15

Simplified Deployment of Robust Real-Time Systems Using Multiple Model and Process Characteristic Architecture-Based Process Solutions

Ciprian Lupu
Department of Automatics and Computer Science, University "Politehnica" Bucharest
Romania

1. Introduction

A common industrial practice is to find some specific control structures for the nonlinear processes that reduce, as much as possible, the design techniques to classic control approaches. There are a lot of situations when the designing of robust controller leads to complex hardware and software requirements. In international literature there are some interesting solutions (Kuhnen & Janocha, 2001; Dai et al., 2003; Wang & Su, 2006) for solving implementation reduction.
In following sections there will be presented, in the first part, some elements of classic robust design of RST control algorithm and on the second, two alternative solutions based on multiple model and nonlinear compensators structures.

2. Some elements about classic RST robust control design

The robustness of the systems is reported mainly to model parameters change or the structural model estimation uncertainties (Landau et al., 1997). A simple frequency analysis shows that the critical Nyquist point (i.e. the point (-1, 0) in the complex plane) plays an important role in assessing the robustness of the system. In this plan, we can trace hodograf (Nyquist place) open-loop system, i.e. the frequency response. The distance from the hodograf critical point system (edge module), i.e. radius centered at the critical point and tangent to hodograf is a measure of the intrinsic robustness of the system. The distance is greater, the system is more robust.

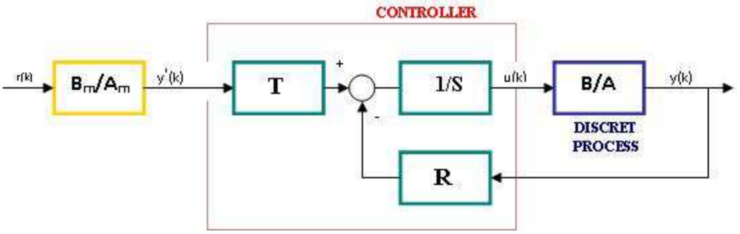


Fig. 1. RST control algorithm structure

For this study we use a RST algorithm. For robustification there are used pole placement procedures (Landau et al., 1997). Fig. 1 presents a RST algorithm.
The R, S, T polynomials are:

$$\begin{aligned} R\left(q^{-1}\right) &= r_0 + r_1 q^{-1} + \ldots + r_{nr} q^{-nr} \\ S\left(q^{-1}\right) &= s_0 + s_1 q^{-1} + \ldots + s_{ns} q^{-ns} \\ T\left(q^{-1}\right) &= t_0 + t_1 q^{-1} + \ldots + t_{nt} q^{-nt} \end{aligned} \tag{1}$$

The RST control algorithm is:

$$S(q^{-1})u(k) + R(q^{-1})y(k) = T(q^{-1})y^*(k) \tag{2}$$

or:

$$u(k) = \frac{1}{s_0}\left[-\sum_{i=1}^{n_S} s_i u(k-i) - \sum_{i=0}^{n_R} r_i y(k-i) + \sum_{i=0}^{n_T} t_i y^*(k-i)\right] \tag{3}$$

where: u(k) - algorithm output, y(k) - process output, y*(k) - trajectory or filtered set point.
When necessary, an imposed trajectory can be generated using a trajectory model generator:

$$y^*(k+1) = \frac{B_m(q^{-1})}{A_m(q^{-1})} r(k) \tag{4}$$

with A_m and B_m like:

$$\begin{aligned} A_m(q^{-1}) &= 1 + a_{m1}q^{-1} + \ldots + a_{mn_{Am}} q^{-n_{Am}} \\ B_m(q^{-1}) &= b_{m0} + b_{m1}q^{-1} + \ldots + b_{mn_{Bm}} q^{-n_{Bm}} \end{aligned} \tag{5}$$

Algorithm pole placement design procedure is based on the identified process' model.

$$y(k) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})} u(k) \tag{6}$$

where

$$\begin{aligned} B\left(q^{-1}\right) &= b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb} \\ A\left(q^{-1}\right) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \end{aligned} \tag{7}$$

The identification (Landau & Karimi, 1997; Lainiotis & Magill, 1969; Foulloy et al., 2004) is made in a specific process operating point and can use recursive least square algorithm exemplified in next relations developed in (Landau et al., 1997):

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\varepsilon^0(k+1), \forall k \in N$$
$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1+\phi^T(k)F(k)\phi(k)}, \forall k \in N \tag{8}$$
$$\varepsilon^0(k+1) = y(k+1) - \hat{\theta}^T(k)\phi(t), \forall k \in N$$

with the following initial conditions:

$$F(0) = \frac{1}{\delta}I = (GI)I, 0 < \delta < 1 \tag{9}$$

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 (Landau et al., 1997). Next expression and Fig. 2 present "disturbance-output" sensibility function.

$$\begin{aligned} S_{vy}(e^{j\omega}) &\overset{def}{=} H_{vy}(e^{j\omega}) = \\ &= \frac{A(e^{j\omega})S(e^{j\omega})}{A(e^{j\omega})S(e^{j\omega}) + B(e^{j\omega})R(e^{j\omega})}, \quad \forall \omega \in R \end{aligned} \tag{10}$$

In the same time, the negative maximum value of the sensibility function represents the module margin.

$$\Delta M\big|_{dB} = -\max_{\omega \in R}\left|S_{vy}(e^{j\omega})\right|\Big|_{dB} \tag{11}$$

Based on this value, in an "input-output" representation (Landau et al., 1997), process nonlinearity can be bounded inside the "conic" sector, presented in Fig. 3, where a_1 and a_2 are calculated using the next expression:

$$\frac{1}{1-\Delta M} \geq a_1 \geq a_2 \geq \frac{1}{1+\Delta M} \tag{12}$$

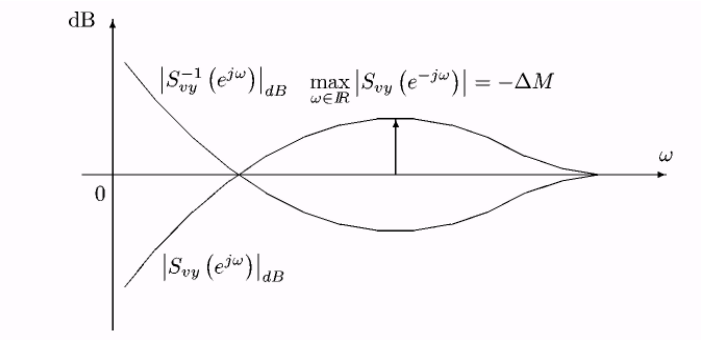


Fig. 2. Sensibility function graphic representation

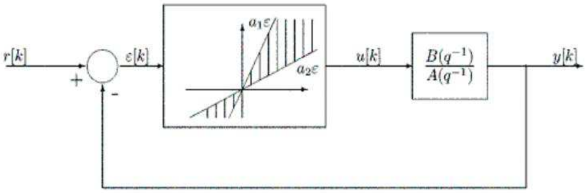


Fig. 3. Robust control design procedure

3. Nonlinear compensator control solution

Various papers and researches target the inverse model control approach; a few of these can be mentioned: (Tao & Kokotovic, 1996; Yuan et al., 2007) etc.

In these researches there have been proposed several types of structures based on the inverse model. According to those results, this section comes up with two very simple and efficient structures presented in Figures 4 and 5. Here, the inverse model is reduced to the geometric inversed process (nonlinear) characteristic – reflection from the first leap of static characteristic of the process, as presented in Figure 6(b).

The first solution (parallel structure) considers the addition of two commands: the first "a feedforward command" generated by the inverse model command generator and the second, generated by a classic, simple algorithm (PID, RST).

The first command, based on the static process characteristic, depends on the set point value and is designed to generate a corresponding value that drives the process' output close to the imposed set point. The second (classic) algorithm generates a command that corrects the difference caused by external disturbances and, according to the set point, by eventual bias errors caused by mismatches between calculated inverse process characteristic and the real process.

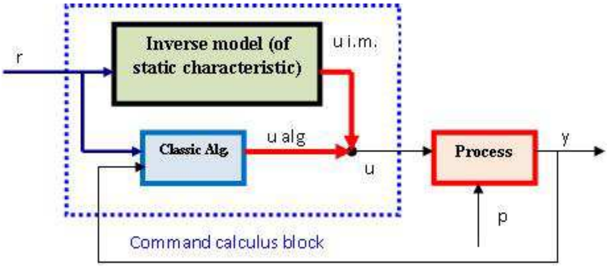


Fig. 4. Proposed scheme for "parallel" structure

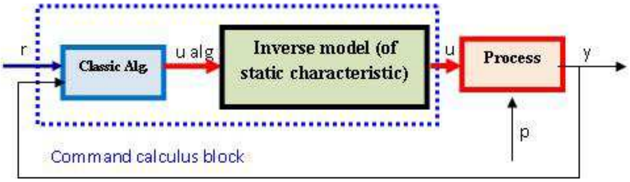


Fig. 5. Proposed scheme for "serial" structure

The second solution (serial structure) has the inverse model command generator between the classic algorithm and the process. The inverse model command generator acts as a nonlinear compensator and depends on the command value. The (classic) algorithm generates a command that, filtered by the nonlinearity compensator, controls the real process.

The presented solutions propose treating the inverse model mismatches that "disturb" the classic command as some algorithm's model mismatches. This approach imposes designing the classic algorithm with a sufficient robustness reserve.

In Figure 4 and 5, the blocks and variables are as follows:

- Process – physical system to be controlled;
- Command calculus – unit that computes the process control law;
- Classic Alg. – control algorithm (PID, RST);
- y – outp5t of the process;
- u – output of the Command calculus block;
- u alg. – output of the classic algorithm;
- u i.m. – output of the inverse model block;
- r – system's set point or reference trajectory;
- p – disturbances.

Related to classical control loops, both solutions need addressing some supplementary specific aspects: determination of static characteristic of the process, construction of inverse model, robust control law design. In next sections we will focus on the most important aspects met on designing of the presented structure.

3.1 Control design procedure

For the first structure the specific aspects of the control design procedure are:

a. determination of the process' (static) characteristic,
b. construction of command generator,
c. robust control law design of classic algorithm.

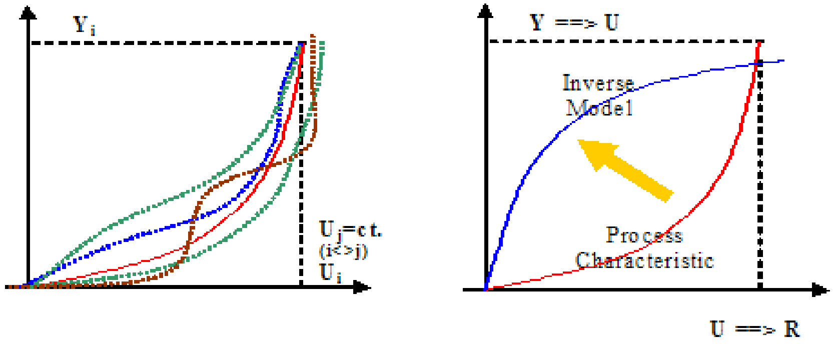
The second structure imposes following these steps:

a. determination of process' characteristic,
b. construction of nonlinearity compensator,
c. designing the classic algorithm based on "composed process" which contains the nonlinearity compensator serialized with real process.

These steps are more or less similar for the two structures. For the (a) and (c) steps it is obvious; for (b) the command generator and nonlinearity compensator have different functions but the same design and functioning procedure. Essential aspects for these steps will be presented.

3.2 Determination of process characteristic

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)* (figure 6 (a)). The command *u(k)* covers all (0 to 100%) possibilities. Because the noise is present, the static characteristics are not identical. The final static characteristic is obtained by meaning of all correspondent positions of these experiments. The graphic between two "mean" points is obtained using extrapolation procedure.



(a) (b)

Fig. 6. (a) - left - Determination of process characteristic. Continuous line represents the final characteristic. (b) - right - Construction of nonlinearity compensator

According to system identification theory, the dispersion of process trajectory can be found using next expression (Ljung & Soderstroom, 1983).

$$\sigma^2[n] \cong \frac{1}{n-1}\sum_{i=1}^{n} y^2[i],\ \forall \mathrm{n} \in N^* \setminus \{1\} \qquad (13)$$

This can express a measure of superposing of noise onto process, process' nonlinearity etc. and it is very important for the control algorithm robust design.

3.3 Construction of nonlinearity compensator (generator)

This step deals with the process's static characteristic „transposition" operation. Figure 6 (b) presents this construction. According to this, *u(k)* is dependent to *r(k)*. This characteristic is stored in a table; thus we can conclude that, for the nonlinearity compensator 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.

3.4 Control law design

The control algorithm's duty is to eliminate the disturbances and differences between the nonlinearity compensator computed command and the real process behavior. A large variety of control algorithms can be used: PID, RST, fuzzy etc., but the goal is to have a very simple one. For this study we use a RST algorithm. This is designed using the pole placement procedure (Landau et al., 1997). Figure 7 presents a RST base algorithm structure. Finally, if it is imposed that all nonlinear characteristics be (graphically) bounded by the two gains, or gain limit to be great or equal to the process static maximal distance characteristic ΔG≥*mg*, a controller that has sufficient robustness was designed.

3.5 Analysis and conclusions for proposed structure

The main advantage consists in using a classic procedure for designing the control algorithm and determining the nonlinearity compensator command block, comparative to robust control design procedures. Well known procedures for identification and law control design are used. All procedures for the inverse characteristic model identification can be included in a real time software application.

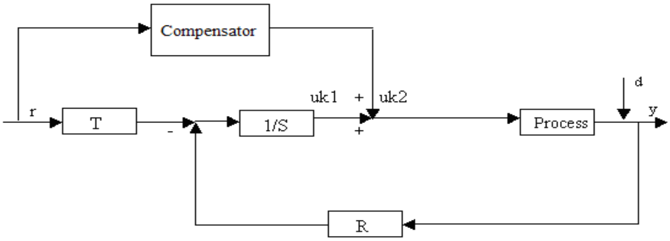


Fig. 7. Parallel RST feedback-feedforward control structure

The system is very stabile due to the global command that contains a “constant” component generated by an inverse static model command block, according to the set point value. This component is not influenced by the noise.

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 law control doesn’t need important resources.

This structure is very difficult to use 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 only be used for stabile processes. In the situations where the process is “running”, the global command is likely to not have enough flexibility to control it.

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

4. Multiple model control solution

The essential function of a real-time control system is to preserve the closed-loop performances in case of non-linearity, structural disturbances or process uncertainties. A valuable way to solve these problems is the multiple-models or multicontroller structure. The first papers that mentioned the “multiple-models” structure/system have been reported in the 90s. Balakrishnan and Narendra are among the first authors addressing problems of stability, robustness, switching and designing this type of structures in their papers Narendra & Balakrishnan, 1997).

Research refinement in this field has brought extensions to the multiple-model control concept. Parametric adaptation procedures – Closed-Loop Output Error (Landau & Karimi, 1997), use of Kalman filter representation (Lainiotis & Magill, 1969), the use of neural networks (Balakrishnan, 1996) or the fuzzy systems are some of the important developments.

Related to classical control loops, multiple-model based systems need addressing some supplementary specific aspects:

- Dimension of multiple-model configuration;
- Selection of the best algorithm;
- Control law switching.

From the multiple-models control systems viewpoint, two application oriented problems can be highlighted:

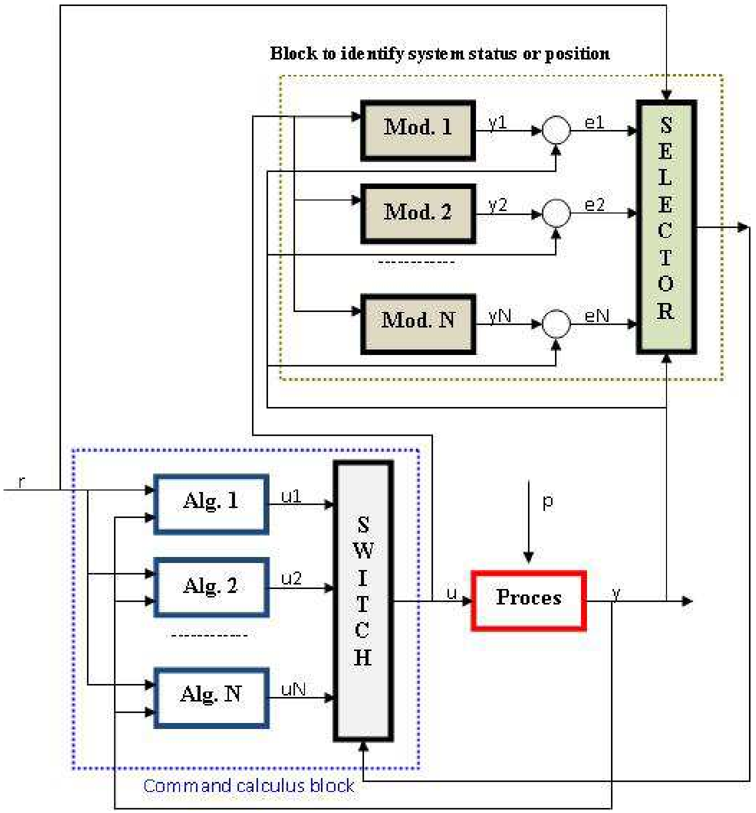


Fig. 8. General scheme for multiple model structure

- Class of systems with nonlinear characteristic, which cannot be controlled by a single algorithm;
- Class of systems with different operating regimes, where different functioning regimes don't allow the use of a unique algorithm or imposes using a very complex one with special problems on implementation.

As function of the process particularity, several multiple-models structures are proposed (Balakrishnan, 1996). One of the most general architectures is presented in Figure 8.

In Fig 8, the blocks and variables are as follows:

- Process – physical system to be controlled;
- Command calculus – unit that computes the process law control ;
- Status or position identification system – component that provide information about the model–control algorithm "best" matching for the current state of the system;
- Mod. 1, Mod. 2, …, Mod. N - previously identified models of different regimes or operating points;
- Alg. 1, Alg. 2, …, Alg. N – control algorithms designed for the N models;
- SWITCH – mix or switch between the control laws;

- SELECTOR – based on adequate criteria evaluations, provides information about the most appropriate model for the system's current state;
- y and y1, y2, …, yN – output of the process and outputs of the N models;
- u – output generate by Command calculus block
- u and u1, u2, uN – output of the Command calculus block and outputs of the N control algorithms, respectively;
- r –set point system or reference trajectory;
- p – disturbances of physical process.

As noted above, depending on the process specifics and the approach used to solve the "control algorithms switching" and/or "the best model choice" problems, the scheme can be adapted on the situation by adding/eliminating some specific blocks. This section focuses on the "switching" problem.

4.1 Control algorithms switching

The logic operation of multiple model system structure implies that after finding the best algorithm for the current operating point of the, the next step consists in switching the control algorithm. Two essential conditions must be verified with respect to this operation:

- To be designed so that no bumps in the applications of the control law are encountered;
- To be (very) fast.

Shocks determined by the switching operation cause non-efficient and/or dangerous behaviors. Moreover, a switch determines a slow moving area of action of the control algorithm, which involves at least performance degradation.

These are the main problems to be solved in designing block switching algorithms. From structurally point of view, this block may contain all implementation algorithms or at least the algorithm coefficients.

4.1.1 Classic solutions

Present solutions (Landau et al., 1997; Dumitrache, 2005) solve more or less this problem and they are based on maintaining in active state all the control algorithms, also called "warm state". This supposes that every algorithm receives information about the process output y(k) and set the point value (eventually filtered) r(k), but only the control law $u_i(k)$ is applied on the real process, the one chosen by the switching block. This solution does not impose supplementary logic function for the system architecture and, for this reason, the switching time between algorithms is short. The drawback of this approach is that when designing the multi-model structure several supplementary steps are necessary.

These supplementary conditions demand the match of the control algorithm outputs in the neighborhood switching zones. The superposition of models identification zones accomplishes this aspect. That can be seen in Fig. 9. As a result of this superposition, the multi-model structure will have an increased number of models.

Other approaches (Dussud et al., 2000; Pages et al., 2002) propose the mix of two or more algorithms outputs. The "weighting" of each control law depends on the distance from the current process operating point and the action zone of each algorithm. Based on this, the switching from an algorithm to another one is done using weighting functions with a continuous evolution in [0–1] intervals. This technique can be easily implemented using fuzzy approach, An example is presented in Fig. 10. This solution involves solving control gain problems, determined by mixing algorithm outputs.

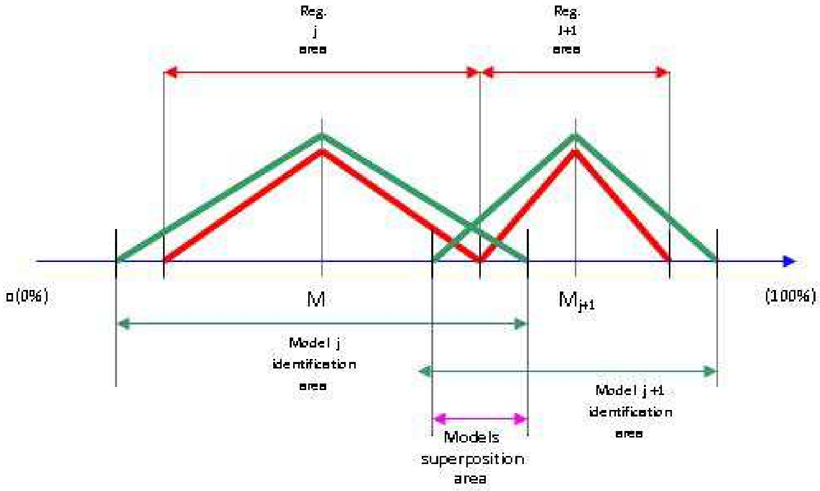


Fig. 9. Superposition of identification zones for two neighbor-models and their corresponding control actions

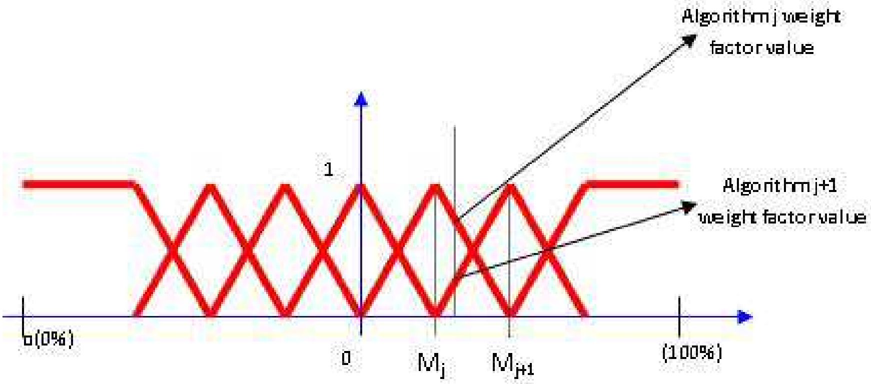


Fig. 10. Algorithms weighting functions for a specified operating position

4.1.2 Proposed solution

In this subsection, there is presented a solution that provides very good results for fast processes with nonlinear characteristics. The main idea is that, during the current functioning of multiple-models control systems with N model-algorithm pairs, it is supposed that just one single algorithm is to be maintained active, the good one, and all the other N-1 algorithms rest inactive. The active and inactive states represent automatic, respectively manual, regimes of a law control. The output value of the active algorithm

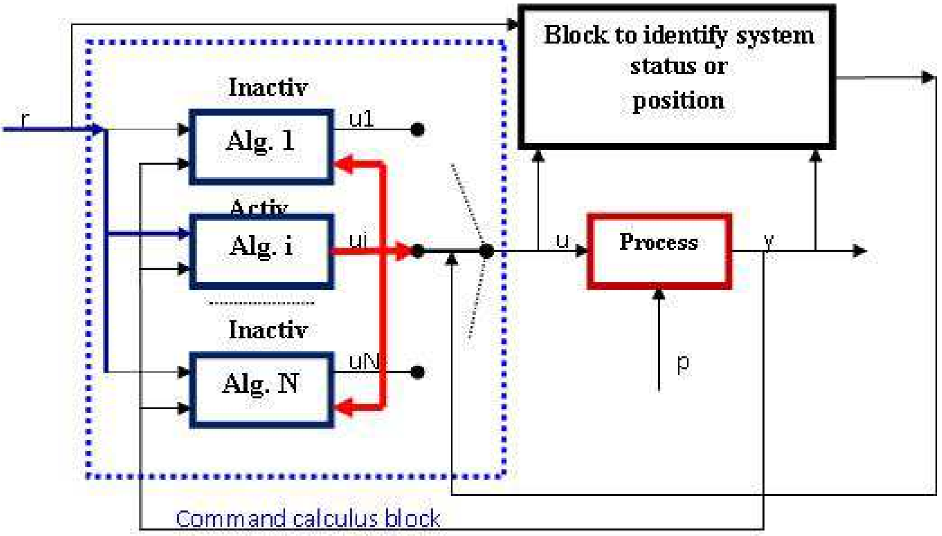


Fig. 11. Proposed multiple model switching solution

corresponds to the manual control for all the other N-1 inactive algorithms., as presented in Fig. 11. In the switching situation, when a “better” *Aj* algorithm is found, the actual *Ai* active algorithm is commuted in an inactive state, and *Aj* in active state, respectively. For a bumpless commutation, the manual–automatic transfer problems must be solved, and the performance solution to this is proposed in the next section.

The system can be implemented in two variants – first - with all inactive algorithms holding on manual regime, or – second - just a single operating algorithm (the active one) and activation of the “new” one after the computation of the currently corresponding manual regime and switching on automatic regime. Both variants have advantages and disadvantages. Choosing one of them requires knowledge about the hardware performances of the structure. After a general view, the first variant seems to be more reasonable.

In all cases, it is considered that the active algorithm output values represent manual commands for the “new” selected one.

4.2 Manual – automatic bumpless transfer

The „key“ of proposed multiple model switching solution performances is based on manual-to-automatc bumpless transfer, so in this section some important elements about are presented.

The practice implementation highlights important problems like manual-to-automatic (M→A)/automatic-to-manual (A→M) regime commutations, respectively turning out/in from the control saturation states; (i.e. manual operation is the situation where the command is calculated and applied by human operator). Of course, these problems exist in analogical systems and have specific counteracting procedures, which are not applicable on numerical systems.

In real functioning, M→A transfer is preceded by “driving” the process in the nominal action zone. To avoid command switching “bumps”, one must respect the following two conditions:

- Process output must be perfectly matched with the set point value;
- According to the algorithm complexity (function of the degrees of controller polynomials), the complete algorithm memory actualization must be waited for.

Neglecting these conditions leads to “bumps” in the transfer because the control algorithm output value is computed using the actual, but also the past, values of the command, process and set point, respectively.

At the same time, there are situations when the perfect “matching” between process output and set point value is very difficult to obtained and/or needs a very long time. Hence, the application of this procedure becomes impossible in the presence of important disturbances.

In the following, these facts will be illustrated using an RST control algorithm (Foulloy et al., 2004), Fig. 1.

In this context, for a inactive algorithm – possible candidate for next active one, since the algorithm output is the manual command set by operator (or active algorithm) and the process output depends on command, the set point remains the only “free” variable in the control algorithm computation. Therefore, the proposed solution consists in the modification of the set point value, according to the existent control algorithm, manual command and process output (Lupu et al., 2006).

Memory updating control algorithm is done similarly as in the automatic regime. For practical implementatio a supplementary memory location for the set point value is necessary. From Eq(3), results the expression for the set point value:

$$y^*(k)=\frac{1}{t_0}[\sum_{i=0}^{n_S} s_i u(k-i)+\sum_{i=0}^{n_R} r_i y(k-i)-\sum_{i=1}^{n_T} t_i y^*(k-i)] \qquad (14)$$

When the set point (trajectory) generator Eq(4) exists, keeping all the data in correct chronology must be with respect to the following relation:

$$r(k)=\frac{A_m(q^{-1})}{B_m(q^{-1})}y^*(k) \qquad (15)$$

System operation scheme is presented in Fig. 13.

Concluding, this solution proposes the computation of that set point value that determines, according to the algorithm history and process output, a control equal to the manual command applied by the operator (or active algorithm). At the instant time of the M→A switching, there are no gaps in the control algorithm memory that could determine bumps. An eventually mismatching between the set point and process output is considered as a simple change of the set point value. Moreover, this solution can be successfully used in cases of command limitation.

The only inconvenient of this solution is represented by the necessary big computation power when approaching high order systems, which is not, however, a problem nowadays.

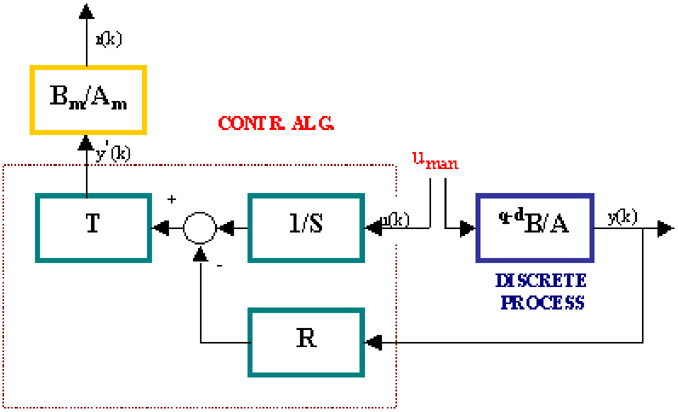


Fig. 13. Computation of the set point value for imposed manual command

5. Experimental results

We have evaluated the achieved performances of the multi-model control structure and nonlinear compensator control using a hardware and software experimental platform, developed on National Instruments LabWindows/CVI. In figure 14, one can see a positioning control system. The main goal is the vertical control of the ball position, placed inside the pipe; here, the actuator is air supply unit connected to cDAQ family data acqusition module.

The obtained results are comparead to very complex (degree = 8) RST robust algorithm. Total operations number for robust structure is 24 multiplies and 24 adding or subtraction.

The nonlinear relation between the position Y (%) and actuator command U (%) is presented in Figure 15. One considers three operating points P_1, P_2, and P3 on the plant's nonlinear diagram (Figure 15). Three different models are identified like: M_1 (0-21%), M_2 (21-52%) and M_3 (52-100%). These will be the zones for corresponding algorithms.

According to the models-algorithms matching zones (Lupu et al., 2008), we have identified the models M_1, M_2 and M_3, as being appropriated to the following intervals (0-25%), (15-55%) (48-100%), respectively. For a sampling period T_e=0.2 sec, the least-squares identification method from Adaptech/WinPIM platform (Landau et al., 1997) identifies the next models:

$$M_1 = \frac{0.35620 - 0.05973q^{-1}}{1 - 0.454010q^{-1} - 0.09607q^{-2}}$$

$$M_2 = \frac{1.23779 - 0.33982q^{-1}}{1 - 0.98066q^{-1} - 0.17887q^{-2}}$$

$$M_3 = \frac{2.309530 - 0.089590q^{-1}}{1 - 0.827430q^{-1} - 0.006590q^{-2}}$$

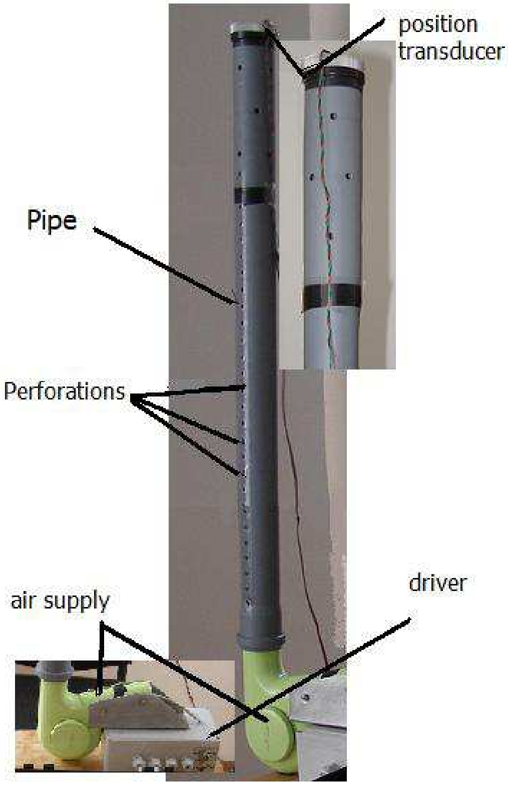


Fig. 14. Process experimental platform

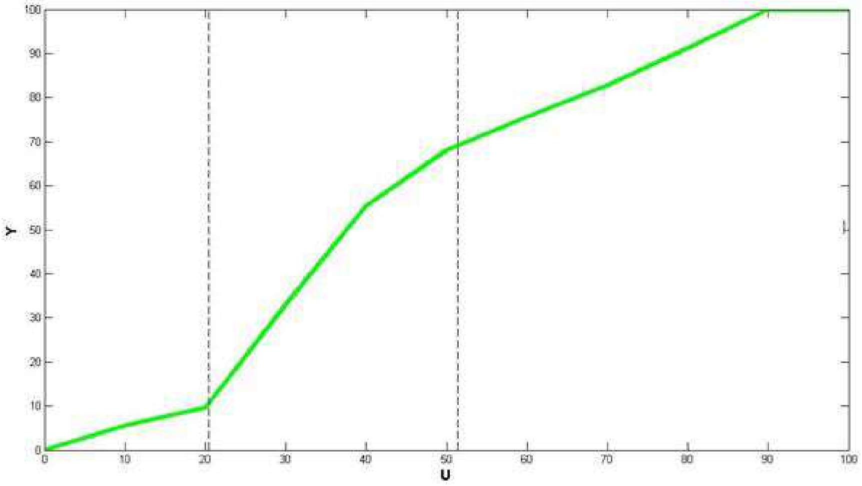


Fig. 15. Nonlinear diagram of the process

In this case, we have computed three corresponding RST algorithms using a pole placement procedure from Adaptech/WinREG platform (Landau et al., 1997). The same nominal performances are imposed to all systems, through a second order system, defined by the dynamics $\omega_0 = 3.0$, $\xi = 2.5$ (tracking performances) and $\omega_0 = 7.5$, $\xi = 0.8$ (disturbance rejection performances) respectively, keeping the same sampling period as for identification.
All of these algorithms control the process in only their corresponding zones.

$$R_1(q^{-1}) = 1.670380\ \ \text{-}0.407140q^{-1}\ \ \text{-}0.208017q^{-2}$$

$$S_1(q^{-1}) = 1.000000\ \ \text{-}1.129331q^{-1}\ +\ 0.129331q^{-2}$$

$$T_1(q^{-1}) = 3.373023\ \ \text{-}3.333734\mathrm{q}^{-1}\ +\ 1.015934\mathrm{q}^{-2}$$

$$R_2(q^{-1}) = 0.434167\ \ \ 0.153665q^{-1}\ \ \text{-}0.239444q^{-2}$$

$$S_2(q^{-1}) = 1.000000\ \ \text{-}0.545100q^{-1}\ \ \text{-}0.454900q^{-2}$$

$$T_2(q^{-1}) = 1.113623\ \ \text{-}1.100651\mathrm{q}^{-1}\ +\ 0.335417\mathrm{q}^{-2}$$

$$R_3(q^{-1}) = 0.231527\ \ \text{-}0.160386q^{-1}\ \text{-}8.790\text{E-}04q^{-2}$$

$$S_3(q^{-1}) = 1.000000\ \ \text{-}0.988050q^{-1}\ \ \text{-}0.011950q^{-2}$$

$$T_3(q^{-1}) = 0.416820\ \ \text{-}0.533847\mathrm{q}^{-1}\ +\ 0.187289\mathrm{q}^{-2}$$

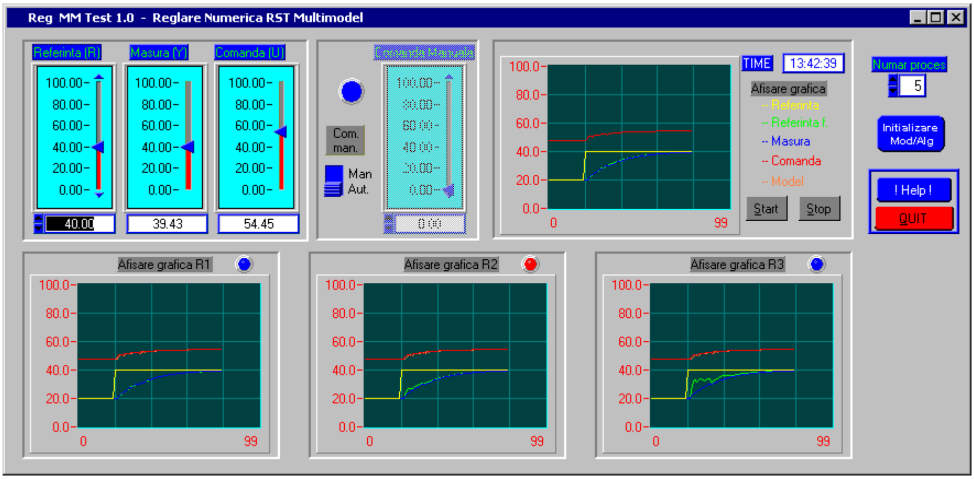


Fig. 16. Multi-model controller real-time software application

To verify the proposed switching algorithm, a multi-model controller real-time software application was designed and implemented, that can be connected to the process. The user interface is presented on Figure 16.

On the top of Figure 16, there are respectively the set point, the output and control values, manual-automatic general switch, general manual command and graphical system evolution display. On the bottom of Figure 16, one can see three graphical evolution displays corresponding to the three controllers (R_i, S_i, T_i, i=1...3). The colors are as follows: yellow – set point value, red – command value, blue – process output value and green – filtered set point value.

Using this application, few tests were done to verify the switching between two algorithms. The switching procedure is determinate by the change of the set point value. These tests are:

a. from 20% (where algorithm 1 is active) to 40% (where algorithm 2 is active). The effective switching operation is done when the filtered set point (and process output) becomes greater than 21%. Figure 17(a) presents the evolutions.

b. from 38% (where algorithm 2 is active) to 58% (where algorithm 3 is active). The effective switching operation is done when the filtered set point (and process output) becomes greater than 52%. Figure 17(b) presents the evolutions.

In both tests, one can see that there are no shocks or that there are very small oscillations in the control evolution by applying this approach. Increasing the number of models-algorithms to 4 or 5 could eliminate the small oscillations.

To verify the nonlinear compensator control structure, a second real-time software application was designed and implemented, that can be connected with the process. The

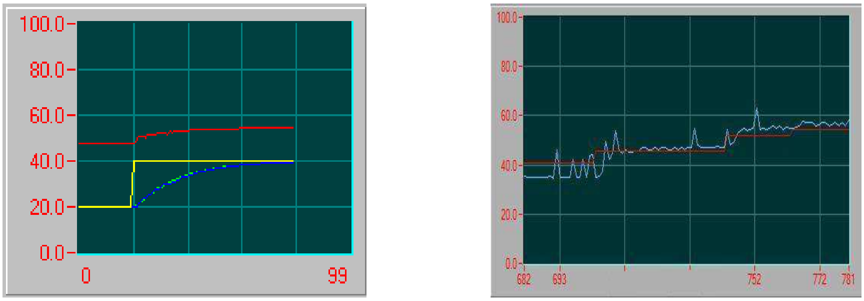
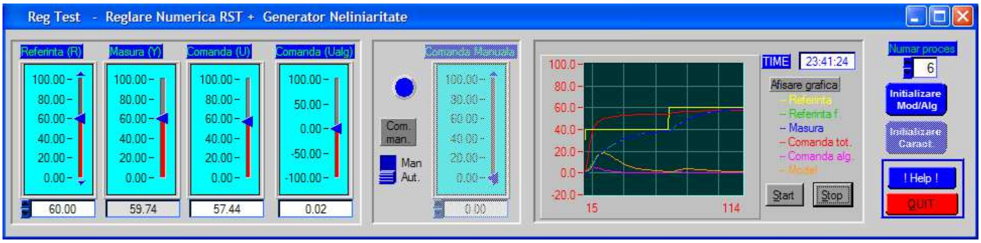


Fig. 17. a) (left) switching test; b) (right) switching test



Referinta (R) 60.00

Masura (Y) 59.74

Comanda (U) 57.44

Comanda (Ualg) 0.02

Comanda Manuala 0.00

TIME 23:41:24

Numar proces 6

Fig. 18. Nonlinear compensator controller real-time software application

user interface is presented on Figure 18. This application implement the scheme proposed in Fig 7 and allows the user in a special window, to construct the nonlinear compensator.
Using this application, that contains a simple second order RST algorithm, few tests were effectuated to verify the structure. These tests are:

a. Determination of inverse model characteristic. Figure 19(a) presents this evolutions and contains the corresponding *r(k)-u(k)* data pairs obtained by dividing the total domain (0-100%) in 10 subinterval (0-10, 10-20 etc).
b. Testing structure stability on different functioning point. Figure 19(b) presents these evolutions.

On (a) test one can see the nonlinear process model characteristics identification procedures. The second one, present that there are no shocks and the system is stable on different functioning points.
For proposed control structure, presented in Figure 7 was identified a wery simple model:

$$M = \frac{1.541650}{1-0.790910q^{-1}}$$

In this case, we have computed the corresponding RST algorithms using a pole placement procedure from Adaptech/WinREG platform. The nominal performances are imposed, through a second order system, defined by the dynamics ω_0 = 2, ξ = 0.95 (tracking performances) and ω_0 = 1.1, ξ = 0.8 (disturbance rejection performances) respectively, keeping the same sampling period as for identification.

$$R(q^{-1}) = 0.083200 \ \text{-}0.056842q^{-1}$$

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

$$T(q^{-1}) = 0.648656 \ \text{-}1.078484\text{q}^{-1} \ + \ 0.456187\text{q}^{-2}$$

To calculate corresponding command for a single controller presented before, there are used 7 multiplies and 7 adding or subtraction operations.
For the second control structure, in addition to command calculus operation here is the calculus of direct command. This depends on software implementation. For PLC, particular

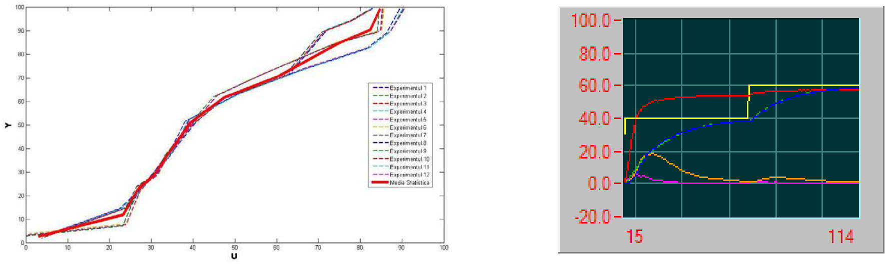


Fig. 19. a) Process static determination test; b) functioning test;

and real time process computer, in general, where (C) code programming can be used, in a solution or other similar implementation:

```
// segment determination
segment = (int)(floor(rdk/10));
// segment gain and difference determination
panta = (tab_cp[segment+1] - tab_cp[segment]) * 0.1;
// linear value calculus
val_com_tr = uk + 1.00 * (panta * (rdk - segment*10.0) + tab_cp[segment]);
```

there are necessary 10 multiplies and 4 adding or subtraction operations (the time and memory addressing effort operation is considered equal to a multiply operation). Total operations number for nonlinear compensator structure is 17 multiplies and 14 adding or subtraction.

Because the multi-models control structure must assure no bump commutations, all of 3 control algorithms work in parallel (Lupu et al., 2008). So, for multiple model structure, to calculate corresponding command for a C1 controller 9 multiplies and 9 adding or subtraction operations are used, for C2 9 multiplies and 9 adding or subtraction operations and for C3 9 multiplies and 9 adding or subtraction operations, total number 27 multiplies and 27 adding or subtractions.

As mentioned before, total operations number for classic robust structure is 24 multiplies and 24 adding or subtraction.

It is visible that nonlinear compensator structure has a less number of multiplies and adding or subtraction comparative to classic multi-model solutions and robust control approach.

In the same time multi-model and robust control solutions have comparative numbers of implemented operations. The choice of solution depends on process features and used hardware.

This means that the system with nonlinear compensator is faster or needs a more simplified hardware and software arquitecture.

6. Conclusions

The first proposed method (multiple models) is a more elaborated one and needs a lot of precise operations like data acquisition, models identification, and control algorithms design. For these reasons it allows us to control a large class of nonlinear processes that can contain nonlinear characteristics, different functioning regimes etc.

The second proposed method (inverse model) does not impose complex operations, it is very easy to use, but it is limited from the nonlinearity class point of view. This structure is very difficult to use for the system that doesn't have a bijective static characteristic or have different functioning regimes.

7. Acknowledgment

This work was supported by CNCSIS - IDEI Research Program of Romanian Research, Development and Integration National Plan II, Grant no. 1044/2007 and „Automatics, Process Control and Computers“ Research Center from University „Politehnica“ of Bucharest. (U.P.B. – A.C.P.C.) projects.

8. References

Balakrishnan, J., (1996). *Control System Design Using Multiple Models, Switching and Tuning*, Ph. D. Dissertation, University of Yale, USA

Dai, X., He, D., Zhang, T. & Zhang, K., (2003). Generalized inversion for the linearization and decoupling control of nonlinear systems, *Proc. Of IEE Control Theory Application*, pp. 267-277

Dumitrache, I., (2005). *Engineering the automatic* (*Ingneria Reglarii Automate)*, Politehnica Press, Bucuresti, 2005, ISBN 973-8449-72-3

Dussud, M., Galichet S. & Foulloy L.P., (2000). Application of fuzzy logic control for continuous casting mold level control, *IEEE Trans. on Control Systems Technology*, 6. no. 2, pp. 246-256, ISSN 1063–6536

Foulloy, L., Popescu, D. & Tanguy, G.D., (2004). *Modelisation, Identification et Commande des Systemes*, Editura Academiei Romane, Bucuresti, ISBN 973-27-1086-1

Kuhnen, K. & Janocha, H. (2001). Inverse feedforward controller for complex hysteretic nonlinearities in smart-material systems*, Proc. of the 20th IASTED-Conf. on Modeling, Identification and Control, Insbruck*, pp. 375-380, ISBN 0171-8096

Lainiotis, D. G. & Magill, D.T, (1969). Recursive algorithm for the calculation of the adaptive Kalman filter weighting coefficients, *IEEE Transactions on Automatic Control*, 14(2):215–218, April, *ISSN* 0018-9286

Landau, I. D. & Karimi, A., (1997). Recursive algorithm for identification in closed loop: a unified approach and evaluation, *Automatica*, vol. 33, no. 8, pp. 1499-1523, ISSN 0005-1098

Landau, I. D., Lozano, R. & Saad, M. M', (1997). *Adaptive Control*, Springer Verlag, London, ISBN 3-540-76187-X

Ljung, L. & Soderstroom, T., (1983). *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, Massashusetts, ISBN 10 - 0-262-12095-X.

Lupu C., Popescu D., Ciubotaru, Petrescu C. & Florea G., (2006). Switching Solution for Multiple Models Control Systems, *Proc. of MED'06*, The *14th Mediterranean Conference on Control Automation*, 28-30 June, 2006, Ancona, Italy, pp. 1-6, ISBN: 0-9786720-0-3

Lupu, C., Popescu, D., Petrescu, C., Ticlea, Al., Dimon, C., Udrea, A. & Irimia, B., (2008). Multiple-Model Design and Switching Solution for Nonlinear Processes Control, *Proc. of ISC'08, The 6th Annual Industrial Simulation Conference*, 09-11 June, Lyon, France, pp. 71-76, ISBN 978-90-77381-4-03

Narendra, K. S. & Balakrishnan, J., (1997). Adaptive Control using multiple models, *IEEE Transactions on Automatic Control*, vol. 42, no. 2, pp. 171 – 187, *ISSN* 0018-9286

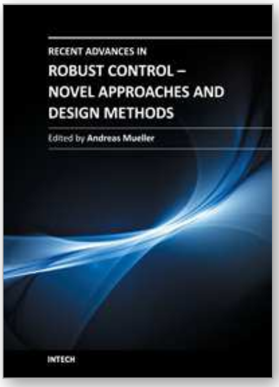
Pages, O., Mouille P., Odonez, R. & Caron, B., (2002). Control system design by using a multi-controller approach with a real-time experimentation for a robot wrist, *International Journal of Control*, vol. 75 (no. 16 & 17), pp. 1321 – 1334, ISSN: 1366-5820

Tao, G. & Kokotovic, P., (1996). *Adaptive control of systems with actuator and sensor nonlinearities*, Wiley, N.Y., ISBN 0-471-15654-X

Yuan, X., Wang, Y. & Wu, L., (2007). Adaptive inverse control of excitation system with actuator uncertainty, *Wseas Transactions on System and Control*, Issue 8, Vol.2, August 2007, pp. 419-428, ISSN 1991-8763

Wang, Q. & Su, C., (2006). Robust adaptive control of a class of nonlinear systems including actuator hysteresis with Prandtl–Ishlinskii presentations, *Automatica*, vol. 42, 2006, pp. 859 – 867, ISSN 0005-1098

Wang, Q. & Stengel, R., F., (2001). Searching for robust minimal-order compensators, *ASME Journal Dynamic Systems, Measurement and, Control*, vol. 123, no. 2, pp. 233–236, 2001, ISSN 0022-0434



Recent Advances in Robust Control - Novel Approaches and Design Methods
Edited by Dr. Andreas Mueller

ISBN 978-953-307-339-2
Hard cover, 462 pages
Publisher InTech
Published online 07, November, 2011
Published in print edition November, 2011

Robust control has been a topic of active research in the last three decades culminating in H_2/H_\infty and \mu design methods followed by research on parametric robustness, initially motivated by Kharitonov's theorem, the extension to non-linear time delay systems, and other more recent methods. The two volumes of Recent Advances in Robust Control give a selective overview of recent theoretical developments and present selected application examples. The volumes comprise 39 contributions covering various theoretical aspects as well as different application areas. The first volume covers selected problems in the theory of robust control and its application to robotic and electromechanical systems. The second volume is dedicated to special topics in robust control and problem specific solutions. Recent Advances in Robust Control will be a valuable reference for those interested in the recent theoretical advances and for researchers working in the broad field of robotics and mechatronics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ciprian Lupu (2011). Simplified Deployment of Robust Real-Time Systems Using Multiple Model and Process Characteristic Architecture-Based Process Solutions, Recent Advances in Robust Control - Novel Approaches and Design Methods, Dr. Andreas Mueller (Ed.), ISBN: 978-953-307-339-2, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-robust-control-novel-approaches-and-design-methods/simplified-deployment-of-robust-real-time-systems-using-multiple-model-and-process-characteristic-ar

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821