Design Of Robust Adaptive Controller

Régis Canart and Pierre Borne

Laboratoire d'Automatique et d'Informatique Industrielle de Lille
Ecole Centrale de Lille
Cité Scientifique BP 48
59 651 Villeneuve d'Ascq Cedex
FRANCE
e-mails : canart@ec-lille.fr ; p.borne@ec-lille.fr

Abstract: The aim of this paper is to develop a robust method to identify and control a MIMO nonlinear process in real time. Contrary to some approaches like neural networks techniques, the identification of the process is performed by adjusting a linear autoregressive model, which is defined at each time around the operating point of the process. This approach is an efficient way to find quickly and at low computational cost, an approximate model of the plant. It is necessary that a fast adaptation of this model takes into account the parameters changes and the nonlinearities of the process. This is also useful to reduce the effect of disturbances. The control law is then deduced from this model, such that errors between the outputs of the process and the desired outputs are reduced with a specified dynamics. This adaptive controller is applied to a manipulator whose aim is to control a laser beam.

Keywords: Identification, Least squares, Control Theory, Prediction, Disturbance

1. Introduction

We consider that the system to control is nonlinear and time varying. Furthermore we suppose that the process cannot be known exactly due to parameters incertitude. The system behaviour is also subject to disturbances and parameters changes. Under these conditions, to obtain a robust control, the variations of the process should be taken into account[1]. This only could be done by on line identification.

To identify nonlinear systems, dynamical neural networks can be used successfully [2,4,5]. They appear as powerful tools. Their complex architecture obviously gives better results and a more accurate model than the identification by linear models. However because of their complex architecture, huge calculations are often necessary. A second drawback will be the long training time required to obtain a good adaptation of the weights, especially with layered networks and backpropagation learning algorithm. This shortcoming is generally inappropriate to identify a time varying system. Simplified architectures like flat functional-link networks and fast learning algorithms based on the recursive least-square method have been proposed to improve neural networks performances. Other algorithms based on the Lyapunov stability theory have been widely studied to guarantee stability of the weight adaptive law [6, 7, 8]. However the main difficulty is to design both a fast identification method and a robust control law.

The method for getting the model of the process in real time is explained in the first Section. In the second part we develop a control law derived from the linear model. The control law is defined such as errors between the desired and predicted trajectory of outputs are minimised. The trajectory tracking is accurate and errors decay towards zero in spite of some important disturbances.

2. Identification

It is well-known the fact that many physical systems may be considered linear about an operating point for small changes of the system state. Under this assumption, a continuous nonlinear physical system can be described by a linear model in the neighbourhood of the considered operating point. A convenient way to represent a linear dynamical system is to use an autoregressive model *AR(p)*. The period of delay and the number of delays p are defined with respect to the process dynamics.Thus, we assume that the process can be described at an operating point (Y_{ope}, X_{ope}, U_{ope}) by the multi-input multi-output (MIMO) system of discrete equations :

$$(Y_{k+1} - Y_{ope}) = J_x(X_k - X_{ope}) + J_u(U_k - U_{ope}) \qquad (1)$$

where,

Y_{k+1} denotes the outputs of the process at time instant $k+1$,

X_k is a regression vector of Y at successive time intervals k, $k-1$, $k-nbry+1$.

U_k is a regression vector of the control input u at discrete time k, $k-1$, $k-nbru+1$.

$$X_k = \begin{bmatrix} Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix} \text{ and } U_k = \begin{bmatrix} u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix}$$

nbry denotes the order of regression for vector X_k.

nbru is the order of regression of U_k.

J_x and J_u are the matrices of the linearized model.

The number of inputs is defined as *nbu* and the number of outputs is *nby*.

Setting,

$$\Delta Y = Y_{k+1} - Y_{ope}$$

$$\Delta X = X_k - X_{ope}$$

$$\Delta U = U_k - U_{ope}$$

the model can be rewritten as

$$\Delta Y = J_x \Delta X + J_u \Delta U \quad (2)$$

The parameters J_x and J_u of this model must be computed such as the model gives a good approximation of the plant behaviour. We assume that the approximation is correct if the measures (Y_{k+1}, X_k, U_k) verify the equations of the model. To find suitable parameters J_x and J_u, we define a set of measures (Y_{set}, X_{set}, U_{set}) on a moving window, where the most recent measure replaces the oldest one. The size of the data window is constant and is equal to *nbm*. We have to determine the number of measures to accumulate such that there are enough data for an accurate identification. However to save computational time and resources, a too large data set is not advisable.

Moreover, to correctly achieve the identification, persistent excitation of the process is required. This condition means that the input-output signals must be sufficiently rich in frequency. In our case, this assumption cannot be imposed, because identification is achieved on -line. To overcome this difficulty, and try to keep sufficiently rich signals in matrices Y_{set}, X_{set} and U_{set}, a test is executed before inserting a new measure in the data set. A data set updating takes place if the distance *d1* between the desired and actual outputs is greater than a threshold d_{ide}, or if the distance *d2* between the outputs Y_k and the last vector of the data set is greater than d_{ide}. This threshold d_{ide} is defined according to the desired accuracy. In other words, the data set is updated if the outputs vary sufficiently or if the error is important, which may occur for instance during initialisation or because of large parameters variations.

The distances *d1*, *d2* are defined as,

$$d_1 = \left\| Y_{d,k} - Y_k \right\| \quad (3)$$

$$d_2 = \left\| Y_k - Y_{set,nbm} \right\| \quad (4)$$

with $Y_{d,k}$ the desired outputs, Y_k the system's outputs at discrete time k, and $Y_{set,nbm}$ the last measure added to the data set Y_{set}.

Since we wish to identify a local model of the process, we need to define the point where the identification is performed. This point called identification point (Y_{ide}, X_{ide}, U_{ide}) is defined as the mean vector of the data matrices (Y_{set}, X_{set}, U_{set}). We deduce the centred data set ΔY_{set}, ΔX_{set} and ΔU_{set} which is computed as follows :

$$\Delta Y_{set} = Y_{set} - Y_{ide} 1_{nbm}^T \quad (5)$$

$$\Delta X_{set} = X_{set} - X_{ide} 1_{nbm}^T \quad (6)$$

$$\Delta U_{set} = U_{set} - U_{ide} 1_{nbm}^T \quad (7)$$

with 1_{nbm} a vector of *nbm* rows with ones, and T the transpose operator.

The identification consists in minimising errors between ΔY_{set} and $J_x \Delta X_{set} + J_u \Delta U_{set}$ by adapting the matrices J_x and J_u by ΔJ_x and ΔJ_u respectively.

The adaptation for each row of J_x and J_u can be treated separately. Then, for the row i of the model, we have the equation:

$$\Delta Y_i = J_{x,i}\Delta X + J_{u,i}\Delta U \qquad (8)$$

where the subscript i denotes the row i of the matrices J_x and J_u, and the element i of the vector ΔY.

The errors between the model and measures are defined as :

$$E_{set,i} = \Delta Y_{set,i} - J_{x,i}\Delta X_{set} - J_{u,i}\Delta U_{set} \qquad (9)$$

The variations of the model modify the errors by $\Delta E_{set,i}$:

$$\Delta E_{set,i} = -\Delta J_{x,i}\Delta X_{set} - \Delta J_{u,i}\Delta U_{set} \qquad (10)$$

To minimise errors and to prevent too large variations of $J_{x,i}$ and $J_{u,i}$, we define the following quadratic criterion H.

$$H = \frac{1}{2}((E_{set,i} + \Delta E_{set,i}) + (E_{set,i} + \Delta E_{set,i})^T + \Delta J_{x,i} R_x \Delta J_{x,i}^T + \Delta J_{u,i} R_u \Delta J_{u,i}^T) \qquad (11)$$

with R_x and R_u, two positive definite diagonal matrices.

Then, we have to solve a familiar least-squares problem for which the optimal solution is such as :

$$\frac{\partial H}{\partial \Delta J_{x,i}} = 0,\ \frac{\partial H}{\partial \Delta J_{u,i}} = 0,\ \frac{\partial^2 H}{\partial \Delta J_{x,i}^{\ 2}} > 0 \text{ and } \frac{\partial^2 H}{\partial \Delta J_{u,i}^{\ 2}} > 0.$$

These conditions lead to :

$$\begin{bmatrix} R_x + \Delta X_{set}\Delta X_{set}^T & \Delta X_{set}\Delta U_{set}^T \\ \Delta U_{set}\Delta X_{set}^T & R_u + \Delta U_{set}\Delta U_{set}^T \end{bmatrix} \begin{bmatrix} \Delta J_{x,i}^T \\ \Delta J_{u,i}^T \end{bmatrix} = \begin{bmatrix} \Delta X_{set} \\ \Delta U_{set} \end{bmatrix} E_{set,i}^T \qquad (12)$$

To find the new matrices $J_{x,i}$ and $J_{u,i}$, we have to solve a linear system equation.

For convenience, we set :

$$A = \begin{bmatrix} R_x + \Delta X_{set}\Delta X_{set}^T & \Delta X_{set}\Delta U_{set}^T \\ \Delta U_{set}\Delta X_{set}^T & R_u + \Delta U_{set}\Delta U_{set}^T \end{bmatrix}$$

$$x_i = \begin{bmatrix} \Delta J_{x,i}^T \\ \Delta J_{u,i}^T \end{bmatrix} \text{ and } b_i = \begin{bmatrix} \Delta X_{set} \\ \Delta U_{set} \end{bmatrix} E_{set,i}^T.$$

Hence, the system can be rewritten as :

$Ax_i = b_i$, with A a symmetric positive definite matrix.

Since the input and output scales can be very different and because of the roundoff errors and the machine precision, we need improve the accuracy of the solution x_i, by scaling A and b_i. The scaled system must be equivalent to the former, and is defined such that the diagonal elements of the scaled matrix A_{scale} are equal to 1.

The transformation of the system $Ax_i = b_i$ in $DADD^{-1}x_i = Db_i$

$$\text{with } D = \begin{bmatrix} \sqrt{A_{1,1}} & & & \\ & \sqrt{A_{i,i}} & & \\ & & \cdots & \\ & & & \sqrt{A_{n,n}} \end{bmatrix}^{-1}$$

yields to the following equivalent system

$$A_{scale}x_{scale} = b_{scale} \qquad (13)$$

with

$$A_{scale} = DAD$$

$$x_{scale} = D^{-1}x_i$$

$$b_{scale} = Db_i$$

By definition of the matrix A, the diagonal elements are all greater than or equal to the values R_x or R_u. These parameters allow us to compute D and to avoid an ill-conditioning problem. By solving the system $A_{scale}x_{scale} = b_{scale}$ with a Cholesky factorisation, a forward elimination and a backward substitution [3], we deduce the least square solution $x_i = Dx_{scale}$. The Cholesky algorithm is time efficient and ensures numerical robustness.

The new matrices $J_{x,i}$ and $J_{u,i}$ of the model are

$$\left[J_{x,i} J_{u,i}\right]_{new} = \left[J_{x,i} J_{u,i}\right] + x_i^T \qquad (14)$$

The adaptation for the *nby* rows of J_x and J_u are obtained by solving $Ax_i = b_i$ for the *nby* vectors x_i and b_i. .

Because of the noise on measure, the filtering of data with a matrix F can significantly improve the results by reducing the noise effect.

$$\Delta Y_{set} F = J_x \Delta X_{set} F + J_u \Delta U_{set} F \qquad (15)$$

The simplest method consists in computing the average of *nbf* measures to obtain the filtered measures. In this case F is defined as a *nbm* by *nbm-nbf* matrix :

$$F = \frac{1}{nbf} \begin{bmatrix} 1 & & \\ \vdots & 1 & \\ 1 & \vdots & \ddots \\ & 1 & & 1 \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

The initialisation of J_x and J_u with predetermined values improves the start-up and provides better results at the beginning of the control. Furthermore, during the first *nbm* steps, a small sinusoidal control is applied to the system such that the data set is initialised with non zero values. After this preliminary identification, the control is switched on.

3. Control Law

The goal of the control law is to reduce errors between the desired and actual responses to the smallest possible value. The desired trajectory at future instants ($Y_{d,k+1}$, $Y_{d,k+2}$...) is assumed to be unknown. Consequently we have to determine it from the elapsed desired outputs. We must also estimate the future outputs of the process from the identified model and the past measures.

The future desired outputs are defined at discrete time $k+1$, $k+2$, ..., $k+h$ by an extrapolation from the earlier desired outputs. To smooth the trajectory, we estimate the slope ΔY_d between $Y_{d,k}$ and $Y_{d,k-n}$. Thus, the desired trajectory is computed as follows :

$$\Delta Y_d = \frac{Y_{d,k} - Y_{d,k-n}}{n} \qquad (16)$$

$$Y_{d,k+i} = Y_{d,k+i-1} + \Delta Y_d \quad \text{for } i = 1 \text{ to } h. \qquad (17)$$

Therefore the future desired trajectory is

$$Y_{d,fut} = \begin{bmatrix} Y_{d,k+h} \\ \vdots \\ Y_{d,k+1} \end{bmatrix}$$

The prediction of the future trajectory of the system is one of the key problems. It is necessary to accurately predict the state of the process. The prediction of the outputs derives from the model. On the one hand, it depends on the past values of outputs and inputs, and obviously on the other hand on the future control inputs, we want to optimise. Due to modelling errors, we cannot forecast the system outputs with reasonable errors on many steps into the future. The optimisation is performed with respect to only h predictions, with h a small integer (from 4 up to 10 for instance).

From the model (1), we derive the equation

$$Y_{k+1} = J_x X_k + J_u U_k + v_{ope} \qquad (18)$$

which can also be written

$$y_{k+1} = \left[J_{x,1} \cdots J_{x,nbry}\right] \begin{bmatrix} y_k \\ \vdots \\ y_{k-nbry+1} \end{bmatrix} + \left[J_{u,1} \cdots J_{u,nbru}\right] \begin{bmatrix} u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix} + v_{ope} \qquad (19)$$

$$\text{with } v_{ope} = Y_{ope} - J_x X_{ope} - J_u U_{ope} \qquad (20)$$

Y_{ope}, X_{ope}, and U_{ope} are the operating points readily obtained from measurements with a first order filter :

$$Y_{ope,k+1} = \alpha Y_{ope,k} + (1-\alpha) Y_k$$
$$X_{ope,k+1} = \alpha X_{ope,k} + (1-\alpha) X_k$$
$$U_{ope,k+1} = \alpha U_{ope,k} + (1-\alpha) U_k$$

with $\alpha = 0.9$

Using h times the model (19), we deduce the following prediction.

$$\begin{bmatrix} Y_{k+h} \\ Y_{k+h-1} \\ \vdots \\ Y_{k+2} \\ Y_{k+1} \end{bmatrix} = \left[\begin{array}{ccccc|cccc} 0 & J_{x,1} & \cdots & J_{x,nbry} & & & & & \\ & 0 & J_{x,1} & \cdots & J_{x,nbry} & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & 0 & J_{x,1} & \cdots & J_{x,nbry} & & \\ & & & & 0 & J_{x,1} & \cdots & J_{x,nbry} & \end{array}\right] \begin{bmatrix} Y_{k+h} \\ \vdots \\ \vdots \\ Y_{k+2} \\ Y_{k+1} \\ \hline Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix} +$$

$$+ \left[\begin{array}{cccc|cccc} J_{u,1} & \cdots & J_{x,nbru} & & & & & \\ & J_{u,1} & \cdots & J_{x,nbru} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & J_{u,1} & \cdots & J_{x,nbru} & & \\ & & & & J_{u,1} & \cdots & J_{x,nbru} & \end{array}\right] \begin{bmatrix} u_{k+h-1} \\ \vdots \\ u_{k+2} \\ u_{k+1} \\ \hline u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix} + \begin{bmatrix} v_{ope} \\ v_{ope} \\ \vdots \\ v_{ope} \\ v_{ope} \end{bmatrix} \qquad (21)$$

For convenience, the previous equation can be rewritten :

$$Y_{fut} = J_{x,fut} Y_{fut} + J_{u,fut} U_{fut} + J_{x,past} Y_{past} + J_{u,past} U_{past} + V_{ope} \qquad (22)$$

with

$$Y_{fut} = \begin{bmatrix} Y_{k+h} \\ \vdots \\ Y_{k+1} \end{bmatrix}, U_{fut} = \begin{bmatrix} u_{k+h-1} \\ \vdots \\ u_{k+1} \end{bmatrix}, U_{past} = \begin{bmatrix} u_k \\ \vdots \\ u_{k-nbru+1} \end{bmatrix}, Y_{past} = \begin{bmatrix} Y_k \\ \vdots \\ Y_{k-nbry+1} \end{bmatrix}, V_{ope} = \begin{bmatrix} v_{ope} \\ \vdots \\ v_{ope} \end{bmatrix}$$

$$J_{x,fut} = \begin{bmatrix} 0 & J_{x,1} & \cdots & J_{x,nbry} & \\ & 0 & J_{x,1} & \cdots & J_{x,nbry} \\ & & \ddots & \ddots & \ddots \\ & & & 0 & J_{x,1} \\ & & & & 0 \end{bmatrix}, J_{x,past} = \begin{bmatrix} & & \\ 0 & & \\ J_{x,nbry} & 0 & \\ \vdots & \ddots & 0 \\ J_{x,1} & \cdots & J_{x,nbry} \end{bmatrix}$$

$$J_{u,fut} = \begin{bmatrix} J_{u,1} & \cdots & J_{x,nbru} & 0 \\ & J_{u,1} & \cdots & J_{x,nbru} \\ & & \ddots & \ddots \\ & & & J_{u,1} \\ & & & 0 \end{bmatrix}, J_{u,past} = \begin{bmatrix} & & \\ 0 & & \\ J_{u,nbry} & 0 & \\ \vdots & \ddots & 0 \\ J_{u,1} & \cdots & J_{u,nbry} \end{bmatrix}$$

The subscripts *fut* and *past* denote respectively the future and the past values.

Setting $V_{past} = J_{x,past} Y_{past} + J_{u,past} U_{past} + V_{ope}$, and solving for Y_{fut}, we obtain :

$$Y_{fut} = \left(I - J_{x,fut}\right)^{-1} \left(J_{u,fut} U_{fut} + V_{past}\right) \quad (23)$$

For calculation ease, we set $K_{x,fut} = \left(I - J_{x,fut}\right)^{-1}$.

The matrix $K_{x,fut}$ is an upper triangular matrix, which can be constructed with the following formula :

$$I - J_{x,fut} = \begin{bmatrix} I & -J_{x,1} & \cdots & -J_{x,nbry} & & \\ & I & -J_{x,1} & \cdots & -J_{x,nbry} & \\ & & \ddots & \ddots & \ddots & \\ & & & I & -J_{x,1} & \\ & & & & I & \end{bmatrix}, K_{x,fut} = \begin{bmatrix} I & K_1 & K_2 & \cdots & K_n \\ & I & K_1 & \cdots & K_{n-1} \\ & & \ddots & & \vdots \\ & & & I & K_1 \\ & & & & I \end{bmatrix}$$

The solution of the matrices K_i are

$$K_1 = J_{x,1},$$

$$K_i = \sum_{k=1}^{i-1} K_{i-k} J_{x,k} + J_{x,i} \quad \text{for } i \le \text{nbry},$$

$$K_i = \sum_{k=1}^{nbry} K_{i-k} J_{x,k} \quad \text{for } i > \text{nbry},$$

The expression of the predicted outputs becomes :

$$Y_{fut} = K_{x,fut} \left(J_{u,fut} U_{fut} + V_{past}\right) \quad (24)$$

Since the optimisation of the *nbu x h* vector U_{fut} leads to numerous calculations, we specify the following constraint on U_{fut}.

$$U_{fut} = \begin{bmatrix} u_k + h\delta u \\ \vdots \\ u_k + 2\delta u \\ u_k + \delta u \end{bmatrix} \quad (25)$$

with u_k the control at instant k and δu the variation of the control for the next control inputs.

Then, instead of solving a large system to evaluate the whole vector U_{fut}, the problem is to find the vector δu.

With this assumption, the prediction is given by :

$$Y_{fut} = K_{x,fut} J_{u,fut} \begin{bmatrix} hI \\ \vdots \\ 2I \\ I \end{bmatrix} \delta u + K_{x,fut} \left(V_{past} + J_{u,fut} \begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix} \right) \quad (26)$$

To yield the desired response, we establish the criterion G with a first term defined to minimise the errors between $Y_{d,fut}$ and Y_{fut}. The second part of G is defined such that errors decay progressively. It improves the response which becomes smoother and less oscillatory. To obtain full rank matrices and ensure numerical robustness, a third term is added to G.

Then, the criterion G is

$$G = \frac{1}{2}\left(e_{fut}^T Q e_{fut} + \Delta e_{fut}^T \Delta e_{fut} + \delta u^T R_u \delta u\right) \quad (27)$$

where e_{fut} and $\Delta_{e,fut}$ are

$$e_{fut} = Y_{d,fut} - Y_{fut} = Y_{d,fut} - K_{x,fut}\left(V_{past} + J_{u,fut} \begin{bmatrix} U_k \\ \vdots \\ U_k \end{bmatrix}\right) - K_{x,fut} J_{u,fut} \begin{bmatrix} hI \\ \vdots \\ 2I \\ I \end{bmatrix} \delta u$$

$$\Delta e_{fut} = \begin{bmatrix} e_{fut,k+h} - e_{fut,k+h-1} \\ \vdots \\ e_{fut,k+2} - e_{fut,k+1} \\ e_{fut,k+1} - e_k \end{bmatrix} =$$

$$\left[\begin{array}{ccc|c} I & -I & & \\ & \ddots & \ddots & \\ & & I & -I \\ \hline & & & I \;\; -I \end{array}\right] \left[\begin{array}{c} e_{fut,k+h} \\ \vdots \\ e_{fut,k+1} \\ \hline e_k \end{array}\right]$$

Q and R_u are positive definite diagonal matrices, which are respectively weighting factors on the speed of convergence and the variation δu. The weighting factor R_u must be small, but greater than the machine precision. For single precision numbers, R_u is taken equal to $10^{-6}\ I$. Q is defined such that we obtain a good trade-off between the precision of the trajectory tracking and the robustness with respect to the noise effect, the modelling errors and parameters changes. Indeed for a small value of Q ($Q=0.01\ I$), the transient response is long, but the sensitivity to noise is small, whereas for $Q=I$ errors decrease at a high rate. So the effect of disturbances is quickly reduced and the time to reach a steady-state is short. But the response is more oscillatory and the magnitude and variations of the control signal are higher, so it requires more energy. We encounter a very common trade-off situation. In Section 3, satisfactory results are obtained with $Q = 0.3\ I$.

Setting,

$$\Delta e_{fut} = \begin{bmatrix} P_{11} & 0 \\ P_{21} & -I \end{bmatrix} \begin{bmatrix} e_{fut} \\ e_k \end{bmatrix},$$

G is rewritten as

$$G = \frac{1}{2}(e_{fut}^T Q e_{fut} + e_{fut}^T P_{11}^T P_{11} e_{fut} + e_{fut}^T P_{21}^T P_{21} e_{fut} - e_{fut}^T P_{21}^T e_k - e_k^T P_{21} e_{fut} + e_k^T e_k + \delta u^T R_u \delta u)$$

For convenience, we rewrite e_{fut} and define the matrix R_e as follows :

$$R_e = Q + P_{11}^T P_{11} + P_{21}^T P_{21}$$

$$e_{fut} = V - M\delta u$$

with

$$V = Y_{d,fut} - K_{x,fut}\left(V_{past} + J_{u,fut}\begin{bmatrix} u_k \\ \vdots \\ u_k \end{bmatrix}\right) \text{ and}$$

$$M = K_{x,fut} J_{u,fut} \begin{bmatrix} hI \\ \vdots \\ 2I \\ I \end{bmatrix}$$

The criterion G becomes

$$G = \frac{1}{2}((V - M\delta u)^T R_e (V - M\delta u) - (V - M\delta u)^T P_{21}^T e_k - e_k^T P_{21}(V - M\delta u) + e_k^T e_k + \delta u^T R_u \delta u)$$

The optimal solution δu is such that

$$\frac{\partial G}{\partial \delta u} = 0 \text{ and } \frac{\partial^2 G}{\partial \delta u^2} > 0$$

Hence, the equation to solve is

$$\left(M^T R_e M + R_u\right)\delta u = M^T R_e V - M^T P_{21}^T e_k \tag{28}$$

By the definition of matrices M and P_{21}, we have $M^T P_{21}^T = 0$

So the optimal δu is a solution of the linear system

$$\left(M^T R_e M + R_u\right)\delta u = M^T R_e V \tag{29}$$

As before in Section 1, the system is scaled to improve the accuracy of results. Then the Cholesky factorisation, the forward elimination and the backward substitution are achieved to find the solution δu.

Given the physical limits of actuators, and in order to protect the process against damages due to possibly huge values of control, it is necessary to limit the control inputs between a lower limit and an upper limit. The value of this saturation depends on the characteristics of actuators and the system under consideration.

The noise on measurements can be neglected either . To reduce its effect, one can filter output and input signals. However filters generally introduce another dynamics and a delay which often alters wrongly the control law and in the worst case can determine a loss of stability. To prevent such a drawback, we consider that the filter can be used only when the variations of the desired outputs and errors are lower than the desired accuracy Y_{acc}. Furthermore, instead of filtering the output Y_k, we have applied a filter on the control variation δu. The filter has been realized by multiplying δu by a factor α defined as follows :

$$\alpha = 1 \qquad \text{if } \left\|Y_{d,k} - Y_k\right\| + \left\|\Delta Y_d\right\| \geq Y_{acc}$$

$$\alpha = \left(\frac{\left\|Y_{d,k} - Y_k\right\| + \left\|\Delta Y_d\right\|}{Y_{acc}} \right)^{\frac{1}{2}}$$

$$\text{if } \left\|Y_{d,k} - Y_k\right\| + \left\|\Delta Y_d\right\| < Y_{acc}$$

With this factor α, no undesirable effect is introduced when errors or variations of the desired trajectory are large. But, in a steady-state, the effect of noise on the control will be reduced. So, we obtain a control law which is less sensitive to noise.

Then we deduce the control law :

$$u_{k+1} = sat(u_k + \alpha\delta u) \tag{30}$$

with sat the saturation function.

In case of such actuators as electric motors, we also have to take into account the dead zone of the device. Indeed, due to friction, motors accelerate only if the control input is greater than a threshold. To suppress the deadband nonlinearity, the value of control must be at least equal to the threshold. Consequently, we introduced an offset in the control law (Eq 31).

Thus the input values applied to the process actuators are :

$$u_{actuator,k+1} = sat(u_k + \alpha\delta u) + u_{offset}\, sign(u_k + \alpha\delta u) \tag{31}$$

4. Experimental Results

The proposed adaptive control law is applied to a mechanical process called TALC. This machine is designed to test the efficiency of control laws. The goal is to control the angular position of a mirror, such that a laser beam reflected by the mirror points at a desired location or trajectory on a screen.

Clearly, the function between the rotation angles of the mirror and the coordinates of the spot on the screen is nonlinear because of the reflection and projection transformations. Furthermore, the

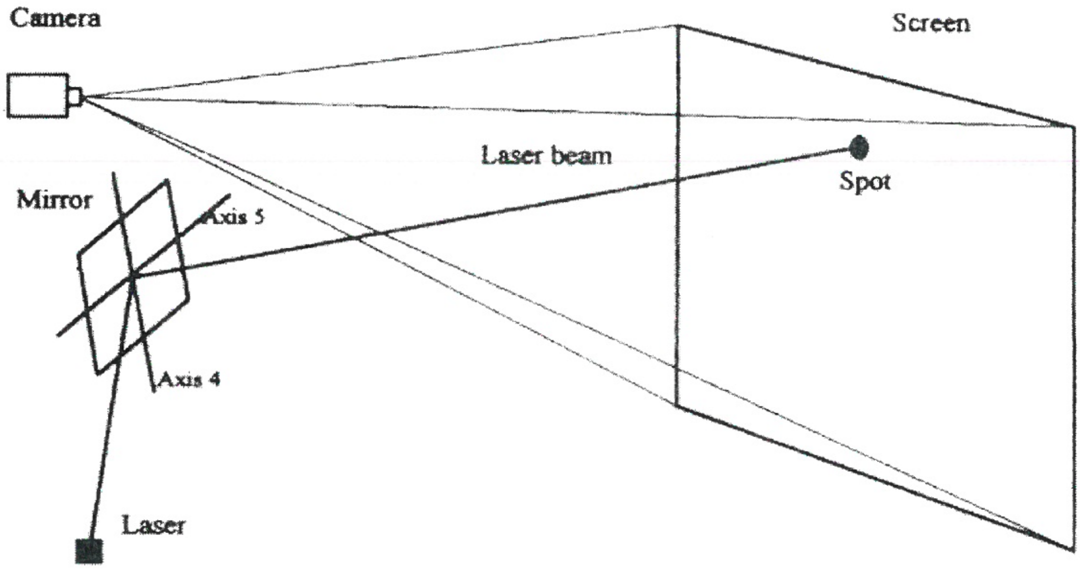

Figure 1. Overall Structure

coordinates depend on the distance between the mirror and the screen, which is considered as an unknown parameter. A camera takes pictures of the screen at more than 100 frames per second. Then, an image processing algorithm gives the measurement of the spot coordinates x_{cam} and y_{cam} (in pixel).The overall structure of the process is shown in Figure 1.

The mirror is set on a platform articulated around five axes as shown in Figure 2. The control inputs are the tension of motors 4 and 5 which rotate the mirror around axes 4 and 5. Motors 1, 2 and 3 cause disturbances on the angular position of mirror.

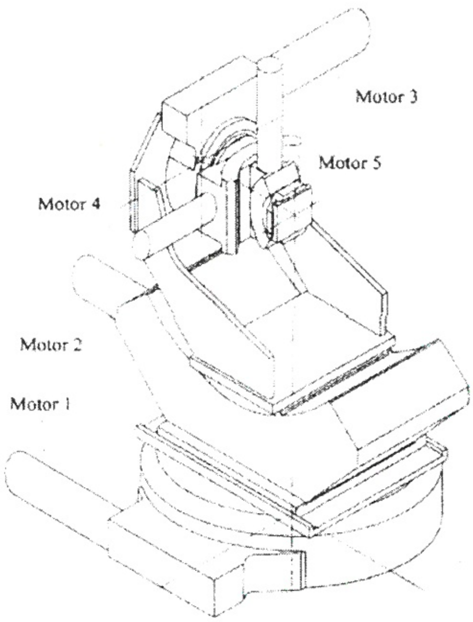

Figure 2. Platform TALC

The first step of the proposed scheme is to identify the function between the spot coordinates (outputs) and the control inputs. This function gives an approximation of the process behaviour, which is used to deduce the control to apply on motors 4 and 5. The process is assumed to be a second order system with a sampling period taken equal to 0.01s. It is worth noting that the only other known parameters of the process are the maximal value of control and the dead zone of actuators. The control offsets for motors 4 and 5 are respectively 0.02 V and 0.08 V. The number of measures in the data set is 50, and the threshold d_{ide} used to test if a measure must be inserted into the data set, is 8 pixels. The desired accuracy of tracking is 4 pixels. The reference trajectory is a cycle of 6 s, defined by two sinusoidal functions :

$$x_{desired} = 60\cos(\frac{2pi}{3}t)$$

$$y_{desired} = 60\cos(\frac{2pi}{6}t)$$

The tracking error at instant k is defined as the distance between the actual and desired position of the spot.

$$e_k = \sqrt{(x_{desired} - x_{cam})^2 + (y_{desired} - y_{cam})^2} \quad (32)$$

The desired and actual trajectories without any disturbance are shown in Figure 3. The mean and maximal values of the errors are only about 2 and 6 pixels. The plots of control inputs for motors 3 and 4 are shown in Figure 3. The noise effect on the control inputs can be reduced by choosing in the criterion G a smaller value of Q, but in this case the tracking is less accurate. A good trade-off between the sensitivity to noise and the accuracy is obtained with $Q=0.31$.

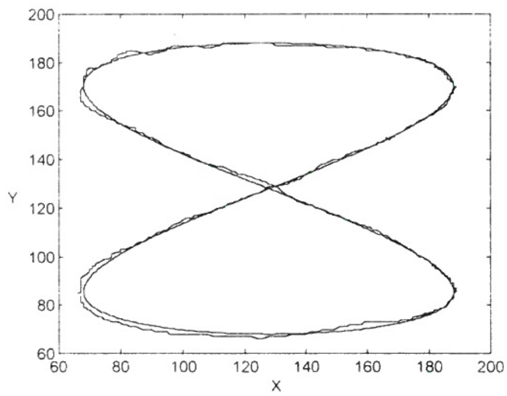

Figure 3. Desired and Actual Trajectories Without Any Disturbance

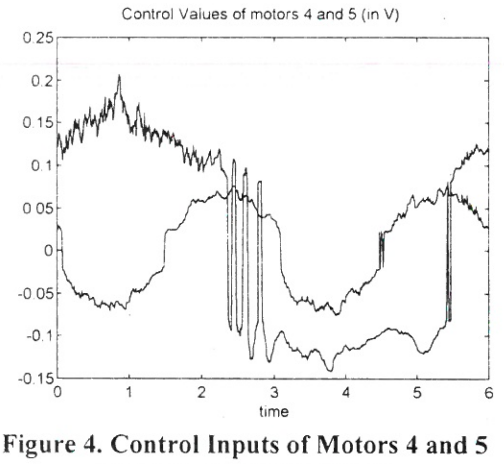

Figure 4. Control Inputs of Motors 4 and 5 Without Any Disturbance

Disturbances are introduced on axes 1, 2 and 3. Their effects on the spot, with the controller switched off, are shown in Figure 5. The spot describes a cycle in 3 seconds. The variations on axes X and Y are about 90 and 60 pixels.

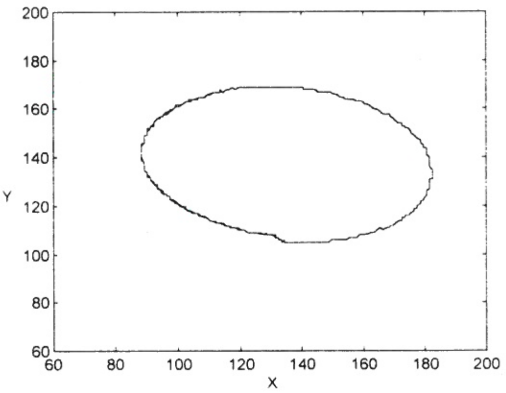

Figure 5. Effect of Disturbances on the Spot Position

The results with the controller switched on and these disturbances are shown in Figure 6. In spite of the disturbances, the tracking is accurate. The mean value of errors is about 2.5 pixels and the maximal error is equal to 8 pixels. The effect of disturbance signals is almost completely eliminated. Only small deviations and oscillations of the spot can be observed.

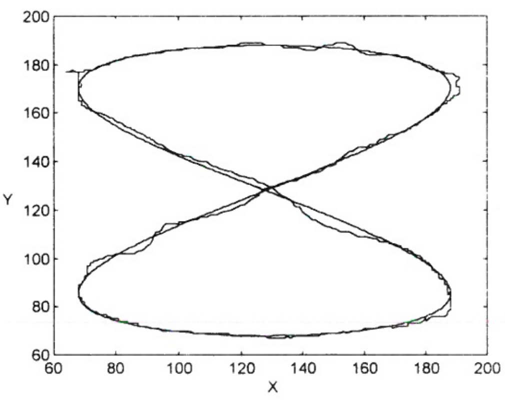

Figure 6. Desired and Actual Trajectories With Disturbances

Accuracy and rejection of perturbations are satisfactory. This experiment highlights the performances and robustness of the controller and its ability to reduce the effect of disturbances.

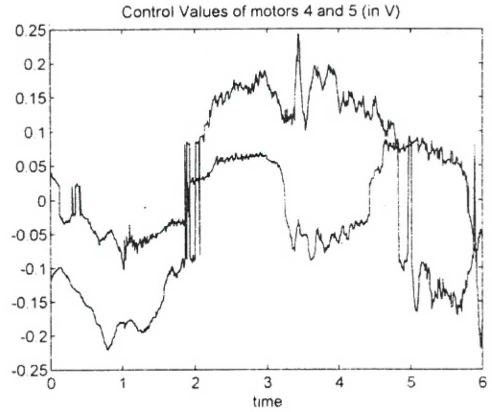

Figure 7. Control Inputs of Motors 4 and 5 With Disturbances

5. Conclusion

In control system theory, robust and adaptive methods are very attractive. With the proposed method, there is no need for a perfect knowledge of the system to deduce the control law. Only basic information about the process and the features of actuators is necessary to define the controller parameters. On-line identification with a least-squares approach provides a linear model of the process about the operating point. The adaptation law of the model is fast, and consequently this method can be effectively applied to a large class of nonlinear and time varying processes.

The controller is established from the model and defined to track a reference trajectory with the desired accuracy and celerity. The control criterion provides a robust control law, which reduces the effect of disturbances and sensor noise. Finally, experimental results emphasize the efficiency of the proposed adaptive control.

REFERENCES

1. BISHOP, R.H., **Adaptive Control of Space Station With Control Moment Gyros**, IEEE CONTROL SYSTEMS, Vol. 12, No 5, October 1992.

2. HUNT, K., SHRBARO, D., ZBIKOWSKI, R. AND GAUTHROP, P.J., **Neural Networks for Control Systems - A Survey**, AUTOMATICA, Vol. 28, No.6, 1992, pp. 1083-1112.

3. GOLUB, G.H. AND VAN LOAN,C.F., **Matrix Computations**, JOHNS HOPKINS UNIVERSITY PRESS, Baltimore, USA, 1989.

4. OHBAYASHI, M., HIRASAWA, K., HASHIMOTO, M. and MURATA, J., **Robust Using Second Order Derivative of Universal Learning Network**, Proceedings of IEEE International Conference on Systems Man and Cybernetics, Vol. 2, October 1996, pp. 1184-1189.

5. NARENDRA, K. S. and PARTHASARATHY, K., **Identification and Control of Dynamical Systems Using Neural Networks**, IEEE TRANSACTIONS ON NEURAL NETWORKS, Vol.1, No. 1, March 1990.

6. ROVITHAKIS,G.A. and CHRISTODOULOU, M.A., **Adaptive Control of Unknown Plants Using Dynamical Neural Networks**, IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS, Vol. 24, No.3, March 1994.

7. ROVITHAKIS,G.A. and CHRISTODOULOU, M.A., **Direct Adaptive Regulation of Unknown Nonlinear Systems Via Dynamic Neural Networks**, IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS, Vol. SMC-25, No.12, December 1995, pp. 1578-1594.

8. SADEGH, N., **A Nodal Link Perceptron Network With Applications To Control of A Nonholonomic System**, IEEE TRANSACTIONS ON NEURAL NETWORKS, Vol.6, No.6, November 1995.

Régis Canart graduated in 1995 from the Engineering School ICAM in Lille, France. He is currently a researcher at the Laboratory of Automatic Control and Informatics of Lille (LAIL), where he is achieving his Ph.D thesis. His scientific interests are system identification, adaptive control and neural networks for control applications.

Professor Pierre Borne graduated from the University of Lille in 1967 and obtained the Master degrees in Applied Mathematics, Physics, Mechanics and Electronics in 1968. In 1968 he also obtained the Diploma of Engineer of the IDN (French "Grande Ecole" with the new name "Ecole Centrale de Lille"). He obtained the Ph.D degree in Automatic Control in 1970 and Dsc. of Physics in 1976.
He is now Professor of Automatic Control, head of the Automatic Control Department and Scientific Director of Ecole Centrale de Lille.
He is involved in various international activities with IEEE/SMC Society and with IMACS. In 1996 he became Fellow IEEE.

His research interests include stability analysis of non-linear systems, and various aspects of robust control of non-linear and complex systems.