Design Procedure for Nonlinear Multivariable Processes Control

LUPU CIPRIAN, POPESCU DUMITRU, UDREA ANDREEA, DIMON CATALIN

Automatics and Computer Science
Polytechnica University of Bucharest
313, Splaiul Independentei, Sector 6
ROMANIA

Abstract: This study presents comparative results of real time implementing of three methods for control a multivariable process with nonlinear static characteristics. The first two methods are classic and are based on decentralized control and respectively, on static decoupling procedures. The third proposed method is developed using a combined feedforward and feedback control scheme with nonlinear compensators. The use of this last structure imposes solving some specific problems, like process’s static characteristic determination, construction of a nonlinear compensator or robust control law design. The applicability of the proposed method and of the two classic methods is proved using a real-time structure based on RST control algorithms. In the end, its software implementation and the obtained results are also shown.

Key-Words: multivariable control, nonlinear process, real time system

1 Introduction

It is a common industrial practice to reduce a multivariable control problem to SISO control approach. There are a lot of valuable methods, strategies and solutions obtained in researches for solving this problem. Few of these are base on decentralized control strategy and respectively, on static decoupling procedures.

In this paper there are presented some comparative results of three methods real time implementation for the control of a multivariable process with nonlinear static characteristic. The first two methods are classical and are based on decentralized control and respectively, on static decoupling procedures. For the third method, the authors propose an improved scheme and the corresponding design procedures, based on a combined feedforward and feedback structure.

Shortly, the decentralized control strategy design the SISO control loops of a MIMO process as totally independent loops, as shown in Fig. 1. Each control algorithm is robust enough to reject the disturbances determined by all other “parallel” loops functioning.

Supplementary, the static decoupling procedures come up with an additional decoupling block, generally introduced between the multivariable process and the N independent control algorithms (Fig. 2). The product between the decoupling block (matrix) and the process’s static gain matrix is an identity matrix.

![Fig. 1. Decentralized control principle](image1)

![Fig. 2. Static decoupling scheme](image2)

The two presented methods provide very good results in normal situations, when the process’s nonlinearities do not have important effects. If the nonlinearities are important, for real time functioning, there can be observed: decreasing performances or limitations (like bad reference tracking and the reduction of control system range).

To solve these problems, starting with decentralized scheme, the authors propose a control scheme that compensates the process’s static nonlinearities. This solution implies that for each control loop two commands should added: “a direct command – feedforward command” generated by...
the inverse model command generator, and the second, generated by a classic and very simple algorithm (PID, RST etc.).

Several papers and researches on this type of structure, also named inverse model, exist. Few of these, with a very fortuity choosing excuse, can be mentioned: [1], [8], [10]. According to them, the paper proposes a very simple and efficient version, presented in Fig. 3.

![Proposed control scheme for multivariable decoupled process, with nonlinearity compensation](image)

**Fig.3.** Proposed control scheme for multivariable decoupled process, with nonlinearity compensation

On Fig. 1, 2, 3 the blocks and variables are:
- **Process** – physical system to be controlled;
- **Command calculus** – computes the control law;
- **Classic Alg.** – control algorithm (PID, RST);
- **y** – output of the process;
- **u** – output of the Command calculus block;
- **u alg.** – output of the classic algorithm;
- **u i.m.** – output of nonlinearity compensator (the inverse static model) block;
- **r** – system’s set point or reference trajectory;
- **p** – disturbances of physical process.

Here, the first command (**u i.m.**), based on the process static characteristics, is dependent on set point value and is designed to generate a corresponding value to drive the process’s output close to imposed set point. The second (classic) algorithm (**u alg.**) generates a command that, correct the difference caused by external disturbances and according to set point, by eventual bias error caused by mismatches between calculated inverse process characteristic and situation from real process.

This solution proposes the treatment of these inverse model mismatches, that “disturb” the first command, as a second command classic algorithm’s (identification) model mismatches. This imposes the design of a classic algorithm with a corresponding robustness reserve. For this reason, designing the second algorithm takes in two steps:

- the design of a classic algorithm based on a model identified in a real functioning point – fortuity selected or, on the middle of process characteristic;
- verification of algorithm’s robustness and its improvement, if necessary, using a (re)designing procedure.

Related to classical control loops, inverse model control need addressing some supplementary specific aspects:
- Determination of static characteristic of the process;
- Construction of inverse model;
- Robust control law design.

The structure presented before can be used to control multivariable processes that support decentralized procedures. These imply decomposition of an MIMO (N inputs and M outputs) process in a max(N,M) (usually N=M) parallel and independent processes. A singular process has a main “canal” from u to y, and possible, a lot of secondary “canals” from u_i to y_j, where i≠j. All secondary “canals”, which represent connections between parallel processes, can be considered as disturbances, nonlinearities, process identification mismatches etc., from the point of view of the main “canal”. Fig. 4 presents the main and secondary “canals” for u_i to y_j process.

![Decentralized or decomposition procedure for an MIMO process](image)

**Fig.4.** Decentralized or decomposition procedure for an MIMO process

Particular, for a process with 2 inputs and 2 outputs the control algorithm is presented in Fig. 3.

In the next sections, we will focus on the most important aspects met while designing the presented structure.

### 2 Inverse Model Design Procedure

As mentioned above, for the proposed inverse model control structure, the supplementary specific aspects...
are: determination of static characteristic of each parallel processes, construction of inverse model and robust control law design. We will present these in the next sections.

2.1 Determination of static characteristic
This operation is based on several experiments of discrete step increasing and decreasing of the command \(u(k)\) of the main “canal” and measuring the corresponding stabilized process output \(y(k)\). The command \(u(k)\) covers all possibilities (0 to 100% in percentage representation). Because the secondary “canals”, which will have all important combinations during experiments, can affect the main “canal”, and because the process is disturbed by noises, usually the static characteristics are not identical. The final static characteristic is obtained by meaning of correspondent position of these experiments. Fig. 5 presents this operation. The graphic between two “mean” points can be obtained using extrapolation procedure.

![Fig.5. Determination of static characteristic of the process. Continuous line represents the final characteristic.](image)

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

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

(1)

This can express a measure of superposing of secondary “canals”, noise that action onto process, process’s nonlinearity etc. and is very important on control algorithm designed robustness. Other possibility is to find the position and the value \(m_g\) of the maximal distance from “mean” characteristic.

2.2 Construction of inverse model
This step deals with the “transposition” operation of the means process’s static characteristic. Figure 6 presents this construction. According to this, \(u(k)\) is dependent to \(r(k)\). This characteristic is stored in a table; thus we can conclude with this, for the inverse model based controller, selecting a new set point \(r(k)\) will impose finding in this table the corresponding command \(u(k)\) that determines a process output \(y(k)\) close to the reference value.

![Fig.6. Construction of inverse model](image)

2.3 Control law design
Control algorithm’s duty is to eliminate the disturbance and differences between inverse model computed command and real process behavior.

![Fig.7. RST control algorithm structure](image)

A large variety of control algorithms can be used here, PID, RST, fuzzy etc., but the goal is to have a very simplified one. For this study we use a RST algorithm. This is designed using pole placement procedure [5]. Fig. 7 presents a RST algorithm. The \(R, S, T\) polynomials are:

\[
R(q^{-1}) = r_0 + r_1 q^{-1} + \ldots + r_m q^{-mr} \\
S(q^{-1}) = s_0 + s_1 q^{-1} + \ldots + s_n q^{-nr} \\
T(q^{-1}) = t_0 + t_1 q^{-1} + \ldots + t_m q^{-mr}
\]

(2)

Algorithm pole placement design procedure is based on identified process’s model.
\[
y(k) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(k)
\]

(3)

where

\[
B(q^{-1}) = b_0 q^{-1} + b_1 q^{-2} + \ldots + b_m q^{-mb}
A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_m q^{-ma}
\]

(4)

The identification is made in a specific process operating point and can use recursive least square algorithm exemplified in next relations developed in [5]:

\[
\dot{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)e_0(k+1), \forall k \in N
\]

(5)

\[
F(k+1) = F(k) - \frac{F(k)\phi(k)\dot{\phi}(k)}{1 + \phi^T(k)F(k)\phi(k)}, \forall k \in N
\]

\[
e_0(k+1) = y(k+1) - \hat{\theta}^T(k)\phi(t), \forall k \in N
\]

with the following initial conditions:

\[
F(0) = \frac{1}{\delta} I = (GI)_I \quad 0 < \delta < 1
\]

(6)

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 is necessary, to calibrate algorithm’s robustness. Next expression and Fig. 8 present “disturbance-output” sensibility function.

\[
S_{vy}(e^{j\omega}) = H_{vy}(e^{j\omega}) = \frac{A(e^{j\omega})S(e^{j\omega})}{A(e^{j\omega})S(e^{j\omega}) + B(e^{j\omega})R(e^{j\omega})}, \forall \omega \in \mathbb{R}
\]

(7)

Finally, if is imposed that all nonlinear characteristics to be (graphically) bounded by the two gains, or gain limit to be great or equal to process static characteristic maximal distance \( \Delta G \geq m_g \), a controller that has sufficient robustness was designed.

3 Analysis of Proposed Structure

In this section we will present a few advantages, disadvantages or limitations and some possible developments of the presented structure.

3.1 Advantages of proposed structure

The main advantage consists in using a classic procedure for designing the control algorithm and determination of the inverse command blocks, comparative to multivariable control design procedures. Well know procedure for identification and control law design are used. As it will be shown in experimental tests, all procedures for the inverse model characteristic identification can be included in a real time software application.

The system is very stable due to the global command that contains a “constant” component generated by an inverse model command block, accordingly to set point value.

A fuzzy logic bloc that can “contain” human experience about some nonlinear processes can
replace the inverse model command generator.

Being not very complex in terms of real time software and hardware implementation, the control law doesn’t need important resources.

### 3.2 Disadvantages or limitations of structure

The main limitation is that this procedure can be applied just for the processes that support decoupling control scheme.

This structure is very difficult to be used for the system that doesn’t have a bijective static characteristic and for systems with different functioning regimes.

Another limitation is that this structure can be used only for stable processes. In the situations where the process is “running”, the global command is likely not to have enough flexibility to control it.

The increased number of experiments for the determination of a correct static characteristic can be another disadvantage.

### 3.3 Possible developing

In situation when the control law becomes very complex, situation cased by difficult determination of process characteristics, the system can be “divided” in two ore more components, becoming a “multiple inverse model system”.

These systems can be easily implemented on PLC structures.

### 4 Experimental Results

We have evaluated the achieved performances of the proposed schemes (Fig. 1, 2, 3) on an experimental software simulator presented in Fig. 10.

![Fig.10. Experimental process simulator](image)

On this installation, the user can control two parameters: level and temperature. According to decentralized principles, the level is controlled by Base quantity input and the temperature respectively, by heating quantity input. Both level and temperature processes have nonlinear characteristics caused by the real physics phenomenon and installation particularities (Fig. 11 medium characteristics).

![Fig.11. Nonlinear characteristics level – left, right - temperature](image)

Between the main “canals” there are (two) secondary influences: “canal”: from level to temperature, and from temperature to level. The influence can be express by static 2x2 gain matrix.

By making some tests:

a) \( u_1=20.0, u_2=0.0 \) we obtain \( K_{11}=2.0, K_{21}=0.245 \)
b) \( u_1=0.0, u_2=20.0 \) we obtain \( K_{22}=1.9, K_{12}=0.5 \).

For these values, the coefficients of the decoupling matrix are:

\[
K_{d11}=0.516655, \quad K_{d21}=-0.06662
\]

\[
K_{d22}=0.543848, \quad K_{d12}=-0.13596
\]

By increasing the command \( u_1 \) from 0.0 to 30.0 and keeping \( u_2=10.0 \) constant the raising time \( T_{s1} \) is 4.5s implying the sampling period \( T_{e1} \)of 0.5 s.

By keeping \( u_1=10.0 \) constant and increasing the command \( u_2 \) from 0.0 to 30.0 we obtain the raising time \( T_{s2} = 1.8s \) that makes the sampling period \( T_{e2}=0.2 \) s.

The models for the level and the temperature processes, obtained by using the WinPIM software are:

\[
M_{level} = \frac{0.60592 + 0.07053q^{-1}}{1 - 0.55799q^{-1} - 0.0358q^{-2}}
\]

\[
M_{temp} = \frac{0.12993 + 0.43005q^{-1} + 0.14257q^{-2}}{1 - 0.75371q^{-1} - 0.15185q^{-2} + 0.03953q^{-3}}
\]

These two models were identified in the lower (40%) region of the static characteristic.

The corresponding controllers, determined by employing the WinREG software are:

- for the level process:

\[
R(q^{-1}) = 0.552689 -0.332786q^{-1} + 0.029291q^{-2}
\]

\[
S(q^{-1}) = 1.0 -1.057657q^{-1} + 0.057657q^{-2}
\]

\[
T(q^{-1}) = 1.478306 -1.893358q^{-1} + 0.664246q^{-2}
\]

- for the temperature process:
\[ R(q^{-1}) = 0.492338 - 0.388487q^{-1} + 0.108353q^{-2} + 0.027731q^{-3} \]
\[ S(q^{-1}) = 1.0 - 0.591022q^{-1} - 0.308964q^{-2} - 0.100014q^{-3} \]
\[ T(q^{-1}) = 1.423386 - 1.823019q^{-1} + 0.639569q^{-2} \]

For the three proposed solutions a set of tests was performed. The references value was changed both in the region where the models were identified (40%), as well as in the superior zone, where the nonlinearities are profound.

In next figures the evolution curves are represented using next color code: yellow – set point; green – filtered set point; blue – process output; red – control structure output (total command); purple – RTS algorithm output; orange – identified model output;

Fig.12. Real time performance for decentralized algorithm solution: level – left, right - temperature

Fig.13. Real time performance for decoupling algorithm solution: level – left, right - temperature

Fig.14. Real time performance for proposed algorithm solution: level – left, right - temperature

In these tests it can be observed that:
- all three solutions are stable;
- the last solution is the only one that can track the reference while it is varied on the entire domain (0-100%);
- the decoupling solution presents a command saturation problem;

- the first solution presents the inconvenient that the two loops influence each other, conducting to a slightly instability.

5 Conclusion

The paper proposes an inverse model structure as a solution for multivariable nonlinear processes. For this structure, for each component, there are presented the design methods. These are based on experimental tests, classic identification and closed loop pole placement method.

An analysis on the advantages and the disadvantages of the proposed structure was made.

The experimental results section presents the evaluated results obtained using a real time software implementation. The tests are made on a software simulator.

During exploitation the inverse model solution does not impose complex operations, it is very easy to use and offers superior performances compared to classical structures.

References: