第12卷第5期 控制理论与应用 Vol. 12, No. 5
1995年10月 CONTROL THEORY AND APPLICATIONS Oct., 1995

Force Control of Manipulators with Neural Networks*

PEI Hailong and ZHOU Qijie
(Department of Automation, South China University of Technology · Guangzhou, 510641, PRC)
T. P. Leung
(Hong Kong Polytechnical University · Hong Kong)

Abstract: In this paper, a new robot force control strategy based on neural network learning is proposed. The controller is composed of two neural networks in cascade. One of them is used to learn the inverse dynamics of the robot, while the other one to learn to express the unknown environment dynamics. With this method, the difficult environment modelling problems can be avoided. To indicate the efficiency of this algorithm, the force control problem of a two-link robot is engaged. Simulation results show that this straightforward method is very feasible.

Key words: neural networks; robot; force control

1 Contact Tasks and Virtual Internal Model

Today's robots need to provide for more sophisticated motions. To meet these demands, the control function required for the robots has become more sophisticated and versatile. Force control is one of the most challenging problems in robot control, and there has been a significant attention directed to compliant or force control tasks[3] in this decade. These tasks can be identified as "interactive tasks", defined as tasks where the manipulator comes into contact with the environment and effects some changes, such as grinding, assembly, and fixturing. Several control methods are currently used for these interactive tasks. Most of them can be categorized into a type of active compliance which is specified in the joint servo either by setting a linear relation between force and displacement (or force and velocity) (shown in Fig. 1), such as impedance control[1], stiffness control[4], or by controlling force in certain degrees while controlling position in the remaining degrees such as compliance control, hybrid control[2].

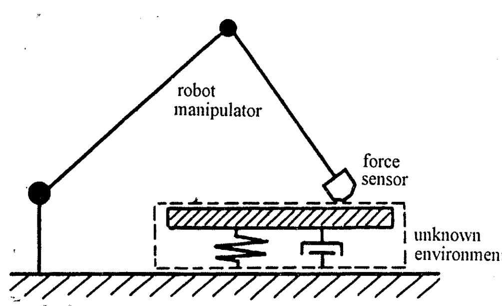

Fig. 1 Contact task

* This work was supported in part by the Beijing-Hong Kong Academic Exchange Centre under Grant 90078/R and Guangdong Science and Technology Research Fund.
Manuscript received Dec. 24 1994, revised, Jul. 20, 1995.

For the sake of simplicity, consider a force pushing against an environment which is usually modeled by a mass, a spring, and a damper as shown in Fig. 1, i. e., with force being supplied by an ideal source F_a the environment has effective stiffiness K_e, damping B_e, and mass M_e which obey the following relation

$$- F_{env} = K_e + B_e \frac{dx}{dt} + M_e \frac{d^2x}{dt^2} \tag{1}$$

where F_{env} is the impact force supplied by the environment, x is displacement of the surface. Most force control strategies are based on this assumption or its variations. However, the model parameters are difficult to determine for the reason that they are not constant in the whole impact procedure, (for a hard surface, the relation between force and displacement is not a linear one), or even can not be expressed explicitly (the mass or inertia affected, for example). Besides, there are many uncertainties in contact tasks can not be included by this model, for example, the effect of frictions. Most force control schemes assume (either implicitly or explicitly) that a model of the environment is available a priori. Though there have been some methods treating this problem, such as adaptation[5] and filtering[6], it is difficult to design a stable and appropriate response with no knowledge of the environment.

Virtual internal model method is an intuitive constructive control strategy. It is composed of a virtual trajectory generator and a general servo controller. In the conventional robotic servo system, the dynamic behaviour of a robot arm, that can be modeled as,

$$\tau = m(\theta)\ddot{\theta} + h(\dot{\theta},\theta) \tag{2}$$

is controlled so that the controlled variables of the arm can track the reference signal as precisely as possible. A general continuous path tracking scheme described as (3) is of this kind

$$\tau = M(\theta)[\ddot{\theta}_d + k_1(\dot{\theta}_d - \dot{\theta}) + k_2(\theta_d - \theta)] + h(\dot{\theta},\theta) \tag{3}$$

where, $\Theta = (\theta,\theta,\theta)^{\mathrm{T}}$ is joint variables, $\Theta_d = (\theta_d,\theta_d,\theta_d)^{\mathrm{T}}$ is desired joint path, and τ is the control torque. If there is a constraint on the end effector, a force is exerted on the end effectror. This can be expressed as

$$\tau = M(\theta)\ddot{\theta} + h(\dot{\theta},\theta) + J^{\mathrm{T}} f_{env} \tag{4}$$

where J is the Jacobian matrix of the robot.

To track the desired path Θ_d on a surface $\Psi(\Theta)$ while exerting a given force $f_d = -f_{env}$ on this suface, virtual model method is introduced. The basic idea is to construct a virtual trajectory Θ_v

$$\ddot{\theta}_v = \Gamma(\Theta_d,\Theta,f_d,f_{env}) \tag{5}$$

so that the robot can generate a force on the constraint surface which can approach f_d properly by trying to track Θ_v. With an appropriately designed virtual model the force error

$$\begin{aligned} e_f &= J^{\mathrm{T}}(f_d - f_{\mathrm{env}}) \\ &= M(\theta)\ddot{\theta}_v + M(\theta)[\ddot{\theta}_d + k_1(\dot{\theta}_d - \dot{\theta}) + k_2(\theta_d - \theta)] + h(\dot{\theta},\theta) - \tau \end{aligned} \tag{6}$$

will approach zero with a proper convergence behavior.

A good virtual model chosen for robot force control should require neither the total decoupling of force and position, nor the unnecessary sacrifice of accuracy in position or force. As mentioned before, there still exists the fundamental problem which is unavoidable in virtual trajectory design, that is the need for knowledge of environment information, without which stability and overall performance cannot be ensured. Besides, the design procedure is not an easy one.

Recently the applications of artificial neural networks have attracted huge attentions of scientists in many fields for its emerging abilities in learning, adapting, and associative memory. There have been several attempts to apply artificial neural networks to control system designs[7]. The robot force control problems with neural networks were discussed in a few papers[8,9]. Based on impedance control method[8] Cohen and Flash gave a method by means of associative search network (ASN) learning to search the appropriate impedance parameters of the controller. This method did avoid the direct teaching signal acquiring, however, because the engaged reinforcement learning scheme is a much time consuming method, the convergence behavior can not be observed distinctly. Fukuda and his colleagues[9] proposed to use the hybrid control method to train a neural network directly to implement hybrid position/force control. Unfortunately, there was not an apparent description of the relation between their force controller and position controller. In this paper, it is proposed to use sensed force and position information to construct a virtual internal model with neural networks through supervised learning. Suppose no knowledge about the robot dynamics and environment dynamics is obtained, while the geometry of the robot and constraint surface is known. The contact task is to control the robot to slide along a given path on this surface while exerting a desired force on the surface. Two neural networks are engaged to construct our force controller. One of them is used to learn the inverse dynamics of the robot, the other one is to learn to express the virtual model. With the proposed separate learning strategy, the neural network can gain a proper virtual trajectory that can meet the demand of the force control tasks perfectly. Different to the two network control strategies mentioned above, the proposed control structure avoids to learn to treat the whole complex control mission with one big network, but decomposes it into two simple ones which can be trained separately. This is called composite neural network controllers in [15][17]. Detail existence conditions and learning procedures are discussed in [17]. To indicate the efficiency of this algorithm, a two-link robot is engaged to perform the force control along a planer surface. Simulation results show that this scheme is very feasible. This attempt is an extension of our early work on neural network robot trajectory control[14]. Furthermore this control method can also be developed easily to other contact tasks in which the contact environment is difficult to model.

The rest of this paper is arranged as folllows: construction of the neural network controller and its separate learning strategy are described in section 2. In section 3, results of the computer simulations are presented. Followed by some conclusions in the last section.

2 Neural Network Force control

In this section, two neural networks are engaged in cascade to perform force control. The first is used to describe the inverse dynamics of the robot manipulator which is invariant, while the another one is to simulate the so called virtual model as mentioned above to generate an appropriate virtual signal to the first network so that the two connected networks can perform force control. The whole system is described in Fig. 2, and the detail of the controller is shown as Fig. 3. For the sake of simplicity, equation (10) is replaced by $\dot{S}_1 = 0$ in Fig. 2, Fig. 3 and Fig. 4.

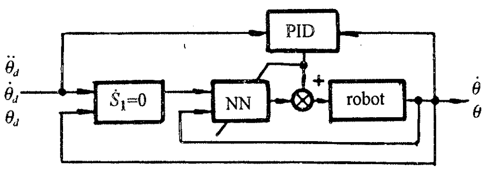

Fig. 2 Neural network force control system

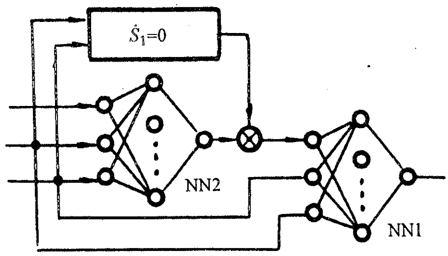

Fig. 3 Neural network controller

The two cascade neural networks can be trained jointly or in turn. But in any case, the teaching signals cannot be obtained directly. To obtain the error signals of the second neural network at its output, the first network must be engaged simultaneously, i. e., the errors at the output of the first network have to be propagated to the second network. To ensure the stability of the whole learning control system, very small learning ratios have to be chosen if combined learning strategy is used. So this method is of poor convergence properties and time consuming. Here it is preferred to train the two networks separately.

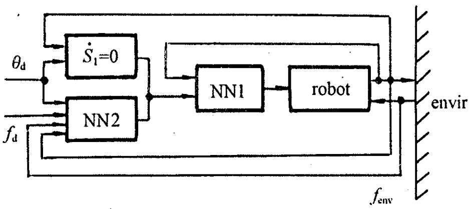

Fig. 4 Inverse dynamics learning

Learning of inverse dynamics

Robot inverse dynamics is a system invariance, so it can be learned through some voluntary movements before the constraint motion. Consider an unknown robot dynamics show as

$$\tau = f(\ddot{\theta}, \dot{\theta}, \theta) \tag{7}$$

and suppose that a simple PID feedback controller represented by

$$\tau_{s_1} = \alpha[(\dot{\theta}_d - \dot{\theta}) + k_1(\theta_d - \theta) + k_2\int_0^t(\theta_d - \theta)\mathrm{d}t],$$

$$= \alpha S_1(e_p), \quad \alpha > 0, \tag{8}$$

can guarantee the convergence of θ, here, $e_p = \theta_d - \theta$ is the joint position error. Usually k_j are chosen such that $p^2 + k_{i1}p + k_{i2}$ is a Hurwitz polynomial, i.e., $S_1(e) = 0$ is a stable surface to the error $e(t)$. A multi-layerd neural network expressed as (9) is used to learn the inverse dynamics of (7),

$$\tau_n = \Phi_1(\ddot{\theta}_0, \dot{\theta}, \theta, w_1) \tag{9}$$

where, w_1 is the adaptable weights, θ_0 is the expected acceleration response of the controlled robot that can be deduced from $\dot{S}_1 = 0$,

$$\ddot{\theta}_0 = \ddot{\theta}_d + k_1(\dot{\theta}_d - \dot{\theta}) + k_2(\theta_d - \theta). \tag{10}$$

With the gradient-like method[12,13], an efficient neural network learning strategy can be derived as

$$\frac{\mathrm{d}w_1}{\mathrm{d}t} = \eta_1 \left(\frac{\partial \Phi_1}{\partial w_1} \right)^{\mathrm{T}} S_1 \tag{11}$$

where η_1 is a positive factor which determines learning rate. With this method the neural network can learn the true inverse dynamics of the robot while $S_1 \to 0$. The details of the training method as well as the trajectory control problems have been discussed in our early work[14], in which a robust designing with variable structrue method is also studied. It should be mentioned that theoretical results in [16] show that with η_1 small enough, the learning system can ensure the stability of the controlled object as well as the learning convergence of neural network controller.

Learning of virtual model

The learning of the second network is similar to that of the first one except that the supervising signals are not what in the work/joint space but the propagated ones of the simple PI force controller shown as (12) through the first network (the simulated inverse dynamics).

$$\tau_{S_2} = \beta[\tau_{f_d} - \tau_f + q\int_0^t (\tau_{f_d} - \tau_f)\mathrm{d}t] = \beta S_2, \quad \beta > 0 \tag{12}$$

where, q are positive constants with appropriate values which can determine force convergence behaviors. This is shown in Fig. 5.

According to equation (5), a neural network which can be expressed as

$$\ddot{\theta}_{n_v} = \Phi_2(\ddot{\theta}_d, \dot{\theta}_d, \theta_d, \dot{\theta}, \theta, w_2) \tag{13}$$

is employed to simulate the virtual model.

Suppose the first network is perfectly learned, then the torque error caused by the inexactly learned virtual model can be describled as

$$\beta S_2(e_f) = \Phi_1(\ddot{\theta}_0 + \delta\ddot{\theta}_{nv}, \dot{\theta}_d, \theta_d, \dot{\theta}, \theta, w_1^*). \tag{14}$$

Similar to the inverse dynamics learning, the learning strategy can be deduced as (15)

$$\frac{dw_2}{dt} = \eta_2 \left(\frac{\partial \Phi_2}{\partial w_2} \right)^{\mathrm{T}} \left(\frac{\partial \Phi_1}{\partial \ddot{\theta}_{n_v}} \right)^{\mathrm{T}} S_1, \tag{15}$$

where, $\delta\ddot{\theta}_{n_v} = \Phi_2(w_2^*) - \Phi_2(w_2)$,and " $*$ " means the desired values.

To ensure the stability of the learning system it is proposed to add an extra position PID controller to compensate the position errors caused by the inexact initial force control virtual model. So the total torque exerted on robot joints is

$$\tau = \Phi_1 + \alpha[(1-\mu)S_1 + \mu S_2]. \tag{16}$$

The whole learning system is shown in Fig. 5. For a rigid surface, this extra PID position controller will not affect the static force error.

On finishing the training, the virtual model is well learned and the PID position controller and the PI force controller are of no use, then they can be cut down from the loop.

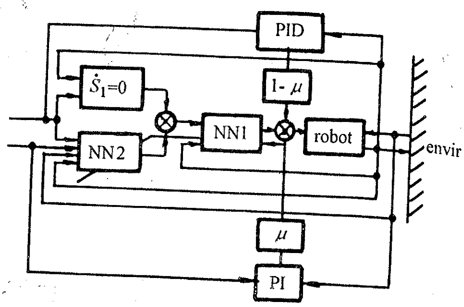

Fig. 5 Virtual model learning

3 Simulation Experiments

To indicate the efficiency of the proposed learning method, force control problem of a two-link robot is demonstrated with computer simulations.

Suppose that nothing about the robot dynamics and the environment dynamics is known. The simulated contact task is preferred as a robot sliding along a frictionless plane with a speed of 0.1m/s, and exerting a desired force f_d on this surface. The work is shown in Fig. 1. This plane is 0.5 meters long and is located 0.8 meters apart from the robot shoulder. The desired force is chosen as

$$f_d = \begin{cases} 25.0t\ \mathrm{N}, & 0 < t < 0.2\mathrm{s}, \\ 5.0\ \mathrm{N}, & t > 0.2\mathrm{s}. \end{cases} \tag{16}$$

The neural networks engaged are of the same structure as that used in the early work[14]. They are three layer feedforward networks, with two nodes in each network output layer, 30 and 20 neurons in each first and second hidden layer separately. The first network (expressing the inverse dynamics) has 6 inputs (3 inputs×2 joints) while the second one (expressing virtual model) has 12 inputs.

The training of the first network is similar to what we did in [14]. The training path is chosen near the contact surface so as to exactly approximate the inverse dynamics of the robot in the vicinity of the surface. Force control problems are usually of oscillatory behaviors, expecially when the environment is rigid. So, to learn to control contact tasks, very small learning rate is proposed to prevent unstable responses from happening.

After gaining the inverse dynamics (i.e., the first network is well trained) the first iteration of force/position learning control is not very satisfactory. This is shown in Fig. 6

and Fig. 7. Fig. 8 and Fig. 9 show that after some certain number of iterations, the oscillatory behaviors of the contact task have been manipulated well to some extent. This is means that the second neural network has gained the virtual model in the specific domain.

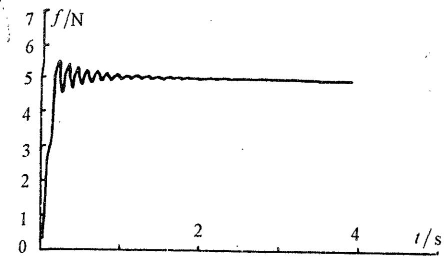

Fig. 6 Environment force

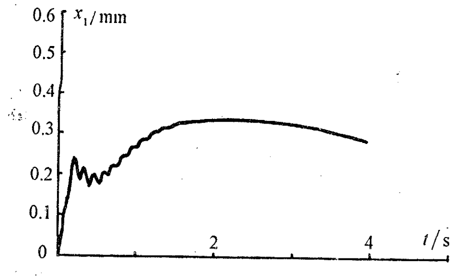

Fig. 7 Position error

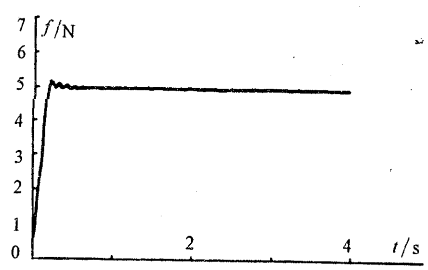

Fig. 8 Environment force

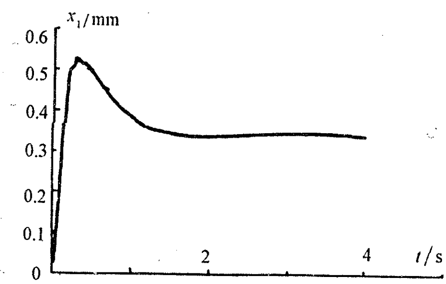

Fig. 9 Position error

In this simulation, the used robot parameters are the same as [14], while the environment parameters is supposed as

$$K_e = 3500\ \text{N/m}, \quad B_e = 1.0, \quad M_a + M_e = 2.5\ \text{kg}.$$

4 Conclusion and Discussion

In this paper, a cascade neural network force controller and its separate learning strategy are proposed. Besides the environment uncertainty learning ability, this controller has ome other advantages. Contact tasks are of complex behaviors, it is not easy to learn these esponses perfectly. In our designing the first network can be trained through voluntary notion. So training task is decomposed. With the two networks connected to accomplish ifferent functions, it is easy to implement the exchange of path tracking to force control. Jsually only path tracking control is installed in most robots, this method is useful to learn virtual model to help these robots to gain the ability of force control.

We just show a simple applicaton of neural network on robot control, further researches of this kind are undertaken in our future work.

References

] Hogan, H.. Impedance Control: An Introduction (Parts I, II, III). Journal of Dynamic Systems, Measurement, and

Control, 1985, 107:1—24
[2] Raibert, M. H. and Craig, J. J.. Hybrid Position/Force Control of Manipulators. ASME Journal of Dynamics systems, Measurement, and Control, 102:126—133
[3] Koivo, A. J.. Fundamentals for Control of Robotic Manipulators. John Wily & Sons, Inc., 1989, New York
[4] Salisbury, K.. Active Stiffness Control of A Manipulator in Cartesian Coordinates. Proc. 19th IEEE Conference on Decision and Control, Albuquerque, 87—97
[5] Fukuda, T., Kitamura, N. and Tanie, K.. Adaptive Force Control of Manipulators with Consideration of Object Dynamics. 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, April 1987, 1543—1548
[6] Jourdan, J. M. and Nagurka, M. L.. Environment Reconstruction and Force/Position Cycling Control of Robots in Interactive Tasks. Proc. of the UAS-Japan Symposium on Flexible Automation, 1988, 123—130
[7] Werbos, P. J.. An Overview of Neural Network for Control. IEEE Control Magazine, 1991, 11(1):40—41
[8] Cohen, M. and Flash, T.. Learning Impedance Parameters for Robot Control Using a Associative Search Network. IEEE Trans. on Robotics and Autom., 1991, 7(3):382—390
[9] Fukuda, T., Shibata, T., Tokita, M. and Mitasuoka, T.. Adaptation and Learning for Robotic Manipulator by Neural Network. Proc. of the 29th IEEE Conference on Decision and Control, Hawaii, 1990, 3283—3288
[10] Kawato, M., Furukawa, K. and Suzuki, R.. A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement. Biol, Cybern. 57, 1987, 169—185
[11] Gomio, H. and Kawato M.. Learning Control for a Closed Loop System Using Feedback-Error-Learning. Proceedings of the *t* Conference on Decision and Control, 1990, 3289—3294
[12] Williams, R. J. and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. Neural Computation, 1989, 1:270—280
[13] Rumelhart, D. E., Hinton, G. E. and Williams, R. J.. Learning Internal Representations by Error Propagating. in D. E. Rumelhart & J. L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, 1986
[14] Leung, T. P., Zhou Q. J. and Pei H. L.. A Robust Neural Network Controller. 1992 American Control Conference, 1992, 983—987
[15] Hai-Long Pei, Qi-Jie Zhou and T. P. Leung. Backward Construction-A Decomposed Learning Method for Robot Force /Position Control. Proc. of IEEE/INNs International Joint Conference on Neural Networks, Baltimore, Jun 1992, 2293—2298
[16] Pei Hai-Long, Zhou Qi-Jie and Leung T. P.. On Learning Convergence of A Kind of Real-Time Neural Network Nonlinear Control Systems. The First Chinese World Congress on Intelligent Control and Intelligent Automation
[17] Pei Hai-Long. Neural Network Control of Nonlinear Systems with Appplications to Robots. Ph. D Thesis, South China University of Technology, Oct. 1992

基于神经网络的机械臂的力控制方法

裴海龙　周其节　　梁天培
(华南理工大学自动化系·广州,510641)　(香港理工大学)

摘要：本文提出一种基于神经网络的力控制方法，由两个串联的神经网络构成机械臂的力控制器，其中一个网络用来学习机械臂本身的逆动力学系统，而另一网络用来学习未知的被接触环境的动力学特征，这种方法避免了困难的接触环境建模问题. 一个双连杆机械臂的力控制的仿真实验描述了这种方法的有效性.

关键词：神经网络；机械臂；力控制

本文作者简介

裴海龙 1965年生.于1993年在华南理工大学自动化系获工学博士学位.现为华南理工大学讲师.主要研究方向是神经网络控制,非线性控制和机器人控制.

周其节 见本刊1995年第3期第350页.

梁天培 见本刊1995年第3期第350页.

(上接第602页)

Tile	1997	Place	Deadline	Further Information
1997 European Control Conference (in cooperation with IFAC)	July 1-4	Brussels Belgium	1 Sept. 1996	M. Gevers/G. Bastin CESAME, Batiment Euler B-1348 Louvain la Neuve, Belgium FAX +32/10/472 180 e-mail: gevers @ auto. ucl, ac, be
IFAC/(IFORS) Symposium (11th) System Identification-SYSID '97	July 8-11	Fukuoka Japan	1 Sept. 1996	Prof. N. Suda Dept. of Systems Engineering Osaka University 1-3 Machikaneyama-cho, Toyonaka 560 Japan FAX +81/6/857 7664
IFAC Symposium (4th) Advances in Control Education ACE 97	July 14-16	Istanbul Turkey	15 Oct. 1996	Prof. A. Talha Dinibütün Istanbul Technical University Mech. Engg. Faculty Gümüssuyu 80191, Istanbul Turkey FAX +90/212-245 0795 e-mail: mkdnib @ tritu. bitnet
1997 Asian Control Conference (in cooperation with IFAC)	July 22-25	Seoul Korea, Rep.	1 Sept. 1996	Prof. Dong-il Cho 1997 ASCC Secretariat Automation and Systems Res. Institute Seoul National University, Bldg. 133 Kwanak-ku, Shinrim-dong, San 56-1 Seoul 151-742, Korean, Rep. FAX +82/2/889-4239 e-mail: ascc @ asri. snu. ac. kr
IFAC/(CIGRE) Symposium Control of Power Plants and Power Systems	August 18-21	Beijing China, P. R.	31 May 1996	Chinese Association of Automation POB 2728, Beijing 100081, China FAX +861/25 45 229
IFAC Symposium Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS '97	August 25-28	Hull UK	30 Sept. 1996	Prof. Ron Patton Dept. of Electronic Engg. Hull University Hull HU6 7RX, UK FAX +44/482/466664 e-mail: r. j. patton @ e-eng. hull. ac. uk