Artificial Neural Network-based Hybrid Force/Position Control of an Assembly Task

Y. Touati, Y. Amirat and N. Saadia, *Member, IEEE*

***Abstract*—In the case of complex robotics tasks, pure position control will be ineffective since forces appearing during the contacts must also be controlled. However, simultaneous position and force control called hybrid control is then required. Moreover, the non-linear plant dynamics, the complexity of the dynamic parameters determination and computation constraints makes more difficult the synthesis of control laws. In order to satisfy all these constraints, an effective hybrid force/position approach based on Artificial Neural Networks for MIMO systems is proposed. This approach realizes, simultaneously, an identification and control, and it is implemented according to two phases: At first, a neural observer is trained off line on the basis of the data acquired during contact motion, in order to realize a smooth transition from free to contact motion. Then, an online learning of the neural controller is implemented using neural observer parameters so that the closed-loop system maintains a good performance. A typical example on which we shall focus is an assembly task. Experimental results on a C5 links parallel robot demonstrate that the robot's skill improves effectively and the force control performances are satisfactory.**

***Index Terms*—Neural network, Hybrid force/position control, Identification and learning**

I. Introduction

Many research works have been developed for better force control and safety. Hybrid force/position control is one of the most important and fundamental approaches to robot control in order to carry out sophisticated jobs. Although it has been studied for many years [1][2], this kind of control has not been implemented for practical use, since the environment affects the dynamics of the whole system. One of the main problems of hybrid force/position control is the difficulty to obtain the dynamic properties related to every environment, where parameters such as impedance and friction are nonlinear, time and space varying. Moreover, in the case of tasks such as assembling or contour following under effort constraints, part contact geometry and motion speed are parameters which may vary from one task to another; and when the dynamics of an environment is unknown or uncertain, a robot controller has to adapt itself. Furthermore, since the desired force is often given as step signals, fast adaptation is expected to avoid accidental damage of both the robot and the environment. For this purpose, some research has been directed towards development of neural networks-based approaches for control of complex processes, and satisfactory results have been obtained [3][11-17].

The connectionist-based approach is based on the connectionist models and its robustness relies on the information acquired during the training phase that implicitly considers all the above mentioned parameters. Model-based methods do not offer a complete solution due to the uncertainties associated during the contact phase as mentioned earlier. Besides, connectionist-based techniques have proved to work reliably when uncertainty is involved due to their generalisation property. Adaptive neuro-force controllers have been successfully demonstrated [18–22] to be effective force controllers for unknown environment. Nevertheless, the major shortcoming of the existing methods is the possibility of undesired oscillation or unexpected overdamped/underdamped responses which may occur until the controller adapts to the environment dynamics. Moreover, a mathematical model of the robotic system must be known a priori and the control inputs must either be calculated from complex dynamic equations in real time or from a pre-calculated and stored array.

In order to cope with this problem and to obtain satisfactory overall closed-loop performances, a new adaptive neural networks approach based hybrid force/position control is proposed. This approach uses three multilayered neural-networks and operates in two stages: At first, a first feed forward neural-network estimator (FFNN-Estimator) is used for estimation of free motion forces. It learns off-line these forces using variable metric method combined with a one dimensional optimization in order to improve robustness and to accelerate convergence [16]. A second feed forward neural-network observer (FFNN-Observer) which is used as an identifier learns off line the dynamic of the system using variable metric method. The third feed forward neural-network controller (FFNN-Controller), which is used as a controller, is first trained off-line to learn the input/output relation from a classical hybrid force position controller using variable metric method. Finally, an online learning of the FFNN-Controller, using FFNN-Observer parameters, is implemented in the proposed control structure, in order to maintain a good closed-loop performance and to compensate for uncertain/unknown dynamics of the robot/environment interaction. At this stage, a

Y. Touati is with the Computer Science and Robotics Lab. LISSI, Paris 12 Val de Marne University, 122 rue Paul Armangot, 94400 Vitry sur Seine, FRANCE (telephone: 001-4180-7382, e-mail: touati@univ-paris12.fr).

Y. Amirat is with the Computer Science and Robotics Lab. LISSI, Paris 12 Val de Marne University, 122 rue Paul Armangot, 94400 Vitry/Seine, FRANCE (telephone: 001-4180-7383, e-mail: amirat@univ-paris12.fr).

N. Saadia is with the Electronics Lab., Algiers Technologies and Sciences University, 16123 Algiers, ALGERIA (telephone: 00213-02124-7950, e-mail: nsaadia@usthb.dz).

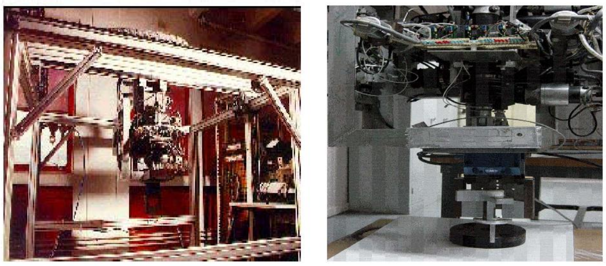
1-4244-0195-X/06/$20.00 ©2006 IEEE

backpropagation method is applied to adjust online the FFNN-Observer and FFNN-Controller parameters. A typical example on which we shall focus is an assembly robot interacting with its environment. Experimental results on a C5 links parallel robot demonstrate that the robot's skill improves effectively and the force control performances are satisfactory, even if the dynamics of the robot and the environment change.

The remainder of the paper is organized as follows. In Section 2, we describe the experimental setup. Section 3 presents the formulation of the force/position control problem and the proposed approach. In section 4, some experimental results are illustrated and analysed, and finally, some remarks and perspective are presented.

II. Experimental Cell Description

The industrial application, on which we shall focus, to validate our approach, concerns an assembly task. The experimental setup is composed of a 2D Cartesian robot linked to a six DOF parallel robot, also called C5 parallel robot (Fig.1(a)). The letter acts as an active force controlled wrist of the Cartesian robot (Fig.1(b)).



(a) (b)

Fig. 1. Whole view of the experimental setup

This setup is intended to perform complex and various force controlled tasks, such as tight tolerance insertion of parts with various shapes and contour following tasks. The 2D Cartesian robot allows moving parts in order to carry them from a given position towards the operational area. It can perform small corrective displacements around the nominal trajectory. The parallel robot consists of a static part and a mobile part linked together by six actuated segments.

In our approach, an assembly task operates in three steps: In the first one, wide amplitude displacements are performed by the 2D Cartesian robot in order to bring the assembly parts to close vicinity. In the second one, a very accurate corrective trajectory is performed by the parallel robot under control of an external vision sensor in order to perform the proper location of the moving part with respect to the receptive part. The vision system measures the relative positioning of the parts to assemble. Finally, the assembly or the final insertion is performed. During this phase, contact between parts may arise. It is therefore necessary to implement a force feedback control of the parallel robot in order to carry out this task successfully. This force feedback is needed for security constraints and to guaranty regularity and quality parts.

III. Proposed Approach

In some industrial applications, such an assembly task, the system interacts with the workspace, and thus its motion is constrained by the task. In this case, pure position control will be ineffective since forces which appear after the contacts must also be controlled. In this case, simultaneous position and force control called hybrid control is then required. Several force-position control approaches have been proposed in the literature, but the main drawbacks within such methods concern mathematical model determination of the controlled system and the computing of control inputs in real time situation. To overcome these problems, a hybrid identification and control approach based on neural networks theory is proposed (Figure 2).

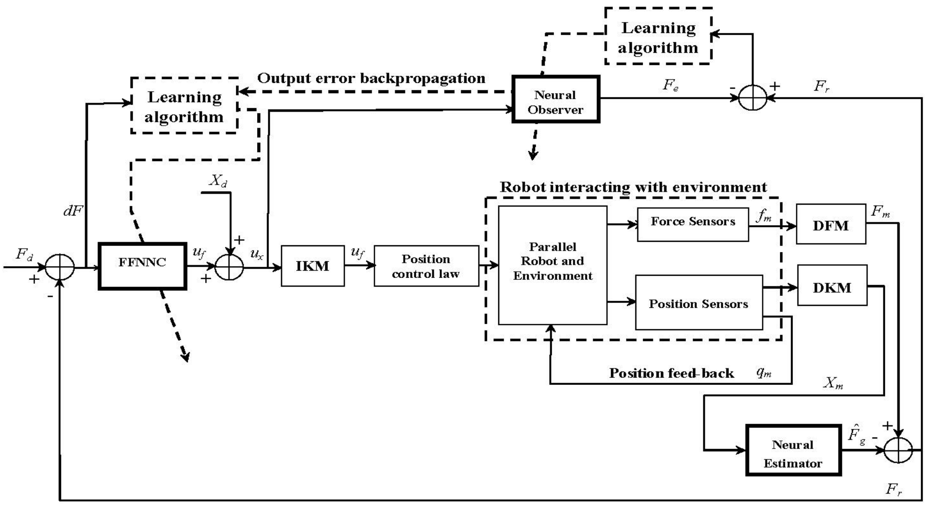


Fig. 2. Proposed hybrid control structure

The direct force model DFM is used to calculate the measured efforts F_m in task space based on the measured ones, f_m, in joint space. The inverse kinematics model IKM and the direct kinematics model DKM calculate the measured trajectories (q_m, X_m), respectively in joint and task spaces. The variables F_d and X_d represent respectively the desired efforts and trajectories in task space. The force control loop is hierarchized with respect to the position control loop, so that the force error dF is corrected, based on the variation of the position u_f, around the nominal position trajectory X_d according to the relationship:

$$u_f = C(dF) \quad (1)$$

Where C represents the force control law, which is generally called compliance model for force control.

The proposed approach uses three multilayered neural networks with a same architecture. It includes one input layer, one output layer and one hidden layer, which implements a nonlinear sigmoid function defined as:

$$f(x) = \frac{1}{2}.X_g \frac{1-\exp(-4x/X_g)}{1+\exp(-4x/X_g)} \quad (2)$$

X_g is a parameter which determines the sigmoid function shape.

A FFNN-Estimator is used for estimation of free motion forces. This network is adjusted off line using the training data obtained from the robot's displacements in the free space. In constrained motion, a FFNN-Controller is used as compliance controller. The implementation of this controller requires two steps: At first, the FFNN-Controller is initialized off line from the identification of an external force position controller. In the second step, the objective is to build a suitable real adaptive neural control. For this purpose, a FFNN-Observer is implemented and acts as an identifier to learn off line plant's dynamic. Then, the FFNN-Controller is implemented, also, online using backpropagation method through the FFNN-Observer.

A. *Off-line Learning Phase*

In this stage, the FFNN-Controller is initialized off line from the identification of an external force position controller. Knowing that, the design of an adaptive control law for nonlinear and unknown systems is extremely difficult, a FFNN-Observer is implemented and trained off line on the basis of data acquired during free and constrained motions by identifying the plant's dynamic. Due to nonlinear resulting from the nonlinear distribution of robot's inertia during a compliant motion, an appropriate neural network (FFNN-Estimator) can be used to estimate the contact forces from the measured ones [3].

In the following, a neural networks conception methodology for nonlinear and unknown systems control is proposed. This methodology, which is based on an assembly task performing such a peg-in-hole insertion, proceeds in three phases:

1) Free Motion Forces Estimation

As illustrated in Fig. 3, this network estimates forces due to the gravitational effects of the robot, and is adjusted off line using the training data obtained from the robot's displacements in the free space.

The objective is to build a suitable model, which when excited by input X_m, produces an output $\hat{F}_g$ that approximates F_g such that the total square error function $J(w_e)$ is optimized by the network, so that:

$$J(w_e) = \frac{1}{2}\cdot(F_g - \hat{F}_g)^2 \quad (3)$$

Where F_g and $\hat{F}_g$ respectively, the desired output network corresponding to the input pattern and the corresponding actual output, and w_e, represents the FFNN-Estimator parameters (weights and bias).

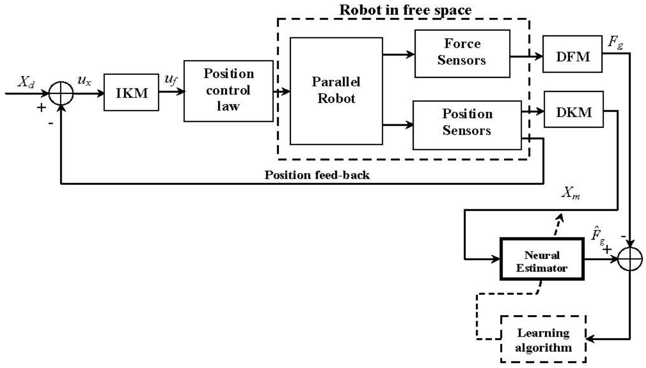


Fig. 3. Free motion forces estimation by the FFNN-Estimator

The FFNN-Estimator is trained off line by performing a learning process based on a Quasi-Newton method combined with one dimensional search [17]. The position control law implements a PID controller to perform an accurate trajectory in free space.

2) Neural Network Controller Learning

In this subsection, the FFNN-Controller is implemented off line to learn the input-output relation from a classical hybrid force/position controller using variable metric method (Figure 4). In this case, a PID force controller parameters are tuned employing an extension of the Taguchi method for multivariable systems, proposed by Ryckebusch [18][17].

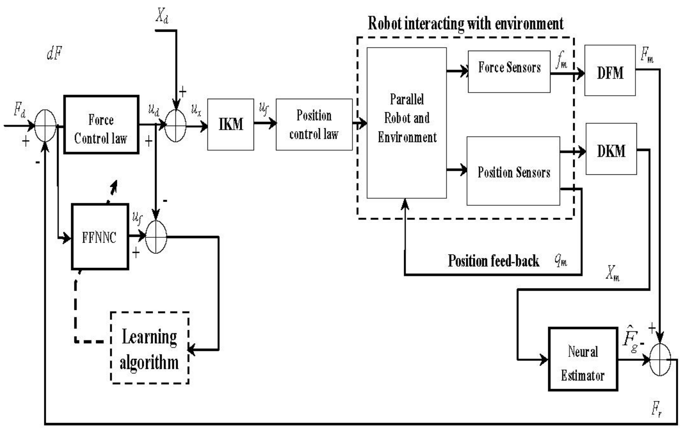


Fig. 4. On-line FFNN-Controller Learning

The parameters of the FFNN-Controller are tuned off line using a Quasi-Newton method, according to the following relation:

$$w_c(k+1) = w_c(k) - \eta_c \frac{\partial J(k)}{\partial w_c(k)} \quad (5)$$

Where η_C and $w_C(w_{Ci}, b_{Ci})$ respectively, the convergence speed and the parameters (weights and bias) of the FFNN-Controller.

The learning process involves the determination of the vector w_C which minimizes a cost function J, given as follows:

$$J(p)=\frac{1}{2}\sum_{k=1}^{p}\left(u_d(k)-u_f(k)\right)^2 \quad (6)$$

The corresponding actual output u_f is defined as follows:

$$u_f = N(w_c, dF) \quad (7)$$

Where u_d, dF and w_c respectively the desired output control, the input and the weight vector of the FFNN-Controller.

3) Plant Dynamic Identification

Once the FFNN-Controller is constructed from classical force/position control learning, the FFNN-Observer is implemented to maps the forward dynamics of the plant, as shown in Fig. 5.

The learning algorithm which is performed off line is based on the combination of two methods. A variable metric method combined with a one dimensional optimization to improve robustness and to accelerate convergence [18]. The FFNN-Observer parameters are initialized randomly and tuned according to the following relation:

$$w_i(k+1)=w_i(k)-\eta_i\frac{\partial E(k)}{\partial w_i(k)} \quad (8)$$

Where, $E(k)$ is the cost function given as follows:

$$E(k)=\frac{1}{2}\left(F_r(k)-F_e(k)\right)^2 \quad (9)$$

Where η_i and $w_i(w_{ii}\ ,\ b_{ii})$ respectively, the convergence speed and the parameters (weights and bias) of the FFNN-Observer. From the desired output vector F_r, the FFNN-Observer output F_e is defined by the following equation:

$$F_e(k)=w_{i2}{}^t(k)\cdot f\left(w_{i1}(k)\cdot u_x(k)+b_{i1}(k)\right)+b_{i2}(k) \quad (10)$$

Where $w=(w_i,b_i)$, are the parameters of the observer, u_x is the neural network input and $f(.)$ is the sigmoid function.

B. Real Time Control Phase

In the second phase, the objective is to build a suitable real adaptive neural control and to increase the performances. For this purpose, the FFNN-Controller is implemented online using dynamic backpropagation method through the FFNN-Observer.

The parameters of the FFNN-Controller, w_c are tuned on line after one trial according to the following relation:

$$w_c(p+1)=w_c(p)-\eta_c\frac{\partial J(p)}{\partial w_c(p)} \quad (11)$$

Where J, is the cost function given by:

$$J(p)=\frac{1}{2}\sum_{k=1}^{p}\left(F_d(k)-F_r(k)\right)^2 \quad (12)$$

Where p is the trial number, F_d is the desired contact force and F_r is the measured one. The FFNN-Controller output is defined according to the following equation:

$$u_f(k)=w_{c2}{}^t(k)\cdot f\left(w_{c1}(k)\cdot(F_d(k)-F_r(k))+b_{c1}(k)\right)+b_{c2}(k) \quad (13)$$

From (12), the derivatives $\frac{\partial J(p)}{\partial w_c(p)}$ can be formulated by:

$$\frac{\partial J(p)}{\partial w_c(p)}=\frac{\partial J(p)}{\partial F_r(p)}\cdot\frac{\partial F_r(p)}{\partial w_c(p)}$$
$$=-\sum_{k=1}^{p}\left(F_d(k)-F_r(k)\right)\cdot\frac{\partial F_r(p)}{\partial u_f(p)}\cdot\frac{\partial u_f(p)}{\partial w_c(p)} \quad (14)$$

Since the observer output approximates the real plant output, from (14), we can write:

$$\frac{\partial F_r(p)}{\partial u_f(p)}=\frac{\partial F_e(p)}{\partial u_f(p)}$$
$$=w_{i1}{}^t(p)\cdot w_{i2}(p)\cdot f'\left(w_{i1}(p)\cdot u_x(k)+b_{i1}(p)\right) \quad (15)$$

The matrix $\frac{\partial u_f(p)}{\partial w_c(p)}$ can be approximated by the following relation:

$$\frac{\partial u_f(p)}{\partial w_c(p)}=\frac{\partial u_x(p)}{\partial w_c(p)} \quad (16)$$

Since the FFNN-Controller output is known, the derivative $\frac{\partial u_x(p)}{\partial w_c(p)}$ can be determined. Hence (14) is defined.

IV. Experimentation Results

In order to validate our approach, we have implemented an assembly task with high accuracy (*tolerance=0.1 mm*). The task consists of inserting a peg in a hole with a minimum force contact (Fig. 5).

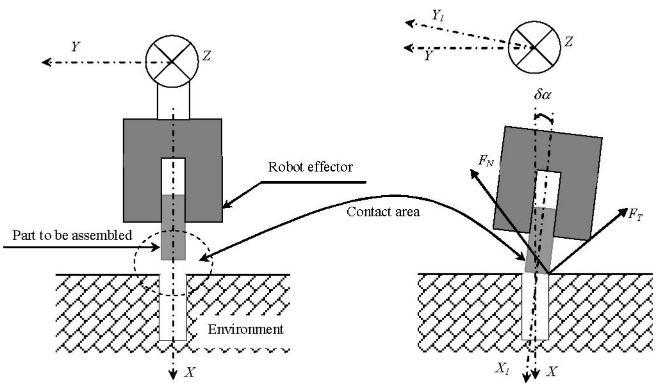


Fig. 5. C5 Parallel Robot Performing an Assembly Task

As shown in Fig. 5, F_N and F_T represent respectively normal and tangential components of the force during constrained motions. The experiments are carried out on the C5 parallel robot.

The first step of the proposed neural network control approach is to estimate free motion forces by implementing a FFNN-Estimator. This network is a multilayered one, made up of three layers, six input-layer neurons to represent the position and the orientation of the end-effector X_m, thirty neurons in single hidden layer and six outputs neurons for the corresponding gravity force vector $\hat{F}_g$. The hyperbolic tangent is used as nonlinear activation function for the single hidden layer while a linear function is chosen for the output layer. The initial weights, w_e, are chosen randomly within the range of [-1;1].

Figure 6 illustrates the time evolution of the free motion forces vector $\hat{F}_g$ estimated by the FFNN-Estimator, with an insertion velocity of 10 mm/s.

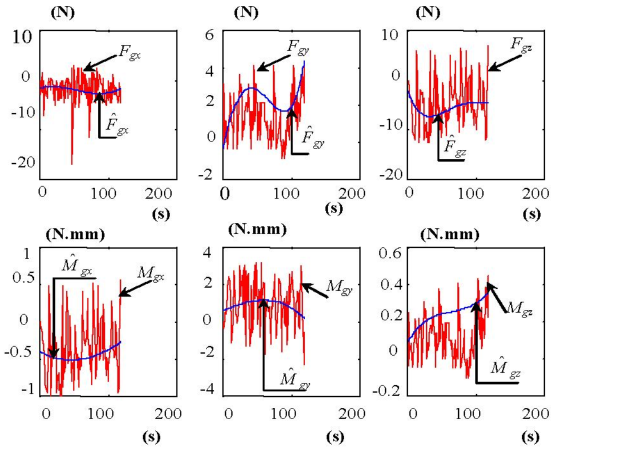


Fig. 6. Free Motion Forces estimation

We remark that the estimated forces components by the FFNN-Estimator exhibit the same behavior than the real free motion forces F_g .

To learn the control, an identical architecture of the FFNN-Estimator has been used except that the six input-layer neurons represent the difference between the desired and actual force vector respectively, F_d and F_r.

In this case, a PID force controller parameters are tuned employing an extension of the Taguchi method for multivariable systems, in order to initialize the FFNN-Controller.

The obtained results show clearly the implemented learning algorithm performances which is based on the Quasi-Newton method (Fig. 7). For this purpose, the convergence speed of the FFNN-Controller, η_c is around 10E-2. After 500 epochs, a sum squared error, defined in (6), equals to 10E-2, which represents a good result.

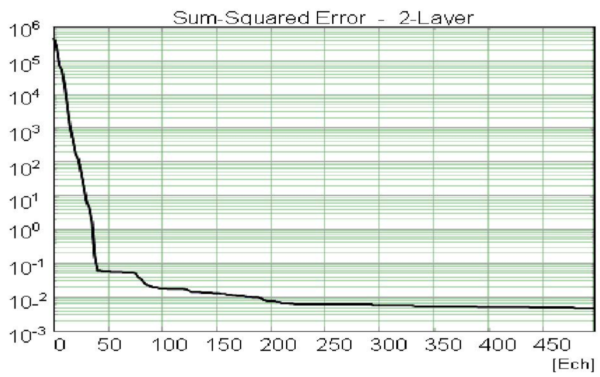


Fig. 7. FFNN-Controller on-line Learning

Once the FFNN-Controller is constructed from classical force/position control learning, the FFNN-Observer is implemented to maps the forward dynamics of the plant. Its architecture is the same as the FFNN-Controller one, except that the six input-layer neurons represent the sum of the desired position vector components X_d and the FFNN-Controller output components u_f.

The learning results of the FFNN-Observer show clearly the robustness and the convergence speed of the implemented training algorithm which is based on the combination of a variable metric method and one dimensional optimization (Fig. 8).

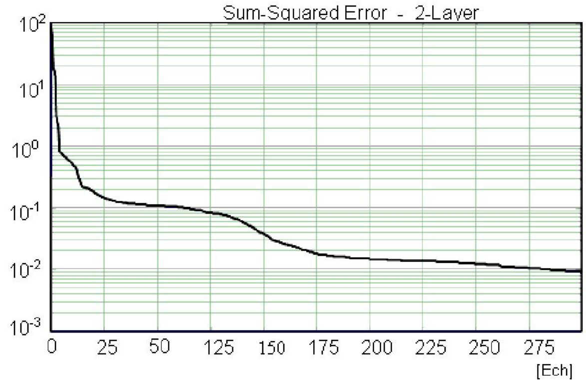


Fig. 8 Off-line FFNN-Observer learning

In this case, the convergence speed of the FFNN-Observer, η_i is around 10E-3. After 300 epochs, a sum squared error, defined in (9), equals to 10E-2.

In the second step, the FFNN-Controller is implemented on-line using dynamic backpropagation method through the FFNN-Observer. Figure 9 shows the time evolution of the obtained contact force vector components (F_x,F_y,F_z,M_x,M_y,M_z) with an Insertion velocity V_i of 10 mm/s.

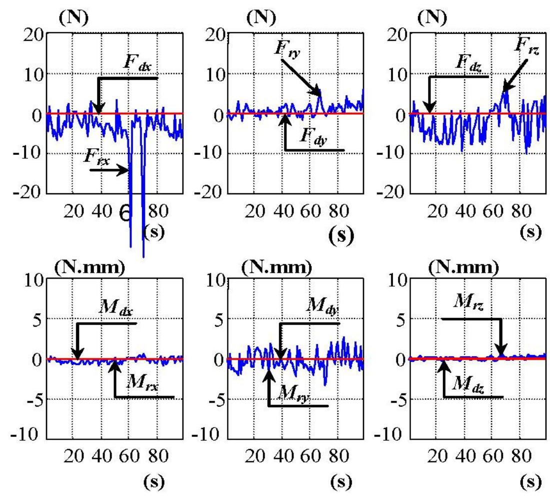


Fig. 9 Desired and Actual Forces on the End-effector (V_i=10 mm/s)

These illustrated results above, show the efficiency of the proposed control approach. The temporal evolution of the contact force vector's components (F_{rx},F_{ry},F_{rz},M_{rx},M_{rz},M_{rz}) with respect to the desired force's components (F_{dx},F_{dy},F_{dz},M_{dx},M_{dz},M_{dz}). The force component $\boldsymbol{F_{rx}}$ shows two peaks, after 60 iterations, indicating the contact between the pin and the receptive part. The remaining components of the contact force vector decreases and stabilize around the desired force vector.

Figure 10 shows an other experimentation results of the time evolution of the obtained contact force vector components (F_x,F_y,F_z,M_x,M_y,M_z) with an insertion velocity V_i of 15 mm/s. It can be observed that components F_{rx}, F_{ry} and F_{rz} exhibit peaks, which indicate occurrences of contact between the peg and the receptive part, and stabilize around 0 N. Similarly, the momentum component exhibits a peak of 0.4 N.mm during contact, then stabilizes around 0.2 N.mm.

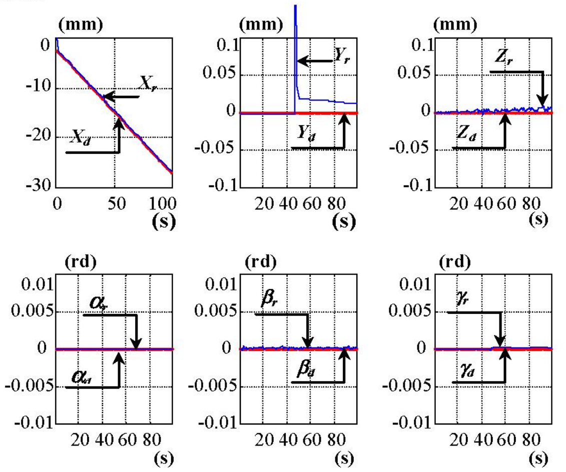


Fig. 10 Desired and Actual Position on the End-effector (V_i=10 mm/s)

When contact forces between parts increase, the corresponding position shows a peak and oscillates around the desired value, as illustrated by the Cartesian position vector component Y_r. Concerning the measured Cartesian orientation vector components, they show the same behavior as the desired ones.

These results show that the proposed control approach leads to a satisfactory dynamic behavior from force contact minimization point of view, even if the system dynamic is unknown.

V. Conclusion

In this paper, a neural networks approach for complex task control involving robot/environment interaction is presented. This approach realizes, simultaneously, an identification and control. It uses three multilayered neural-networks and operates in two stages: At first, a FFNN-Estimator is used for estimation of free motion forces. It learns off-line these forces using variable metric method combined with a one dimensional optimization in order to improve robustness and to accelerate convergence. A second FFNN-Observer which is used as an identifier learns off line the dynamic of the system using variable metric method. The third FFNN-Controller, which is used as a controller, is first trained off-line to learn the input/output relation from a classical hybrid force position controller using variable metric method. Finally, an online learning of the generated FFNN-Controller is implemented in the proposed control structure. During this phase, a backpropagation method is applied to adjust online controller parameters. The approach has been implemented for the control of an experimental setup, including a 2D Cartesian robot linked to a C5 parallel robot, and performing an assembly task. The analysis and evaluation of the obtained results show the suitability and efficiency of the proposed approach. These promising results lead us to further investigation into the use of this control approach for other kinds of tasks.

VI. Acknowledgment

The authors would like to thank the reviewers for their useful remarks.

References

[1] G. Zeng and A. Hemami, "An overview of robot force control", Robotica, vol. 15, pp. 473-482, 1997.

[2] K. Kiguchi and T. Fukuda, "A survey of force control of robot manipulators using soft computing techniques", IEEE Int. Conf. on System, Man, and Cybernetics, pp. 764-769, 1999.

[3] N. Saadia, Y. Amirat, J. Pontnau and N. K. M'Sirdi, "Neural hybrid control of manipulators, stability analysis", Robotica, vol. 19, pp. 41-51, 2001.

[4] D. Psaltis, A. Sideris and A. Yamamura, "A multilayered network Controller", IEEE Control System Magazine, pp. 17-21, 1988.

[5] H. Asada, "Teaching and learning of compliance using neural nets: representation and generation of nonlinear compliance", IEEE Int. Conf. on Robotics and Automation, pp.1237-1244, 1990.

[6] D.H. Nguyen and B. Widrow, "Neural networks for self-learning control systems", IEEE Control System Magazine, vol.10, n°. 3, pp. 18-23, 1990.

[7] P. Antsakis, "Neural networks in control systems", *IEEE Control System Magazine*, vol.10, pp. 3-5, 1990.

[8] J.H. Sira-Ramirez and S.H. Zak, "The adaptation of perceptrons with application to inverse dynamics identification of unknown dynamic systems", *IEEE Trans. on Sys. Man. and Cybernetics* SMC, vol.21, pp. 634-643, 1991.

[9] S.S. Ge and T.H. LEE, "Adaptive neural network control of flexible joint robots based on feedback linearization", *Int. J. Systems Science*, vol. 29, n°. 6, pp. 1229-1234, 1998.

[10] T. Zhang, S.S. Ge and C.C. Hang, "Adaptive output feedback control for general nonlinear systems using multilayer neural networks", *Proc. American Control Conf.*, Philadelphia, vol. 1, pp. 520-524, 1998.

[11] M. Tokita, T. Mitsuoka, T. Fukuda, T. Kurihara, "Force control of robot manipulator by neural network model", *Journal of Robotics and Mechatronics,* vol. 2, pp. 273-281, 1990.

[12] T. Yamada, T. Yabuta, "Neural network controller using autotuning method for nonlinear functions", *IEEE Trans. on Neural Networks,* vol. 3, pp. 595-601, 1992.

[13] K. Kiguchi, D.S. Necsulescu, "Control of multi-DOF robots using neural networks", Proceedings of the Knowledge-Based Systems and Robotics Workshop, pp. 747-754, 1993.

[14] K. Kiguchi, T. Fukuda, "Neural network controllers for robot manipulators application of damping neurons", Advanced Robotics, vol. 12 pp. 191-208, 1998.

[15] S. Jung, T.C. Hsia, "Neural network impedance force control of robot manipulator", *IEEE Trans. on Industrial Electronics*, vol. 45, pp. 451-461, 1998.

[16] Y. Amirat, F. Artigue and F. Pontnau, "A six degrees of freedom parallel robots with C5 links", Robotica, vol. 10, n°. 1, pp. 35-44, 1992.

[17] L. E. Scales, "Introduction to non linear optimization", Springer Verlag, New York, 1985.

[18] N. Saadia, Y. Amirat, J. Pontnau and A. Ramdane-Cherif, "Hybrid force/position control of nonlinear systems using neural networks", Sec. ECPD Int. Conf. on Adv. Rob. Int. Aut. and Act. Sys., Vienna, Austria, pp. 68-73, 1996.

[19] G. Taguchi and S. Konishi, "Orthogonal Arrays and Linear Graphs", ASI Press, 1987.

[20] E. F. Ryckebusch, I. K. Craig, "PID Tuning For A Multivariable Plant Using Taguchi-based Methods", Proc. of the 15th IFAC World Congress On Automatic Control, Barcelona, Spain, July 21-26, 2002.

[21] J.H. Sira-Ramirez and S.H. Zak, "The Adaptation of Perceptrons with Application to Inverse Dynamics Identification of Unknown Dynamic Systems", IEEE Trans. on Sys. Man and Cybernetics, SMC, vol. 21, pp. 634-643, 1991.