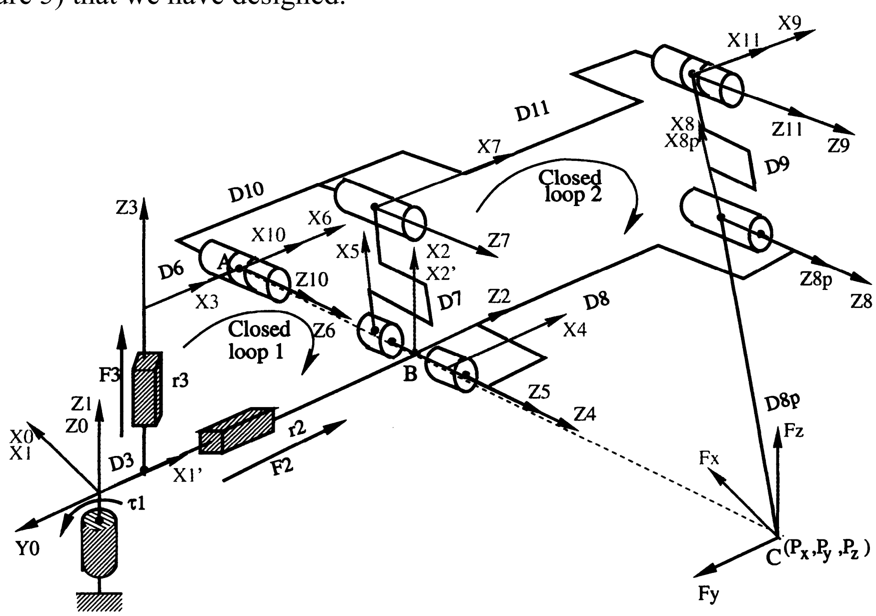

Fig. 4. Leg structure based on a 2D pantograph mechanism

The second problem has been solved using the method explained above. The pantograph mechanism has 2 closed loops and it is not an easy task to derive the dynamic model and the parameters of such mechanism using classical method (Newton-Euler formalism). That is why we have preferred to use Soft-Computing methodology to identify those parameters, which have been used in the control strategy of the robot. ANFIS has been proved to have the best performance in comparison to others methods: FNN (Feedforward Neural Networks Architecture), RBFNN (Radial Basis Function Neural Networks, RKNN (Runge-Kutta Neural Networks), [12].

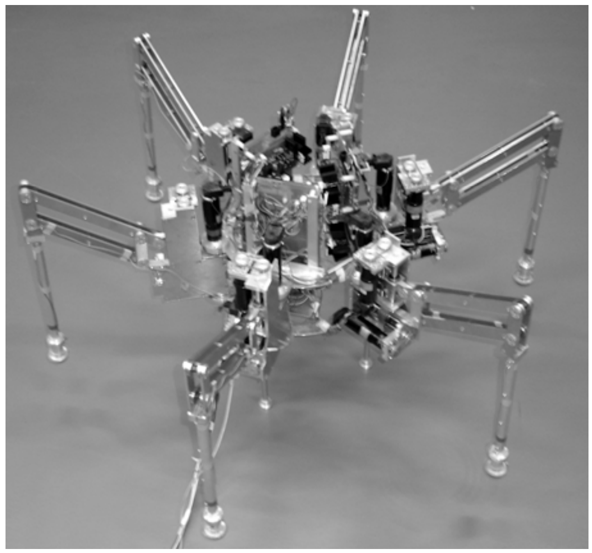

Fig. 5. AMRU5 walking robot

Due to the pantograph mechanism, the actuators responsible of the translation movement *r*3 will only be used to support the body of the robot against gravity forces and in transfer phase. The actuators that generate the translation movement *r*2 and the rotation θ are used in the tracking of the trajectory when the legs are in the stance phase. As we have a decoupling of *r*3 from *r*2 and θ_1, equation (2) can be split in two parts as follows:

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + \begin{pmatrix} H_1(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \\ H_2(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) \end{pmatrix}, \tag{18}$$

$$\tau_3 = A_3 \ddot{\theta}_3 + H_3\left(\theta_3, \dot{\theta}_3\right). \tag{19}$$

We will only consider the system of equations (18) in this paper, to explain that ANFIS joints control (equation (19) is a particular case). Equation (18) will be written as:

$$\mathcal{T} = \mathcal{A}(\theta)\ddot{\theta} + \mathcal{H}(\theta,\dot{\theta}), \tag{20}$$

where $\mathcal{T} = [\tau_1, \tau_2]^T$, $\mathcal{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $\mathcal{H} = [H_1, H_2]^T$ and $\theta = [\theta_1, \theta_2]^T$.

To identify the parameters of the dynamic model described by equation (20), we need its equivalent discrete-time version defined by nonlinear difference equations. To approximate $\dot{\theta}$ and $\ddot{\theta}$, we will use a Taylor series as follows:

$$\theta(k+1) = \theta(k) + \Delta t\dot{\theta}(k) + \frac{\Delta t^2}{2}\ddot{\theta}(k) + O(\Delta t^3), \tag{21}$$

$$\theta(k-1) = \theta(k) - \Delta t\dot{\theta}(k) + \frac{\Delta t^2}{2}\ddot{\theta}(k) + O(\Delta t^3). \tag{22}$$

Equation (21) plus equation (22) gives:

$$\ddot{\tilde{\theta}}(k) = \frac{\theta(k+1) - 2\theta(k) + \theta(k-1)}{\Delta t^2}. \tag{23}$$

Equation (21) minus equation (22) gives:

$$\dot{\tilde{\theta}}(k) = \frac{\theta(k+1) - \theta(k-1)}{2\Delta t}. \tag{24}$$

where Δt is a sampling time.

Equation (20) becomes in discrete-time:

$$\mathcal{T}(k) = \mathcal{A}(\theta(k))\ddot{\theta}(k) + \mathcal{H}(\theta(k),\dot{\theta}(k)). \tag{25}$$

where $\ddot{\theta}(k)$ and $\dot{\theta}(k)$ are expressed by the equations (23) and (24) respectively.

It is necessary to estimate the elements of the inertia matrix $\mathcal{A}(\theta(k))$ and the elements of the matrix $\mathcal{H}(\theta(k),\dot{\theta}(k))$ because they will be used in the strategy control. To estimate those elements, we use the principle that the elements of the inertia matrix are only dependent on $\theta(k)$ and the elements of the matrix $\mathcal{H}$ are dependent on $\theta(k)$ and $\dot{\theta}(k)$, i.e. if we change $\ddot{\theta}(k)$ and we maintain $\theta(k)$ and $\dot{\theta}(k)$ with the same values, the elements of the inertia matrix will remain the same. Suppose we have the same $\theta(k)$, $\dot{\theta}(k)$ for three different angular accelerations $\ddot{\theta}_j(k)$ ($j=a,b$ and c), then

$$\tau_j(k)=\mathcal{A}(\theta(k))\ddot{\theta}_j(k)+\mathcal{H}(\theta(k),\dot{\theta}(k)), \tag{26}$$

where $\tau_j(k)=\left[\tau_{1j}(k)\ \ \tau_{2j}(k)\right]^T$ and $\ddot{\theta}_j(k)=\left[\ddot{\theta}_{1j}(k)\ \ \ddot{\theta}_{2j}(k)\right]^T$.

By applying the principle stated above, we have:

$$\begin{aligned}\tau_{1a}(k)-\tau_{1b}(k)&=A_{11}\left(\ddot{\theta}_{1a}(k)-\ddot{\theta}_{1b}(k)\right)+A_{12}\left(\ddot{\theta}_{2a}(k)-\ddot{\theta}_{2b}(k)\right)\\ \tau_{1a}(k)-\tau_{1c}(k)&=A_{11}\left(\ddot{\theta}_{1a}(k)-\ddot{\theta}_{1c}(k)\right)+A_{12}\left(\ddot{\theta}_{2a}(k)-\ddot{\theta}_{2c}(k)\right)\end{aligned}, \tag{27}$$

$$\begin{aligned}\tau_{2a}(k)-\tau_{2b}(k)&=A_{21}\left(\ddot{\theta}_{1a}(k)-\ddot{\theta}_{1b}(k)\right)+A_{22}\left(\ddot{\theta}_{2a}(k)-\ddot{\theta}_{2b}(k)\right)\\ \tau_{2a}(k)-\tau_{2c}(k)&=A_{21}\left(\ddot{\theta}_{1a}(k)-\ddot{\theta}_{1c}(k)\right)+A_{22}\left(\ddot{\theta}_{2a}(k)-\ddot{\theta}_{2c}(k)\right)\end{aligned}. \tag{28}$$

Solving the systems of equations (27) and (28) we will get:

$$A_{11}=\frac{\left(\ddot{\theta}_{2c}-\ddot{\theta}_{2b}\right)\tau_{1a}+\left(\ddot{\theta}_{2a}-\ddot{\theta}_{2c}\right)\tau_{1b}+\left(\ddot{\theta}_{2b}-\ddot{\theta}_{2a}\right)\tau_{1c}}{\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1c}\right)\ddot{\theta}_{2a}+\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1a}\right)\ddot{\theta}_{2b}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1b}\right)\ddot{\theta}_{2c}}, \tag{29}$$

$$\left(A_{12}\right)_1=A_{12}=-\frac{\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1b}\right)\tau_{1a}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1c}\right)\tau_{1b}+\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1a}\right)\tau_{1c}}{\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1c}\right)\ddot{\theta}_{2a}+\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1a}\right)\ddot{\theta}_{2b}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1b}\right)\ddot{\theta}_{2c}}, \tag{30}$$

$$\left(A_{12}\right)_2=A_{21}=\frac{\left(\ddot{\theta}_{2c}-\ddot{\theta}_{2b}\right)\tau_{2a}+\left(\ddot{\theta}_{2a}-\ddot{\theta}_{2c}\right)\tau_{2b}+\left(\ddot{\theta}_{2b}-\ddot{\theta}_{2a}\right)\tau_{2c}}{\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1c}\right)\ddot{\theta}_{2a}+\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1a}\right)\ddot{\theta}_{2b}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1b}\right)\ddot{\theta}_{2c}}, \tag{31}$$

$$A_{22}=-\frac{\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1b}\right)\tau_{2a}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1c}\right)\tau_{2b}+\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1a}\right)\tau_{2c}}{\left(\ddot{\theta}_{1b}-\ddot{\theta}_{1c}\right)\ddot{\theta}_{2a}+\left(\ddot{\theta}_{1c}-\ddot{\theta}_{1a}\right)\ddot{\theta}_{2b}+\left(\ddot{\theta}_{1a}-\ddot{\theta}_{1b}\right)\ddot{\theta}_{2c}}. \tag{32}$$

The difference between $\left(A_{12}\right)_1$ and $\left(A_{12}\right)_2$ is very small if the two data sets $\left\{\theta_1(k),\ \dot{\theta}_1(k),\ \ddot{\theta}_1(k),\ \theta_2(k),\ \dot{\theta}_2(k),\ \ddot{\theta}_2(k),\ \tau_1(k)\right\}$ and $\left\{\theta_1(k),\ \dot{\theta}_1(k),\ \ddot{\theta}_1(k),\ \theta_2(k),\ \dot{\theta}_2(k),\ \ddot{\theta}_2(k),\ \tau_2(k)\right\}$ constituted converge to a small RMSE. It could be that one of the sets has a small error. If it is the case, we could use the property that the inertia matrix is symmetrical and the ANFIS model which has the smallest error, to find some parameters of the dynamic model. If, for example, the second data gives a small RMSE error, we will use the equations (31) and (32) to find A_{12} ($A_{21} = A_{12}$) and A_{22}. We can then calculate A_{11} from two different angular accelerations as follows:

$$A_{11} = \frac{\tau_{1a} - \tau_{1b} - A_{12}\left(\ddot{\theta}_{2a} - \ddot{\theta}_{2b}\right)}{\left(\ddot{\theta}_{1a} - \ddot{\theta}_{1b}\right)}. \tag{33}$$

After the determination of the matrix $\mathcal{A}\left(\theta(k)\right)$, we can determine the matrix $\mathcal{H}$ with the following equations:

$$H_1\left(\theta(k), \dot{\theta}(k)\right) = \tau_{1j} - A_{11}\ddot{\theta}_{1j} - A_{12}\ddot{\theta}_{2j} \tag{34}$$

$$H_2\left(\theta(k), \dot{\theta}(k)\right) = \tau_{2j} - A_{21}\ddot{\theta}_{1j} - A_{22}\ddot{\theta}_{2j} \tag{35}$$

where *j* could be *a*, *b* or *c*. Here also we can compare the different results obtained with different *j*.

The first step to find the dynamic parameters of the AMRU5 leg was the design of an initial FLC (Fuzzy Logic Controller). With this initial FLC, the leg has been moved randomly and we have collected data sets necessary to train the ANFIS in the purpose of having the dynamic model of the two considered joints. After having used those data in the training of ANFIS architectures, we have obtained a RMSE of about 0.52 for the joint 1 and about 6.01 for the joint 2. The resulted elements of the inertia matrix are shown in Figures 6-9.

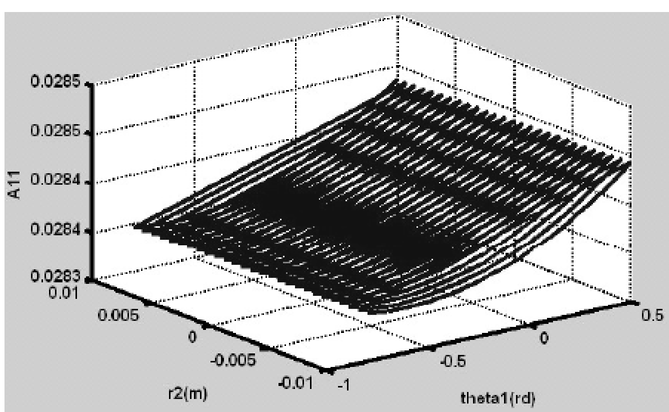

Fig. 6. Element A_{11} of the inertia matrix

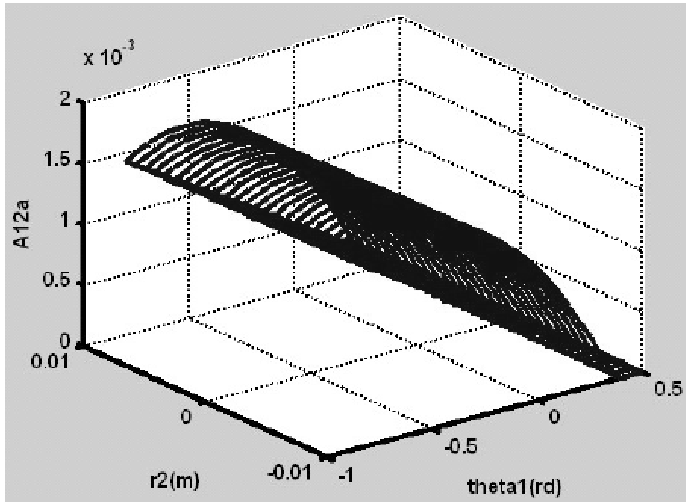

Fig. 7. Element $(A_{12})_1$ of the inertia matrix (first method)

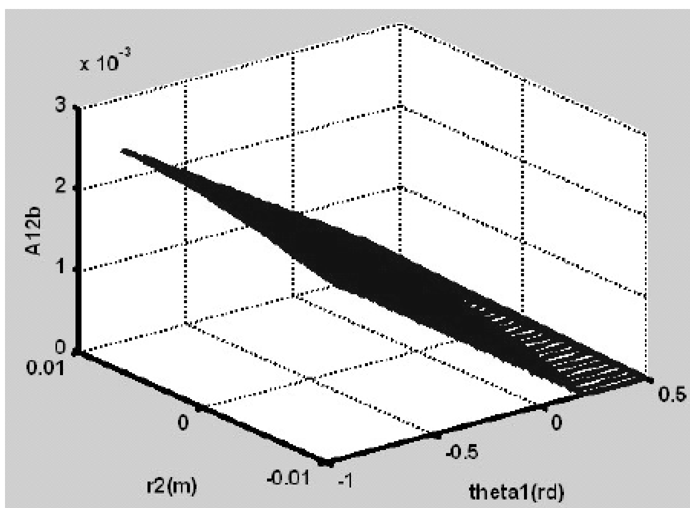

Fig. 8. Element $(A_{12})_2$ of the inertia matrix (second method)

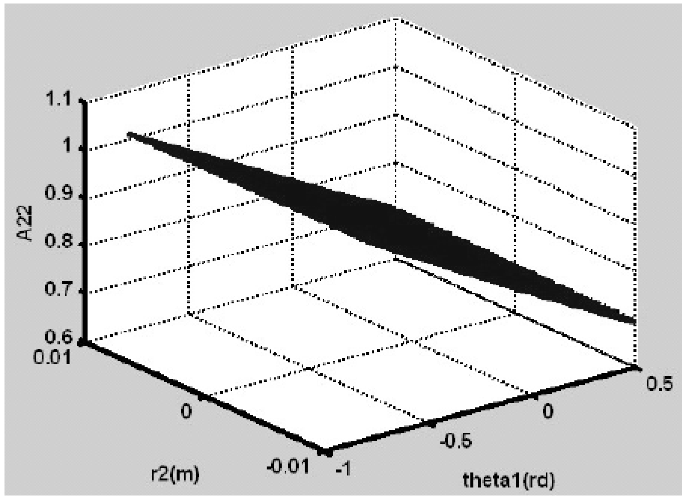

Fig. 9. Element A_{22} of the inertia matrix

5. Conclusion

In the framework of HUDEM project (Belgian Ministry of Defense), a hexapod walking robot with hexagonal architecture has been built. The architecture of the robot has been chosen on the basis of the existing robots survey.

In this paper, we have shown the principle to identify the dynamic parameters of a mechanical system using Soft-Computing methodology. We firstly begin by collecting motion and actuation data from well-selected trajectories of the joints. The assumption saying that the variations of the velocities and accelerations of the joints during 5 consecutive observations are small, allow us to derive these parameters from the measured positions (which include generally noises). This assumption is practically acceptable in several parts of the selected trajectories. Then we use ANFIS architectures in the mapping of motion data of the joints to actuation data of one joint. Finally, to identify the elements of the inertia matrix, we apply on the ANFIS architectures obtained, the principle that the elements of the inertia matrix are only dependent on the joints position. This method has been tested on a two link planar manipulator because we have a mathematical model of it. The comparison between the outputs of the method and the mathematical model proves the validity of it. Then, we have been used this method to identify the dynamic parameters of a pantograph-based leg of AMRU5 walking robot. The use of the parameters obtained in a model-based adaptive controller developed on a leg simulator has been shown the efficiency of the proposed method.

R E F E R E N C E S

[1]. *I. D. Landau, R. Horowitz*, Synthesis of adaptive controllers for robot manipulators using passive feedback system approach, in Proceedings IEEE Int. Conf. Robotics and Automation, Philadelphia, PA, pp. 1028-1033, Apr. 1998.

[2]. *C. Canudas de Wit, H. Olsson, K. J. Astrom, P. Lishinsky*, A new model for control of systems with friction, IEEE Trans. Automatic Control, **vol. 40**, no. 3, pp. 419-425, 1995.

[3]. *C. Canudas de Wit, S. S. Ge*, Adaptive friction compensation for systems with generalized velocity/position friction dependency, Proc. IEEE Conf. Descision and Control, 1997.

[4]. *B. Koskp*, Fuzzy Systems as Universal Approximators, Proc. IEEE Int. Conf. On Fuzzy Systems, San Diego, CA, pp. 1153-1162, 1992.

[5.] *J. J. Buckley*, Sugeno Type Controllers Are Universal Controllers, Fuzzy Sets and Systems, 53:299-303, 1993.

[6]. *J. S. R. Jang*, ANFIS, Adaptiv- network-based fuzzy interference systems, IEEE Transactions of Systems, Man and Cybernetics, 1992, 23(3), pp. 665-685.

[7]. *S. M. Song, K. J. Waldron*, Machine that Walk: The Adaptive Suspension Vehicle, The MIT Press, London, England, 1989.

[8]. *I. Doroftei*, Selected Architectures of Walking Robots: State-of-the-Art Review, Report no. 1, Universite Libre de Bruxelles, SCMERO, Apr. 1999.

[9]. *C. G. Atkeson, C. H. An, J. M. Hollebach*, Estimation of inertial parameters of manipulator loads and links, Int. Journal of Robotics Research, **vol. V**, no. 3, 1986.

[10]. *H. B. Olsen, G. A. Beckey*, Identification of parameters in models of robots with rotary joints, Proc. Of IEEE Int. Conf. on Robotics and Automation, St. Louis, march 25-28, 1985.

[11]. *R. Serban, J. S. Freeman*, Identification and identifiability of unknown parameters in multibody dynamic systems, Multibody Systems Dynamics, Kluwer Academic Publishers, **vol. 5**, no. 4, pp. 335-350, May 2001.

[12]. *M. Onder Efe, O. Kaynak*, A Comparative Study of oft Computing Methodologies in Identification of Robotic Manipulators, Robotics and Autonomous Systems, **vol. 30**, no. 221-230, 2000.

[13]. *P. K. Khosla*, Real-Time Control and Identification of Direct-Drive Manipulators, Department of Electrical and Computer Engineering, Carnegie-Mellon University, August 1986.

[14]. *L. Sciavicco, B. Siciliano*, Modelling and Control of Robot Manipulators, Springer-Verlag Edition, London, 2000.