Applications of Adaptive Control Based on Nonlinear Static Characteristic

CIPRIAN LUPU, DUMITRU POPESCU, ANDREEA UDREA
Department of Automatic Control and Computers Science
University “POLITEHNICA” of Bucharest
313, Splaiul Independentei, sect.6, Bucharest
ROMANIA

Abstract: This paper proposes a real-time control algorithm based on a model reference adaptive control mechanism for a class of nonlinear systems. The reference model is computed off-line, while the controller’s parameters are determined on-line, using a static gain adaptation criterion. The relevance of this approach was tested on several nonlinear processes, using a real-time applications package developed in LabWindows/CVI. A solution designed for a class of nonlinear systems, based on parametric adaptation for an RST controller is presented. Issues on the stability of the solution are also discussed.

Key-Words: model reference adaptive control, adaptive law, real-time application

1 Introduction
The nonlinear systems control is one of the greatest challenges of modern times. One of the solutions for this problem is found in the area of adaptive systems, which in the last 20 years, due to computational advancement, were largely investigated [1], [2], [5]. Adaptive control encountered many challenges in the field of real-time systems, which do not have precise classical models admissible to existing control designs[4].

The essential condition for the real-time control systems is preserving the closed-loop performances in case of non-linearity, structural disturbances or process uncertainties.

There are several adaptive design approaches that work for different types of nonlinear systems: model reference adaptive control, adaptive backstepping control etc [1],[4].

This paper proposes a real-time control algorithm based on a model reference adaptive control mechanism for a class of nonlinear systems. The reference model is computed off-line, while the controller’s parameters are determined on-line, via a static gain adaptation criterion.

For a real time implementation, the algorithm can be divided into two independent parts:

- **the identification algorithm** is used in order to obtain the static model of the plant (Fig. 1)
  - this step gives the actual gain of the plant;
- **controller parameters’ adaptive law** (Fig. 2)
  - which recalculates the controller’s parameters.

![Fig. 1. Identification algorithm used to determine the static model of the plant](image)

![Fig. 2. Adaptive law influence on the controller’s parameters](image)

The proposed structure is based on a series of steps that are made before the real-time running of the application:

- identification and storage of the static characteristic of the process;
- important inflexion points selection on the static characteristic;
• identification of the process model (dynamical) in a randomly chosen functioning point;
• design of the regulation algorithm (PID, RST) - based on the dynamic model.

In the next section, the most important steps are presented.

2 Adaptive algorithm - off-line design

In this section, the actions that take place off-line (before the real-time running of the application), will be explained.

2.1 Plant’s static model determination

This operation is based on several experiments of discrete step increasing and decreasing of the command \( u(k) \) and measuring the corresponding stabilized process output \( y(k) \). The command \( u(k) \) covers all possibilities: 0 to 100% (in percentage representation). Because the process is disturbed by noise, usually, the static characteristics are not identically. The final static characteristic is obtained by meaning of correspondent position of these experiments. Figure 3 presents this operation. The graphic between two “mean” points can be obtained using the extrapolation procedure.

![Fig. 3. Determination of static characteristic of the process (in continuous line - final characteristic)](image)

According to system identification theory, the dispersion of process trajectory can be found using expression (1):

\[
\sigma^2[n] = \frac{1}{n-1} \sum_{i=1}^{n} y^2[i], \quad \forall n \in N' \setminus \{1\} \tag{1}
\]

This can express a measure of superposing of noise onto the process, process nonlinearities and so on, and is very important on control algorithm designed robustness. Another option is to find the position and the value \( mg \) of the maximal distance from the “mean” characteristic.

2.2 Control law design

Control algorithm’s purpose is to eliminate the disturbance and differences between real process’s gain and estimated gain.

![Fig. 4. RST control algorithm structure](image)

A large variety of control algorithms can be used here, PID, RST, fuzzy [2].

For this study, we use a RST algorithm. This is designed using a pole placement procedure [3]. Figure 4 presents the RST algorithm.

The R, S, T polynomials are:

\[
\begin{align*}
R(q^{-1}) &= r_0 + r_1 q^{-1} + \ldots + r_m q^{-nr} \\
S(q^{-1}) &= s_0 + s_1 q^{-1} + \ldots + s_m q^{-ns} \\
T(q^{-1}) &= t_0 + t_1 q^{-1} + \ldots + t_m q^{-nt}
\end{align*}
\] \tag{2}

The algorithm using pole placement design procedure is based on identified process’s model:

\[
y(k) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(k) \tag{3}
\]

where:

\[
\begin{align*}
B(q^{-1}) &= b_0 q^{-1} + b_1 q^{-2} + \ldots + b_m q^{-mb} \\
A(q^{-1}) &= 1 + a_1 q^{-1} + \ldots + a_m q^{-ma}
\end{align*}
\] \tag{4}

The identification is made in a specific process operating point and can use recursive least squares algorithm exemplified in the next relations and fully presented in [5]:

\[
\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1) \phi(k) e^o(k+1), \forall k \in N
\]

\[
F(k+1) = F(k) - \frac{F(k) \phi(k) \phi'(k) F(k)}{1 + \phi'(k) F(k) \phi(k)}, \forall k \in N
\] \tag{5}

\[
e^o(k+1) = y(k+1) - \hat{\theta}^o(k) \phi(t), \forall k \in N,
\]

with the following initial conditions:
The estimated \( \hat{\theta}(k) \) represents the parameters of the polynomial plant model and \( \phi^T(k) \) represents the measures vector.

This approach allows the users to verify, and if necessary, to calibrate the algorithm’s robustness. The following expression presents the disturbance-output sensitivity function.

\[
S_p(e^{j\omega}) = \frac{A(e^{j\omega})S(e^{j\omega})}{A(e^{j\omega})S(e^{j\omega}) + B(e^{j\omega})R(e^{j\omega})}, \quad \forall \omega \in R
\]

(7)

At the same time, the negative maximum value of sensitivity function represents the module margin. It is imposed that the gain’s limit to be greater or equal to the process static characteristic maximal distance \( \Delta G \geq mg \). For our application, a controller that has sufficient robustness was designed.

3 Adaptive algorithm - on-line design

This section presents and arguments the adaptive algorithm which consists in following phases:

- establish the plant’s gain; this is made considering the process position relatively to the inflexion points of the static characteristic. The gain is determined based on the inflexion point that is immediately inferior;
- gain filtering; is made using a very simple filter (first order). It is necessary in order to prevent the rough modification of the controllers parameters. A major change can lead to great discrepancies in the algorithm;
- RST controller adaptation; adaptation of the \( R \) and \( T \) polynomials coefficients in order to maintain the product between the plants gain and the controllers gain constant, so that the initial (off-line) closed-loop performances remain unchanged.

The adaptive structure based on a RST controller is presented in figure 5.

3.1 Plant’s gain

The plant’s gain is determined based on the inflexion point that is immediately inferior \( y_{i-1} \) and the process position at moment \( k - y_p \).

\[
K_{plant} = \frac{y_p - y_{i-1}}{u_p - u_{i-1}}
\]

(8)

3.2 Gain filtering

In order to prevent the rough modification of the controller’s parameters, the actual plant gain is filtered using a first order filter:

\[
C(q^{-1}) = \frac{C_1}{1 + C_2q^{-1}}
\]

(9)

The filtered gain is:

\[
K_f = \frac{C_1}{1 + C_2q^{-1}}K_{plant}.
\]

(10)

The usage of this filter is augmented by the following fact. Because:

\[
u(k) = \frac{1}{s_0}\left[-\sum_{i=1}^{m} s_i u(k-i) - \sum_{i=0}^{n} s_{j+i} y(k-i) + \sum_{i=0}^{n} s_i y^*(k-i)\right]
\]

(11)

is an expression that links the controller’s parameters to \( u(k), y(k) \) and \( y^*(k) \), it is a demand not to abruptly modify these coefficients. The filter attenuates possible brusque changes.

3.3 Adaptive law

The adaptive law compares the output of the system to the output values of precalculated model and modifies the controllers’ parameters in order to maintain the product between the controllers’ gain and the plants’ gain constant:
\[ K_{\text{controller}} \cdot K_{\text{plant}} = c t. \] (12)

That implies that the controller’s parameters are to be recalculated in order to satisfy relation (12). The justification for the parameters adaptation procedure is the following: the transfer function for the closed loop system has the form:

\[
H(q^{-1}) = \frac{T(q^{-1})B(q^{-1})}{1 + R(q^{-1})B(q^{-1})} \frac{S(q^{-1})A(q^{-1})}{S(q^{-1})A(q^{-1})} \] (13)

The model’s gain is \( K_{\text{plant}} = \frac{\sum a_i}{1 + \sum b_i} \). If the static characteristic varies on behalf of an inflexion point, we have a new \( K_{\text{plant}}' \) and we need to recalculate the value for the controller’s gain in order to satisfy the adaptive law.

The controller’s gain would be:

\[ K_{\text{controller}}' = K_{\text{controller}} \cdot \frac{K_{\text{plant}}'}{K_{\text{plant}}} = K_{\text{controller}} F \] (14)

where \( F \) is the applied correction factor.

From this and in order to maintain unchanged relation (13), \( \sum t_i / (1 + \sum s_i) \) and \( \sum r_i / (1 + \sum s_j) \) rapport must be multiplied by \( F \).

If we multiply the parameters \( t_i \) and \( r_i \) of the controller by \( F \), the correction factor, we satisfy the adaptive law.

These parameters modification do not influence the stability of the system.

4 Experimental results

In order to demonstrate the applicability of this solution, a nonlinear process consisting in a tank with a filling point and multiple evacuation points was chosen as an example. The purpose of the control system is to maintain constant the liquid level in the tank.

A process simulator has been created:

![Fig. 6. Tank with single filling-multiple evacuation points simulator](image)

The input-output characteristic for this system can be approximated by a set of medium input and output values and has the form showed in the next figure:

![Fig. 7. Input-output characteristic for this system](image)

In order to test the adaptive controller (figure 8), one must load this characteristic – model reference in the adaptive controller real-time software application and select a number of inflexion points, where the process dynamic changes.

![Fig. 8. Adaptive real-time controller](image)

On this characteristic, a set of inflexion points can be easily observed and learned, so that the new process’s gain and controller’s parameters will be calculated by using the adaptive law.
Accordingly to figure above, we’ve identified the following four functioning intervals (0-20%), (20-50%), (50-90%), respectively (90-100%).

In order to identify the model for the first interval a sampling period $T_e = 0.6$ sec was used and least-squares identification method from Adaptech/WinPIM platform was employed:

$$M(q^{-1}) = \frac{1.13044}{1 - 0.54787q^{-1}}$$

For this model, we have computed the corresponding RST control algorithm for the first region using a pole placement procedure and the Adaptech/WinREG platform. For the poles placement procedure we’ve used a second order system, defined by the dynamics $\omega_0 = 0.5$, $\xi = 0.95$ for tracking performances and $\omega_0 = 1.25$, $\xi = 0.8$ for disturbance rejection performances, keeping the same sampling period as for identification.

The obtained parameters for the first functioning interval are as it follows:

$$R(q^{-1}) = 0.494956 - 0.218212 q^{-1}$$

$$S(q^{-1}) = 1.000000 - 1.000000 q^{-1}$$

$$T(q^{-1}) = 0.884611 - 0.874307 q^{-1} + 0.266440 q^{-2}$$

These initial values for the RST controller, are loaded into the simulator, see Figure 9.

Using this application, a few tests were made in order to demonstrate the fact that the adaptive control mechanism that was implemented can guarantee the closed loop stability under changes of reference and disturbances influence.

We first impose a disturbance of 1% and we test the performances for changes of reference:

- from 10% (where the only pre calculated RST algorithm is active) to 40% (where, normally, because the static gain value changed the RST parameters were recalculated by the adaptive law)

- from 40% to 60% - figure 11;
- from 60% to 90% - figure 12;
- from 90% to 30% - figure 13.

As you can see, the tracking and rejection performances are quite good and the system remains stable.
The effective change of parameters is done when the filtered process output becomes greater than 20%, 50% and 90%.

In all tests, one can see that there are no shocks or oscillations in the control evolution by applying this approach.

Increasing the number of selected inflection points increase the performances if the characteristic substantially changes its form in that points.

5 Conclusion

The method was successfully tested on a nonlinear process using a controller simulator software application implementing the proposed reference model identification and adaptive law.

With regards to the results obtained in the paper, the adaptive method can be successfully recommended for real-time control structures for this class of nonlinear processes.

Acknowledgement

This work was partially supported by IDEI Program of Romanian Research, Development and Integration National Plan II, Grant no. 1044/2007.

References:
[1] Z. Bubnicki, Modern Control Theory, Springer-Verlag, 2005