International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:1, No:7, 2007

A Supervisory Scheme for Step-Wise Safe Switching Controllers

Fotis N. Koumboulis, and Maria P. Tzamtzi

***Abstract*—**A supervisory scheme is proposed that implements Stepwise Safe Switching Logic. The functionality of the supervisory scheme is organized in the following eight functional units: Step-Wise Safe Switching unit, Common controllers design unit, Experimentation unit, Simulation unit, Identification unit, Trajectory cruise unit, Operating points unit and Expert system unit. The supervisory scheme orchestrates both the off-line preparative actions, as well as the on-line actions that implement the Stepwise Safe Switching Logic. The proposed scheme is a generic tool, that may be easily applied for a variety of industrial control processes and may be implemented as an automation software system, with the use of a high level programming environment, like Matlab.

***Keywords*—**Supervisory systems, Safe switching, Nonlinear systems.

I. Introduction

INDUSTRIAL processes are characterized by complex behavior, nonlinearities, lack of analytical models and parameter uncertainty, facts that impose significant difficulties to the development of satisfactory control design approaches and consequently the development of efficient automation systems. To deal with these problems, a new industrial control design approach, based on switching logic, has been introduced in [1]. The proposed hybrid controllers benefit with respect to the ease of design and implementation and their possibility to cover a large operating range of the industrial process.

A supervisory switching control scheme for nonlinear processes (see for example [1]-[4]) consists, in general, of a set of field controllers, each designed to achieve specific performance requirements for a limited range of operation of the nonlinear process, as well as a supervisory controller that implements the switching logic, that is it performs switching between the field controllers, as the process input/output trajectories move between different areas of operation.

A switching approach, called Step-Wise Safe Switching (SWSS), has been first established in [1] for the case of single input-single output (SISO) systems, with unknown description. Assuming that the SISO nonlinear process can be satisfactorily approximated by a set of switching SISO linear systems, derived through identification around a sufficiently dense set of nominal operating points, the SWSS technique is based on the application of controllers that commonly achieve the required performance for two or more adjacent nominal operating points. Moreover, switching between the controllers is allowed to take place only when the process has reached an operating point. This requirement is strict but it can avoid undesirable effects that may come from switching while moving e.g. instability. Thus, the motion between any two different operating points is performed by moving in a step-wise manner between operating areas of an appropriately selected sequence of adjacent nominal operating points. The operating areas are determined with the use of experimental process data.

The SWSS algorithm requires several preparative actions, as for example the selection of the nominal operating points, the determination of the areas of validity for each operating point, the derivation of the corresponding set of linearized systems through identification, and finally the design of "common" controllers, that achieve the required performance simultaneously for two or more adjacent nominal operating points.

The derivation of common controllers is a difficult task that belongs to the field of robust control. In the field of robust control, a variety of control design problems have been solved, as for example stabilizability, model matching, disturbance rejection, input-output decoupling and pole placement (e.g. [5]-[12] and the references therein). For some of these problems the set of controllers have been determined analytically and explicitly but many related problems remain to be solved.

An alternative approach for the design of common controllers, that achieve specific performance requirements simultaneously for two or more models, is based on the application of heuristic approaches (see for example [14] and the references therein). A heuristic algorithm for the design of common PI controllers for multilinear plants, has been also developed and introduced in [15]. The proposed algorithm is generic, in the sense that it does not depend on the degree or the specific structure of the process model, or even the design goal under consideration. Moreover, the algorithm is very simple to use and it can be extended for the derivation of

Manuscript received September 30, 2007. The present work is co-financed by the Hellenic Ministry of Education and Religious Affairs' and the ESF of the European Union within the framework of the "Operational Programme for Education and Initial Vocational Training" (Operation "Archimedes-I")

F.N. Koumboulis is with the Department of Automation, Halkis Institute of Technology, 34400, Psahna, Evia, Greece. (corresponding author, phone:+22280-99674; fax: +22280-99527; e-mail: koumboulis@teihal.gr).

M.P. Tzamtzi is with the Department of Automation, Halkis Institute of Technology, 34400, Psahna, Evia, Greece. (e-mail: tzamtzi@teihal.gr).

common controllers of several types, e.g. PID controllers, or other general order dynamic controllers. However, the increase of the number of controller parameters increases the numerical complexity of the algorithm.

In the present paper we present a supervisory scheme which is used for the implementation of SWSS. The functionality of the supervisory scheme is organized in the following eight functional units: Step-Wise Safe Switching unit, Common controllers design unit, Experimentation unit, Simulation unit, Identification unit, Trajectory cruise unit, Operating points unit and Expert system unit. The supervisory scheme orchestrates both the off-line preparative actions, as well as the on-line actions that implement the SWSS Logic. The proposed scheme may be implemented as an automation software system, with the use of a high level programming environment, like Matlab. It is also important to note that the supervisory scheme presented in the following sections is a generic tool, that may be easily applied for a variety of industrial control processes.

In Section 2, we present in short the SWSS algorithm, introduced in [1]. In Section 3, we present in detail the proposed supervisory scheme, its functionality, its organization in independent software units, as well as the interaction between these units.

II. Step-Wise Safe Switching Logic

Consider a single input-single output process, with discrete-time nonlinear description of the form

$$y(k) = f\big(y(k-1),\ldots,y(k-n),u(k-d),\cdots,u(k-d-m)\big) \quad (1)$$

where $n,m \in \mathbb{N}$, y denotes the output of the process, u denotes the input of the process, d denotes the process delay and f is at least once differentiable. Let $L = \{\ell_1,\ell_2,\ldots,\ell_\mu\}$ denote a set of nominal operating points of the process, where each $\ell_i, i = 1,\ldots,\mu$ is denoted as $\ell_i = [Y_i,U_i]$, with Y_i and U_i denoting the corresponding nominal output and input values, that satisfy the condition

$$Y_i = f\big(Y_i,\ldots,Y_i,U_i,\cdots,U_i\big) \quad (2)$$

Consider also that the process description is approximated by the following set of linear models, which are determined through identification around the nominal operating points:

$$S_i: \; A_i(q^{-1})\Delta_i y(k) = B_i(q^{-1})\Delta_i u(k-d_i) + C_i(q^{-1})\varepsilon_i(k)$$

where $\Delta_i y(k) = y(k) - Y_i$ and $\Delta_i u(k) = u(k) - U_i$ denote perturbations of the output and input variables about the *i*-th nominal operating point ℓ_i and $\varepsilon_i(k)$ denotes the unmodeled error or disturbance in S_i. The nonnegative integer d_i denotes the delay of the model S_i. The operators $A_i(q^{-1})$, $B_i(q^{-1})$ and $C_i(q^{-1})$ are polynomials of the delay operator q^{-1} with real coefficients.

For each nominal operating point ℓ_i we determine a pair of operating areas, named *target* (O_i) and *tolerance* ($\bar{O}_i$) operating areas [1], which constitute experimental approximations of the neighborhood of validity of each local linear model S_i. The target operating area is determined as a rectangle in the (U,Y)-space, according to the following rule [1]: For each step transition between an initial operating point $\rho_s = [Y_s,U_s]$ within O_i to a final operating point $\rho_f = [Y_f,U_f]$ also within O_i, the percentage of deviation between the responses of the nonlinear system (1) and the corresponding linearized system S_i remain smaller than a threshold value $\varepsilon_{\text{target}}$. The tolerance operating area is determined in a similar way using a threshold value $\varepsilon_{\text{tol}} > \varepsilon_{\text{target}}$. Note that the target and the tolerance operating areas are determined using exclusively experimental data.

The nominal operating points are selected dense enough to satisfy the following requirements [1]:

a) The target operating areas of adjacent nominal operating points are overlapping, i.e.

$$O_i \cap O_{i+1} \neq \varnothing, \;\; i = 1,\ldots,\mu-1 \quad (3)$$

b) The union of the target operating areas of ℓ_i and $\ell_{i\pm1}$ is a subset of the respective tolerance operating area of ℓ_i, i.e.

$$\begin{gathered} O_1 \cup O_2 \subset \bar{O}_1, \;\; O_{\mu-1} \cup O_\mu \subset \bar{O}_\mu, \\ O_{i-1} \cup O_i \subset \bar{O}_i,\; O_i \cup O_{i+1} \subset \bar{O}_i, i = 2,\ldots,\mu-1 \end{gathered} \quad (4)$$

The above conditions constitute an experimental formulation of the *dense web principle* [1], according to which the set of linear models S_i, $i = 1,\ldots,\mu$ describes satisfactorily the process behavior.

Finally, consider that for each pair (ℓ_i,ℓ_{i+1}) of adjacent operating points, there exists a common controller $C_{i,i+1}$ that satisfies a set of desired design requirements simultaneously for both linear models S_i and S_{i+1}. Then, the SWSS algorithm is summarized in the following steps [1]:

Step-Wise Safe Switching Algorithm

Step 1: Apply a safe controller until the output variable is relaxed (i.e. reaches a steady state).
Step 2: Set $\lambda = 1$.
Step 3: Read the present operating point, let ρ_λ.
Step 4: Read the desired operating point $\rho_{\lambda+1}$.
Step 5: Choose a pair of adjacent target operating areas

$(O_{\eta\lambda}, O_{\eta\lambda+1})$, such that ρ_λ lies within $O_{\eta\lambda}$ and $\rho_{\lambda+1}$ lies within $O_{\eta\lambda+1}$, else return to Step 1.

Step 6: Switch to controller $C_{\eta_\lambda,\eta_{\lambda+1}}$ and force the closed loop system from ρ_λ to $\rho_{\lambda+1}$.

Step 7: While the I/O values of the process remain within the areas $\bar{O}_{\eta_\lambda} \cup \bar{O}_{\eta_{\lambda+1}}$ wait for a time period $t \le t_{\max}$ or till the system approaches a steady state namely an operating point $\rho^*_{\lambda+1}$, else return to Step 1.

Step 8: Set $\lambda = \lambda + 1$ and return to Step 3.

It is important to note that the safe controller, which is referred to in Step 1, is a controller that has been determined, usually heuristically, to lead to stable (and certainly degraded) operation over the entire region of operation of the process.

III. Supervisory Scheme

In the present section we present a supervisory scheme for SWSS Controllers. The proposed supervisory scheme is implemented as a software system, which may be developed using high level programming tools, like Matlab.

The functionality of the supervisory scheme is organized in the following functional units: Step-Wise Safe Switching unit, Common controllers design unit, Experimentation unit, Simulation unit, Identification unit, Trajectory cruise unit, Operating points unit and Expert system unit. The application of the supervisory scheme for any nonlinear process takes place in two consecutive phases.

During the first phase, which is called *initialization phase*, the supervisory scheme determines all the data which are required for the implementation of the SWSS Logic. Moreover, during this phase, all the decisions have to be made, concerning the type of the common controllers and the design approaches that will be used. More specifically, during the initialization phase the following functions take place:

a) The operating curve of the process is determined with the use of the trajectory cruise unit.

b) A set $L = \{\ell_1, \ell_2, \ldots, \ell_\mu\}$ of nominal operating points, that satisfy the dense web principle, are determined with the use of the operating points unit. Moreover, the corresponding linearized models S_i, as well as the target O_i and tolerance $\bar{O}_i$ operating areas are determined for $i = 1, \ldots, \mu$. To achieve that, the operating points unit uses the identification unit, that performs identification to derive the linearized models, exploiting input/output data provided by the experimentation unit. In the special case when the process model is known, the models S_i may be derived by linearization of the known nonlinear model around the corresponding operating points. The experimentation unit is also used for the determination of the target and tolerance operating areas, which also requires experimental data.

c) The operator, with the support of the expert system unit, selects the approach to be used for the design of common controllers.

d) The common controllers unit determines a common controller $C_{i,i+1}$ for each pair of adjacent operating points ℓ_i and ℓ_{i+1}.

The second phase of the supervisory scheme, which is called e*xecution phase*, comprises the SWSS unit that implements the SWSS algorithm, based on the data derived during the initialization phase.

In special cases, the execution phase may need to be interrupted and then the execution of the whole or part of the initialization phase may need to be repeated. This may be required for example if a) the operating conditions of the process change, b) the range of the process operating area has to be extended beyond the area covered by the current set L of nominal operating points or c) the applied common controllers have to be replaced by controllers of other type.

The flowchart and the block diagram of the proposed supervisory scheme are presented in Figures 1 and 2, respectively.

A. Step-Wise Safe Switching Unit

The Step-Wise Safe Switching unit implements the SWSS algorithm presented in Section 2. More specifically, this unit orchestrates controller switching based on the present and the desired operating point. The operation of this unit requires knowledge of the target O_i and the tolerance $\bar{O}_i$ operating areas for $i = 1, \ldots, \mu$, as well as a set of common controllers $C_{i,i+1}$, $i = 1, \ldots, \mu - 1$.

B. Common Controllers Design Unit

As already mentioned in Section 2, the implementation of the SWSS algorithm requires knowledge of a set of common controllers $C_{i,i+1}$, $i = 1, \ldots, \mu - 1$. Each common controller $C_{i,i+1}$ corresponds to a pair of adjacent operating points ℓ_i and ℓ_{i+1} and is designed to satisfy a desired union set $\wp_i \cup \wp_{i+1}$ of design requirements, where $\wp_i$ and $\wp_{i+1}$ are the design requirements for S_i and S_{i+1}, respectively. The design of a common controller $C_{i,i+1}$ may be formulated in the following two forms:

a) as a control design problem for multilinear models,

b) as a robust control design problem.

The common controller unit may implement both these approaches. The selection between the two approaches is performed by the operator based on the characteristics of the process and its linearized models, the structure of the controller to be designed, as well as the desired design requirements. The expert system presented in a forthcoming section may also make suggestions regarding the selection between these two approaches. The main characteristics of the two design approaches are presented in the following.

It is important to note, that in most practical applications, the common controller unit is executed off-line to derive the required set of common controllers, thus it does not increase the computational burden of the controller's implementation.

1. Controller Design Unit for Multilinear Models

The design of common controllers $C_{i,i+1}$ may be performed following design techniques for multilinear models. More specifically, we consider a system whose behavior switches between the two linear models S_i and S_{i+1}. The application of a common controller $C_{i,i+1}$ to the multilinear model should achieve: a) satisfactory performance of the corresponding closed-loop system within the range of validity of each linear model S_i and S_{i+1} and b) safe and satisfactory performance of the corresponding closed-loop system for all transitions between the two models S_i and S_{i+1}.

Checking whether a candidate controller satisfies the aforementioned performance requirements can be performed either in an analytic way or using simulations. Some results towards the analytic derivation of sufficient conditions for safe transitions within SWSS for the case of known first-order affine-linear models, have been presented in [13]. However, the use of simulation based approaches provides the possibility to extend significantly the class of performance requirements, the class of candidate controller structures, as well as the class of multi-model plants under consideration.

The proposed unit implements a heuristic approach to the design of common controllers. The proposed algorithm is generic, in the sense that is does not depend on the degree or the specific structure of the process model, or even the design goal under consideration. Moreover, the algorithm is very simple to use and it can be applied for the derivation of several types of common controllers. However, the increase of the number of controller parameters increases the numerical complexity of the algorithm. The specification of the heuristic algorithm for the case of PI common controllers of incremental form has been presented in [15].

According to [15], at the first steps of the proposed algorithm, search is performed inside an initial rectangle search area within the space of controller parameters. The points of search are determined by a web of points. The search within the initial area is repeated twice by duplicating the density of the web. If the second search also fails to determine a set of common controllers, the algorithm determines for each linear model S_i and S_{i+1} the rectangles R_i and R_{i+1}, respectively, within which controllers that satisfy the performance requirements have been found. Then the algorithm proceeds with searching repeatedly within a rectangle, which is called the *union rectangle* and is determined as the smallest rectangle that includes in its interior both R_i and R_{i+1}. The density of the web is duplicated at each repetition of the search. Moreover, at each repetition of the search within the union rectangle, the *intersection rectangle* of R_i and R_{i+1} is determined and compared with the corresponding intersection rectangle determined at the previous repetition. The search within the union rectangle will be repeated until a set of common controllers is determined, or until the size of the intersection rectangle does no longer increase. This repeated search within the union rectangle intends to determine with a satisfactory accuracy the intersection rectangle, that is the area inside which common controllers are expected to be found. It is important to determine the whole extent of the intersection area, since otherwise points of the controller parameter space corresponding to common controllers may be missed.

Once the intersection rectangle has been determined, the algorithm proceeds with searching within the intersection rectangle. The search within the intersection rectangle will be repeated twice, with a web of double density at the second repetition. If the algorithm fails to determine a set of common controllers after two consecutive searches within the intersection rectangle, then it stops. In any case, the algorithm will stop if any of the following occurs: a) if a set of common controllers has been found, b) if the density of the web becomes smaller than a threshold value ε, or if the number of search repetitions exceeds a maximum value $I_{\max}$.

As already mentioned, the heuristic algorithm requires knowledge of an initial area of search within the controller parameter space, inside which the algorithm will start seeking for common controllers. The initial search area represents an estimation, that should be available before the implementation of the algorithm, of the intervals inside which the common controller parameters are expected to be found. The determination of this initial search area should be based on any available a priori information about the process, as well as any available information about the range of the controller parameters. For example, restrictions may be imposed on the allowed range of the controller parameters based on the actuator constraints of the plant (see for example [15]).

2. Robust Controllers Design Unit

The design of each common controller $C_{i,i+1}$ may be formulated as a robust control design problem. This formulation requires two basic steps:

Step 1: First, we have to determine a local uncertain model for the plant, as well as to model the corresponding uncertainty, in such a way that the two linear models S_i and S_{i+1} lie within the range of uncertainty of the aforementioned local uncertain model. For example, when both linear systems S_i and S_{i+1} have the same structure, we may select S_i as a nominal system subject to parametric uncertainty, whose range is determined so as to include the parameter values of the system S_{i+1}.

Step 2: Next, we have to design a robust controller that achieves the desired performance requirements for all uncertainties within the prespecified range, and consequently for both linear models S_i and S_{i+1}.

In the field of robust control, a variety of control design problems have been solved. A case of special interest for the control of industrial processes is that of robust dynamic controllers, as for example PI or PID controllers, which may be designed to serve a variety of design requirements. Other interesting cases for control problems in industrial

environment are robust controllers designed to achieve input/output decoupling and/or disturbance rejection for uncertain systems subject to constraints, as the case of uncertain singular systems.

The selection of the approach to be used is performed by the operator based on the characteristics of the process, the desired requirements for the closed-loop system, as well as the structure of the controller to be applied. The expert system presented in a forthcoming section may also make suggestions regarding the selection between the several robust control design approaches. The proposed robust controller design unit implements robust control design that serves the following design requirements:

a) Robust command following with PI ([9]) or PID ([11]) controllers .

b) Robust pole assignment with dynamic controllers ([10]).

c) Robust exact model matching with dynamic controllers ([8]).

d) Robust disturbance rejection for generalized state space systems with static controllers.

e) Robust input-output decoupling for generalized state space systems with static controllers ([12]).

The aforementioned approaches concern linear uncertain systems with nonlinear uncertain structure. Since the linearization of nonlinear models results in linear systems with coefficients that depend nonlinearly on the corresponding operating point input and output values, it becomes obvious that the uncertain models derived in Step 1 are expected in general to present parametric uncertainty of nonlinear structure. Thus the proposed robust control approaches are particularly suited for the design control problem under consideration.

C. Experimentation Unit

This unit is responsible for the execution of small scale experiments. More specifically this unit is called to execute open or closed loop experiments with the application of specific input functions, that correspond to small amplitude deviations from the current operating conditions, and collect the corresponding measurements of the plant's variables. The selection of the specific input function depends on the functionality of the unit that asks for the experimental results. For example, if the experimental results are to be used in order to determine the operating curve or target and tolerance operating areas, small deviation step input functions have to be used. If the experimental results are to be used for identification purposes, more rich input functions may have to be applied, as for example sinusoidal signals.

The applied input should be selected to keep the plant's I/O trajectories close to a nominal operating point. If the applied input drives the I/O trajectories far away from the nominal operating point, the experimentation unit should proceed, for safety reasons, in corrective actions that keep the trajectories within the desired range.

When the plant's nonlinear model is known, the experimentation unit uses the simulation unit to derive the desired data, instead of performing experiments on the plant.

D. Simulation Unit

In cases when the nonlinear model of the plant is known, a simulation unit may be implemented to provide input/output data of the plant. In this case, the experimentation unit uses the simulation unit instead of performing experiments on the plant. The use of the simulation unit, whenever this is possible, simplifies significantly the implementation of the supervisory scheme, since it provides all the input/output data which are required for the initialization phase, without disturbing the plant's normal operation.

E. Identification Unit

When the nonlinear model of the plant is unknown, or it is too complex to be used for control purposes, identification techniques may be applied in order to derive the linear approximate models S_i, that describes satisfactorily the plant's behavior around the corresponding operating point ℓ_i. The proposed identification unit implements least squares recursive identification (see for example [16]). The required input/output data are provided by the experimentation unit. Special care has to be taken in selecting the amplitude and the type of the input signal, based on any available information about the plant's characteristics.

F. Trajectory Cruise Unit

The set of operating points of the process, which are all the points $\ell_i = [Y_i, U_i]$ of the input-output space that satisfy condition (2), form a curve in the input-output space, which is called *operating curve* of the process.

The proposed unit is used for the determination of the operating curve. The functionality of this unit is supported by the experimentation unit, which either realizes experiments on the plant or uses the simulation unit for the case of processes with known models. More specifically, the trajectory cruise unit calls repeatedly the experimentation unit using as input step signals with small amplitude. The input/output values at which the process settles at steady state are recorded as operating points.

G. Operating Points Unit

The operating points unit is used to determine a set of nominal operating points that satisfy the dense web principle, that is they satisfy conditions (3) and (4). In many practical applications, an initial set of nominal operating points, which are significant for the operation of the process, may be a priori available. The operating points unit determines for each nominal operating point the corresponding linearized models, as well as the corresponding target and tolerance operating areas. The determination of the linearized models is performed either by calling the identification unit or using linearization in case the plant's model is known. The determination of the operating areas is performed by calling repeatedly the experimentation unit using as input appropriate step signals of small amplitude. Then the satisfaction of conditions (3) and

(4) is checked. In case these conditions are not satisfied, the operating points unit selects additional candidate nominal operating points that lie on the intervals of the operating curve limited by the current selection of nominal operating points. This procedure is repeated until conditions (3) and (4) are satisfied for all pairs of neighboring nominal operating points.

In case there is not available any a priori set of nominal operating points, then the operating points unit should select an initial candidate set that extends over the whole operating area of interest. Then the execution of the unit should proceed as previously described, until the dense web principle conditions are satisfied. Note that the range of the operating area of interest should be initially determined by the operator, based on the information about the range of the input/output values of the process.

Note also that the selection of any new candidate nominal operating points is performed with the use of the trajectory cruise unit that determines the operating curve of the process.

H. Expert System Unit

The supervisory scheme comprises an expert system, which is used to assist the operator in decision making with respect to several issues of the supervisory scheme's functionality, for which more than one approaches are available. The most important functionality of the expert system concerns the selection of the approach that will be used for the design of common controllers. As already clarified, the proposed common controller design unit may design common controllers following a heuristic design approach or a variety of robust control design approaches. In this case a decision has to be made based on the available information about the characteristics of the process, the available knowledge about the process model, the design requirements and any restrictions concerning the structure of the controller to be applied. The expert system unit implements a decision tree for the selection of the control approach to be applied dependent on the specific process characteristics.

IV. Conclusion

In the present paper we have presented a supervisory scheme in the form of an automation software system, which is used for the implementation of Step-Wise Safe Switching. Step-Wise Safe Switching is a supervisory control technique which is suitable to control nonlinear plants that cannot be adequately served by a single field controller. Step-Wise Safe Switching, as other logic based switching techniques, provides the possibility to control a complex nonlinear plant using simple structure field controllers. However, its implementation requires a number of off-line and on-line actions, including experimentation, identification, field controller design, controller switching, etc. The proposed supervisory scheme organizes these actions in eight distinct cooperating functional units. The functionality of these units, the basic guidelines for their implementation, as well as the data exchanges between them and their interconnection have been presented. It is important to note that the supervisory scheme is a generic tool that may be easily applied for a variety of industrial control processes.

References

[1] F.N. Koumboulis, R.E. King, A. Stathaki, "Logic-Based Switching Controllers – A stepwise safe switching approach", *Information Sciences*, vol. 177, pp. 2736–2755, 2007.

[2] A. Leonessa, W.M. Haddad, V. Chelaboina, "Nonlinear system stabilization via hierarchical switching control", *IEEE Trans. on Autom. Control*, vol. 46, pp. 17-28, 2001.

[3] M.W. McConley, B.D. Appleby, M.A. Dalheh, E. Feron, "A computationally efficient Lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees", *IEEE Trans. on Autom. Control*, vol. 45, pp. 33-49, 2000.

[4] E.F. Costa, V.A. Oliveira, "Gain scheduled controllers for dynamic systems using sector nonlinearities", *Automatica*, vol.38, pp. 1247-1250, 2002.

[5] J. Ackermann, A. Barlett, D. Kaesbauer, W. Sienel, and R. Steinhouser, *Robust Control Systems with Uncertain Physical Parameters*, London, U.K.: Springer-Verlag, 1993.

[6] B.R. Barmish, *Tools for Robustness of Linear Systems*, McMillan Publ. Co., New York, 1994.

[7] F. N. Koumboulis, M. G. Skarpetis, "Robust triangular decoupling with application to 4WS cars", *IEEE Trans. on Automatic Control*, vol. 45, no. 2, 2000, pp. 344-352.

[8] F. N. Koumboulis, M. P. Tzamtzi, M. G. Skarpetis, "Robust Exact Model Matching for SISO Systems via Finite Precision Dynamic Output Feedback", *14th Mediterranean Conf. on Control and Autom (2006)*, Italy, 2006.

[9] F. N. Koumboulis, M. G. Skarpetis, M. P. Tzamtzi, "Robust PI Controllers for Command Following with Application to an Electropneumatic Actuator", *14th Mediterranean Conf. on Control and Autom. (2006)*, Italy, 2006.

[10] F.N. Koumboulis, M.P. Tzamtzi, M.G. Skarpetis, "Finite Precision Controllers for Robust Pole Assignment of Linear Systems with Nonlinear Uncertain Structure", *11th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2006)*, Prague, Czech Republic, September 20-22, 2006, pp. 725-732.

[11] F.N. Koumboulis, M.G. Skarpetis, M.P. Tzamtzi, "Robust PID Controller Design with Application to a Flight Actuator", *32nd Annual Conf. IEEE Industrial Electr. Soc. (IECON'06)*, Paris, France, 7-10 Nov. 2006, pp. 4725-4730.

[12] F.N. Koumboulis, M.G. Skarpetis, M.P. Tzamtzi, "Robust Input – Output Decoupling for Singular Systems with Nonlinear Uncertain Structure Presentation", *European Control Conf.*, Kos, Greece, 2007, pp. 3946-3950.

[13] F.N. Koumboulis, M.P. Tzamtzi, "Towards Analytic Solutions of Step-Wise Safe Switching for Known Affine-Linear Models", *Int. Conf. on Numerical Analysis and Applied Mathematics 2007 (ICNAAM 2007)*, Corfu, Greece, Sep. 2007.

[14] F.W. Glover, G.A. Kochenberger (Eds.), *Handbook of Metaheuristics*, in Series: International Series in Operations Research & Management Science, vol. 57, 2003, Springer.

[15] F. N. Koumboulis, "On the heuristic design of common PI controllers for multi-model plants", *10th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA 2005)*, Italy, pp. 975-982, 2005.

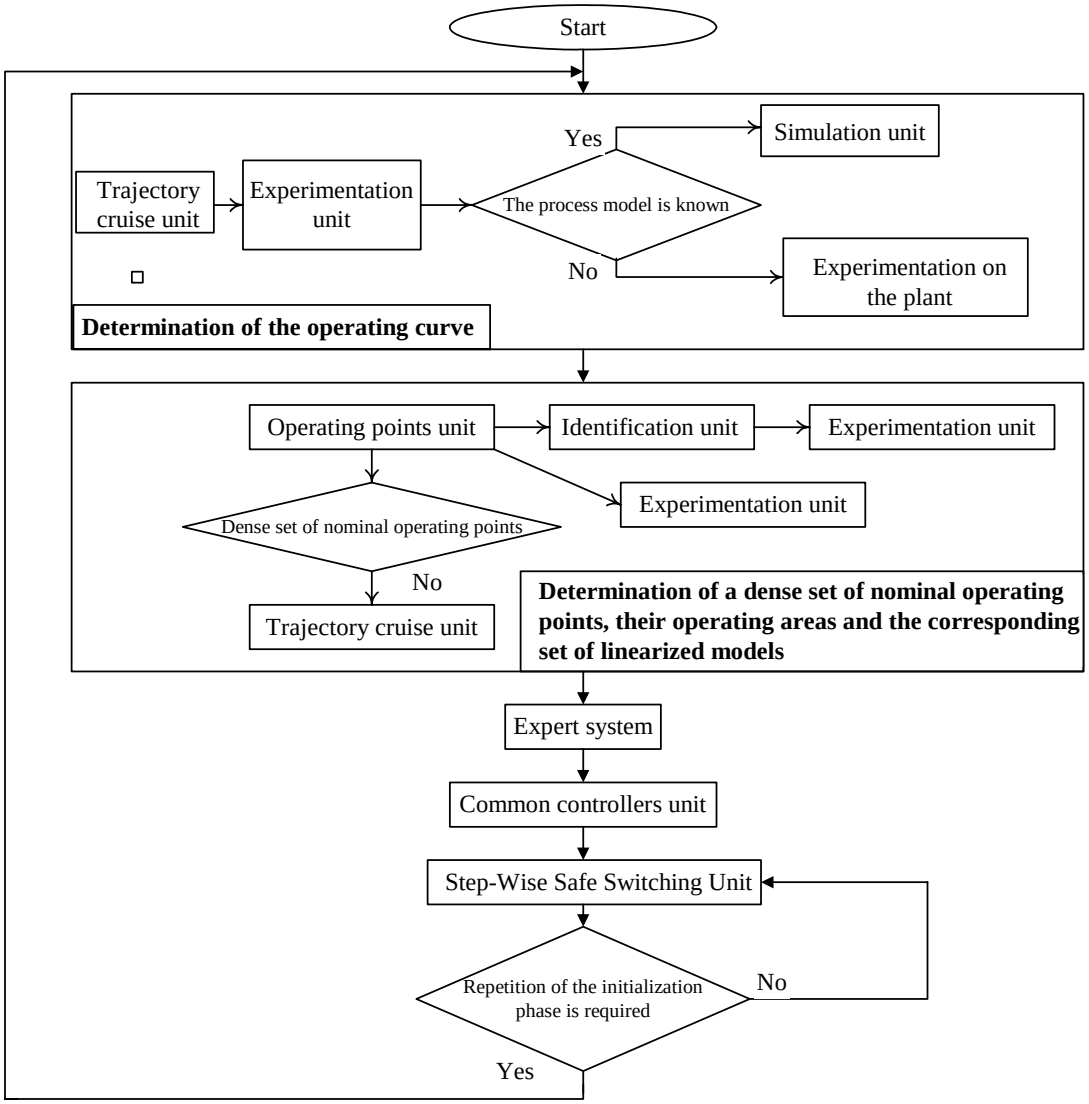

Fig. 1 Flow chart of the Supervisory Scheme

: declares the time sequence of the unit's execution within the supervisory scheme

: declares that the unit at which the arrow initiates activates the execution of the unit at which the arrow ends

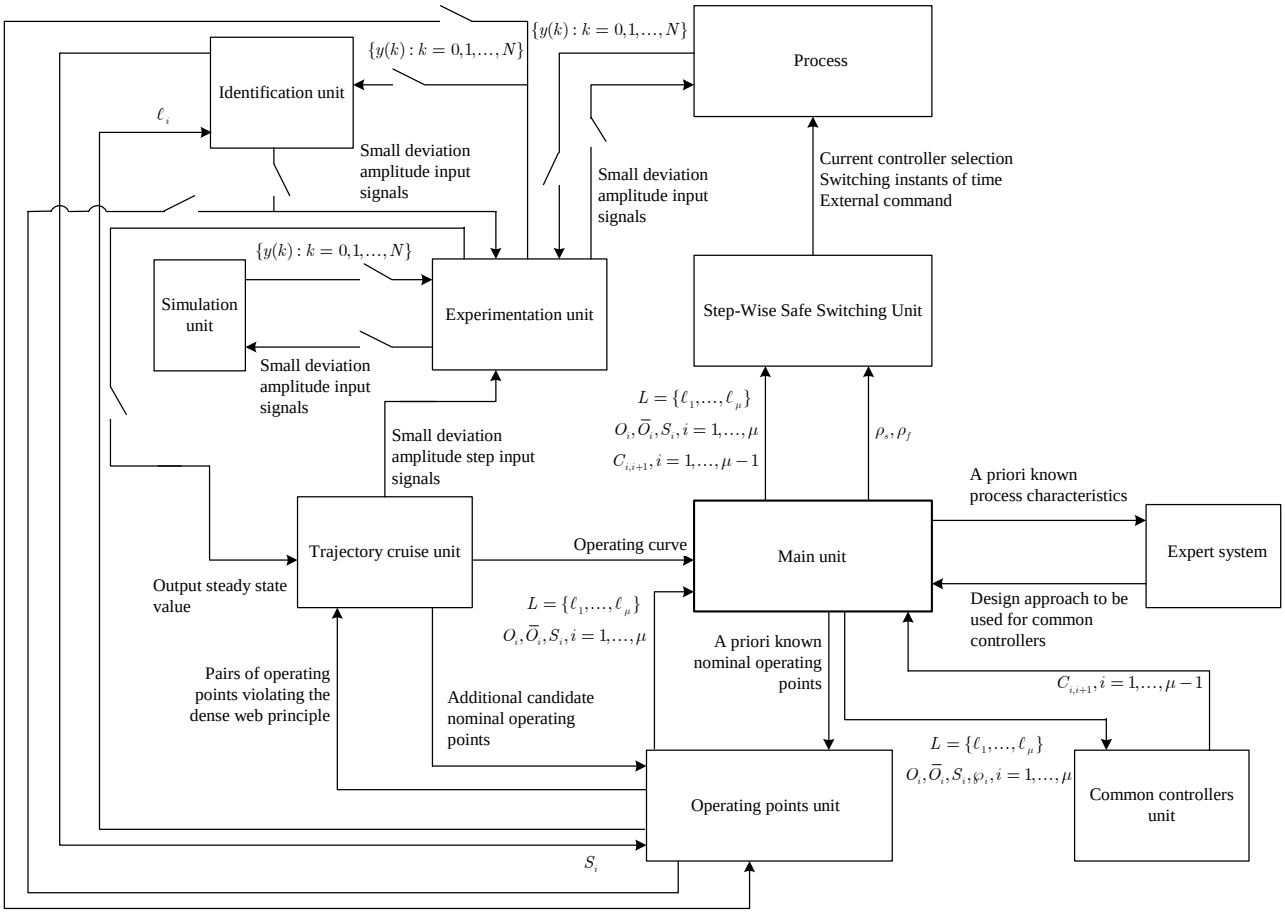

Fig. 2 Block Diagram of the Supervisory Scheme