We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

4

Knowledge-Based Control for Robot Arm

Aboubekeur Hamdi-Cherif
[1]Qassim University, Computer Science Department, Buraydah
[2]Université Ferhat Abbas Setif (UFAS), Computer Science Department, Setif
[1]Saudi Arabia
[2]Algeria

1. Introduction

The present research work reports the usability of knowledge-based control (KBC) as an alternative control method with specific concentration robot arm (RA). This novel control approach is based on the combination of inferences and calculations. It is dictated by the advent of microprocessor technology which has been one of the sources of inspiration for techniques spanning the whole spectrum of controllers design. KBC can contribute to build simple proportional integral and derivative (PID) control schemes (Åström et al., 1992) to large classes of regulators such as self-tuning regulators and model-reference adaptive controllers, among others (Hanlei, 2010). Because knowledge base systems (KBSs) research has focused on implementing heuristic techniques, the corresponding knowledge-based controllers can justly be considered as the next logical step in control design and implementation (Handelman *et al.*, 1990). The main characteristics of knowledge-based controllers is that they incorporate years-long human expertise under the form of machine-understandable heuristic rules. In KBC, the knowledge elicited from human experts is codified and embodied within the KB in the form of IF-THEN rules. As a result, the KB technology takes into account the increase in system complexity. This sophistication is naturally encountered as efforts are made to stretch the limits of system performance and integrate more capabilities as a response to technological advances (Calangiu *et al.*, 2010). In addition, the inherent ability of KBSs to support incremental expansion of capabilities and provide justification for recommendations or actions is offered by conventional programming techniques. Serious considerations are being given to increasing system reliability by predicting algorithm failure in RAs control and reconfiguring control laws in response to algorithm failure due to instability/chattering, or large RAs parameter variations.

The knowledge-based control (KBC) benefits as applied to RA are to:
- Implement/incorporate heuristics within the RA control schemes.
- Diagnose or predict algorithm failure.
- Identify changes in RA parameters or structure.
- Recalculate control laws based upon knowledge of the current RA parameters.
- Select appropriate control laws based on the current RA responses.
- Execute supportive control logic which has been used for practical controllers in the past.
- Provide an explanation of the situation to the user as and when requested.

However KBC approach is not without issues. Indeed, KBC design problem requires elicitation / acquisition and coding of the "useful expertise", gained by humans over a lifetime. It is highly difficult to find proper ways of extracting this expertise from the human experts. Discerning "usefulness", avoiding unnecessary data and finding ways of optimizing this knowledge representation is not a straight-forward task. How far are we from common sense-based control? This paper extends the limits of RA control using KBC approach in order to reach this distant end.

The main issue of the present work is to answer positively our central question, *i.e.*, whether it is possible to integrate the diversified methods dealing with dynamical systems control exemplified by RA control, while concentrating on KBC as an alternative control method. We describe the epistemological characteristics of a framework that is believed to integrate two distinct methodological fields of research *i.e.*, artificial intelligence (AI)-based methods where KBC is partly rooted, on the one hand, and control theory, where RA control is formulated, on the other hand. Blending research from both fields results in the appearance of a richer research community. Emphasis is now made on RA control as a prelude to other classes of robotic systems; ultimately enhancing full programmable self-assembly compounds (Klavins, 2007). The chapter is organized as follows. In Section 2, the main KBC issues are discussed. Section 3 presents KBC within the general area of intelligent control and places KBC with respect to generalized hybrid control. Section 4 summarizes RA control in standard mathematical terms. Section 5 deals with an architecture for KBC for RA as an alternative control method followed by a conclusion and future developments.

2. Knowledge-based control issues

2.1 Our specific problem

The specific problem we want to tackle can broadly be expressed as follows:
Given:
- A plant configuration library describing the actual system to be controlled,
- A library of control algorithms with various degrees of complexity,

Find:
> One (class of) algorithm (s) that control one plant configuration.

Application:
> Address simulation of RAs dynamics under various control schemes.

For doing this, consider two complementary environments, *i.e.* a numeric environment responsible for making calculations (trajectory, control law,…) and a symbolic environment responsible for making logical inferences incorporating human experience. These two environments are the main components of any KBC architecture. Two modes of operation are therefore possible. In the numerical or exploitation mode, the program generates the outputs using imposed algorithms. In the inferential or exploration mode, the algorithm is not known before hand. Using the codified expertise in the KB, the program has to choose it from a library before firing the numeric mode. For the sequel, we first start by considering standard RA control and then KBC within the larger context of intelligent control.

2.2 From standard RA control to KBC

RA control is the process whereby a physical system, namely a set of robotic linked arms, is made compliant with some prescribed task such as following an imposed trajectory or

keeping in pace with a given angular velocity (Siciliano, 2009). Welding and assembly-line robots are popular examples of RA industrial applications. RA control is a much diversified field. As a result, it makes concentrated research a difficult task. While RA control has been extensively studied from the pure control side (Lewis *et al.*, 2003), for the last four decades, or so, very little attention has been made with regard to KBC. Indeed, the symbolic approach efforts as applied to control at large remain quite isolated (Martins *et al.* 2006). Our fundamental aim is to contribute to the integration of RA control within KBC, considered within a larger intelligent control methodology; this latter being defined as a computational methodology that provides automatic means of improving tasks from heuristics (Hamdi-Cherif & Kara-Mohamed, 2009). As a subfield of intelligent control, KBC attempts to elaborate a control law on the basis of heuristics. KBC aim is therefore consistent with the overall goal of intelligent control and, as such, automatically generates a control law from heuristic rules and actual facts describing the actual RA status (control law, errors, trajectories).

2.3 Pending control issues

Although KBC is a promising applied research area, there remain many challenges to be addressed. The main pending issues are:

- the system under control can be very complex (*e.g.* nonlinearities in robot arm (RA)) ;
- our knowledge of the system is imprecise (*e.g.* unknown RA parameters, unknown conditions of operation) although gradually increasing during operation, in the optimistic case of successful identification process,
- the influence of the environment is strong (*e.g.* outside perturbation, modeling errors), may vary and may even influence the current task,
- the goal of the system is described symbolically and may have internal hierarchy to be further investigated and structured.

If the answer to these challenges can be obtained from human experts, then this knowledge is codified within the KB by knowledge engineers. If the answer is unknown, then offline experimentation is done by control engineers to gradually build an answer and codify it in the KB. In any case, the KBC designer has to constantly upgrade the KB with human expertise and/or manual experimentations.

2.4 Overview of related works

Few authors have addressed the issue of designing and developing systems that cater for general-purpose RA control. For example (Yae *et al.* 1994) have extended the EASY5 - the Boeing Engineering and Analysis SYstem - incorporating constrained dynamics. (Polyakov *et al.* 1994) have developed, in MATHEMATICA™, a symbolic computer algebra system toolbox for nonlinear and adaptive control synthesis and simulation which provides flexible simulation *via* C and MATLAB™ code generation. MATHEMATICA™ has also been used in a simulation program that generates animated graphics representing the motion of a simple planar mechanical manipulator with three revolute joints for teaching purposes (Etxebarria, 1994). A toolbox is available for RA control running on MATLAB™ (Corke, 1996). For supplementary and more general applications of computer algebra to CACSD (computer-aided control system design), we refer to (Eldeib and Tsai, 1989). Recent research directions aim at the development of operating systems for robots, not necessarily for the

RA class. An overview of ROS, an open source robot operating system has been recently reported. ROS is not an operating system in the traditional sense of process handling and scheduling. It provides a structured communications layer above the host operating systems of a heterogeneous cluster. ROS was designed to meet a specific set of challenges encountered when developing large-scale service robots as part of the so-called STAIR project [http://stair.stanford.edu/papers.php]. The way how ROS relates to existing robot software frameworks, and a brief overview of some of the available application software which uses ROS are reported in (Quigley, *et al.* 2009). However, none of these works addressed the issue of using the KBC approach to solve the RA control problem. Hence our solution.

3. Solution components

3.1 KBC within intelligent control
3.1.1 The area of intelligent control
One of the fundamental issues that concerns intelligent control is the extent to which it is possible to control the dynamic behavior of a given system independently of

- its complexity,
- our capability of separating it from the environment and localizing it,
- the context in which this system operates,
- the forms of knowledge available and the categories it manipulates,
- the methods of representation.

As formulated, this issue cannot be handled by either control theory or artificial intelligence (AI). Indeed, control theory has a very localized mostly numerical vision of the problem. This prevents it from looking beyond the localized constraints self-imposed by the designer and hidden within the mechanism of the mathematical representation. From the standpoint of AI, the available knowledge-related methods cannot easily handle dynamic systems and have very little consideration for numerical manipulation. Indeed, computations of margins of stability, controllability, observability are alien to AI. Moreover, both control theory and artificial intelligence (AI) cannot properly operate out of the operations research (OR) paradigm. Its queues, graphs and game-theoretic situations are typical of the variety of control applications. That is why an early proposal for the definition of intelligent control is to consider this field as the intersection of the three previously-cited disciplines namely control theory, AI and OR, (Saridis, 1987). Other fields such as soft computing represented by fuzzy, genetic, neural systems and their combinations, on the one hand, and cognitive science, on the other hand have been progressively integrated within the intelligent control discipline over the last three decades, or so (Lewis *et al*; 2003).

3.1.2 Landmarks of intelligent control
Intelligent control is a term that first appeared in the seventies and later developed in (Saridis , 1987). An early, but constantly refined definition of this field describes itself as that area beyond adaptive, learning and self-organizing systems which represents the meeting point between artificial intelligence (AI), automatic control (AC) and operations research (OR). A tremendous body of literature has been developed to account for the description / design within this novel paradigm. International intelligent control symposia have been

held every year since 1985 and numerous contributions appear regularly in the specialized and thoroughly documented literature where novel original definitions and applications of the field are proposed *e.g.* (Rao, 1992), (Handelman *et al.*, 2010). Extensions of the field are reported by (Åström, 1989), (Åström and MacAvoy, 1992), and (Cellier *et al.*, 1992). Other approaches have also been considered by researchers like the cognition-oriented approach with applications, (Meystel, 1994). Among the several advanced theoretical and applied results are those due to (Saridis, 1987) who proposed the so-called an entropy-based theory for hierarchical controller design based on the so-called "principle of decreasing precision with increasing intelligence". More recently, methods concentrated on soft computing methods such as:

a. *Neural networks (NNs).* In (Kwan *et al.*, 2001), a desired compensation adaptive law-based neural network (NN) controller is proposed for the robust position control of rigid-link robots where the NN is used to approximate a highly nonlinear function. Global asymptotic stability is obtained with tracking errors and boundedness of NN weights. No offline learning phase is required as learning is done on-line. Compared with classic adaptive RA controllers, parameters linearity and determination of a regression matrix are not needed. However, time for converging to a solution might be prohibitive.

b. *Fuzzy-Genetic.* In (Merchán-Cruz and Morris, 2006), a simple genetic algorithm planner is used to produce an initial estimation of the movements of two RAs' articulations and collision free motion is obtained by the corrective action of the collision-avoidance fuzzy units.

3.1.3 Scope of intelligent control

Intelligent control as a discipline provides generalization of the existing control theories and methods on the basis of the following elements (Astrom and MacAvoy, 1992):

*   combined analysis of the plant and its control criteria,
*   processes of multisensor operation with information (knowledge) integration and recognition in the loop,
*   man-machine cooperative activities, including imitation and substitution of the human operator,
*   computer structures representing these elements.

3.1.4 Specific issues in intelligent control

One of the main drawbacks of intelligent control is that, up to now, there is no established terminology identifiable with this discipline. There remains an inertia in following conventional views and recommendations. This attitude hinders the development of intelligent control ideas and methods. For the purpose of immediate applications, we will concentrate on a small area of intelligent control. On the one hand, we will focus on the use of numerical/exploitation (procedural) and inferential/exploration processing (declarative, rule-based) systems. The former describe the RA control algorithms while the latter represent the way in which the expertise is explored and used in firing the adequate algorithm according to the actual situation (plant, errors). In the multiresolutional control architectures for intelligent machines proposed by (Meystel, 1991), the general structure of the intelligent controller is described by a set of feedback loops. Each one of these loops is declared for a particular resolution level and works with a different time-scale. Resolution of

a given level is defined by (Meystel, 1994) as "the size of undistinguishability zone for the representation of goal, plan and feedback law."

3.2 KBC as a generalized hybrid control methodology
3.2.1 Hybrid control
KBC can alternatively be considered with respect to hybrid control. In the early sixties, the discipline of hybrid control referred to controlled systems using both discrete and continuous parts. This discipline spanned a substantial area of research from basic switched linear systems to full-scale hybrid automata. Later, symbolic control methods came to include abstracting continuous dynamics to symbolic descriptions, instruction selection and coding in finite-bandwidth control applications, and applying formal language theory to the continuous systems domain. A number of results have emerged in this area with a conventional control-theoretic orientation, including optimal control, stability, system identification, observers, and well-posedness of solutions. At the same time, symbolic control provides faithful descriptions of the continuous level performance of the actual system, and as a result, provides a formal bridge between its continuous and the discrete characteristics (Egerstedt *et al.,* 2006).

3.2.2 Generalized hybrid control
Generalized hybrid control is meant to incorporate logic and control, whether discrete or continuous. For our KBC concern, we will consider KBC as an integration of pure control and logical inference as expressed by either propositional logic or first-order logic (FOL). As a result, KBC addresses questions at the highest level, *i.e.,* at the level of symbols, and as such stands half-way between computer science and logic, on the one hand and control theory, on the other hand. A whole research area is to be investigated whereby results from hybrid control are to be mapped onto generalized hybrid control. As for now, a new line of research in hybrid systems has been initiated that studies issues not quite standard to the controls community, including formal verification, abstractions, model expressiveness, computational tools, and specification languages. These issues were usually addressed in other areas, such as software engineering and formal languages (Hamdi-Cherif, 2010).

3.3 Overall architecture for intelligent control
Intelligent machines are those that perform anthropomorphic tasks, autonomously or interactively and/or proactively with a human operator in structured or unstructured, familiar or unfamiliar environments. The intelligent controller represents the driving force that allows intelligent machines achieve their goals autonomously. It embodies functions of inferences as well as conventional control based on numeric processing. When such environments treat more than one state of the process to be controlled, as in the case of RA control, then it is careful to separate between control and inference, both functionally and architecturally. To this end we propose, in Figure 1, an overall architecture for intelligent control which considers the following levels:

3.3.1 Formulation level
At the formulation level, we find a hierarchical task formulation / task negotiation process. In the worst case situation, this formulation elaborates a model of an imprecise and incomplete plant.

3.3.2 Controller-plant matching level

This level uses knowledge from the KB to decide which controller algorithm is suitable for a given plant when operating under some prescribed user-defined specifications and other additional constraints. This level is further expanded in Figure 2.

3.3.3 Reasoning level

At this level, a KB contains the necessary knowledge to solve the controller-plant matching problem on the basis of the formulation. For the obtainment of the final controller-plant matching, a hybrid numeric / symbolic system representation has to be used. Some trade-off tasks as part of control process have to be considered.
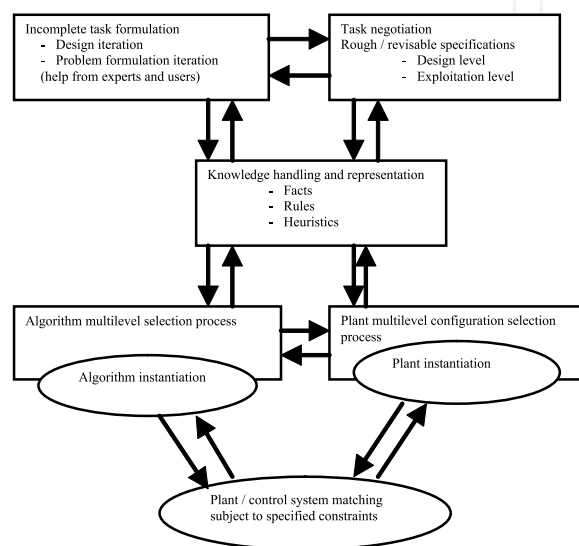

Fig. 1. Overall architecture for intelligent control

3.4 Inference issues

In evaluating any knowledge base system (KBS), and therefore any knowledge-based control (KBC) system, a wide range of criteria can be considered. We will define a generic framework for a description of the inferential part intervening in the KBC system. There are thousands of such systems ranging from free software (*e.g.* CLIPS, http://clipsrules.sourceforge.net/) to large industrial advanced packages such as G2™ from Gensym™ (http://www.gensym.com)

3.4.1 Knowledge base structure

Under this heading, we describe whether the system provides the representation by frames, messages, object-oriented languages, semantic nets, among others.

3.4.2 Type of logic involved

The usual types of logic available in KBS shells/systems are :
- *Propositional*: Boolean with no variables.
- *Predicate or first order logic (FOL):* Boolean with variables.
- *Temporal*: involves time in reasoning.
- *Fuzzy*: handles uncertainty, imprecision.

- *Non-monotone*: handles changing data.
- *Default*: handles situations like "most controllers are acceptable for these specifications".
- *Modal*: handles situations like "it's possible that", "it has been shown that this type of controller does not fit".
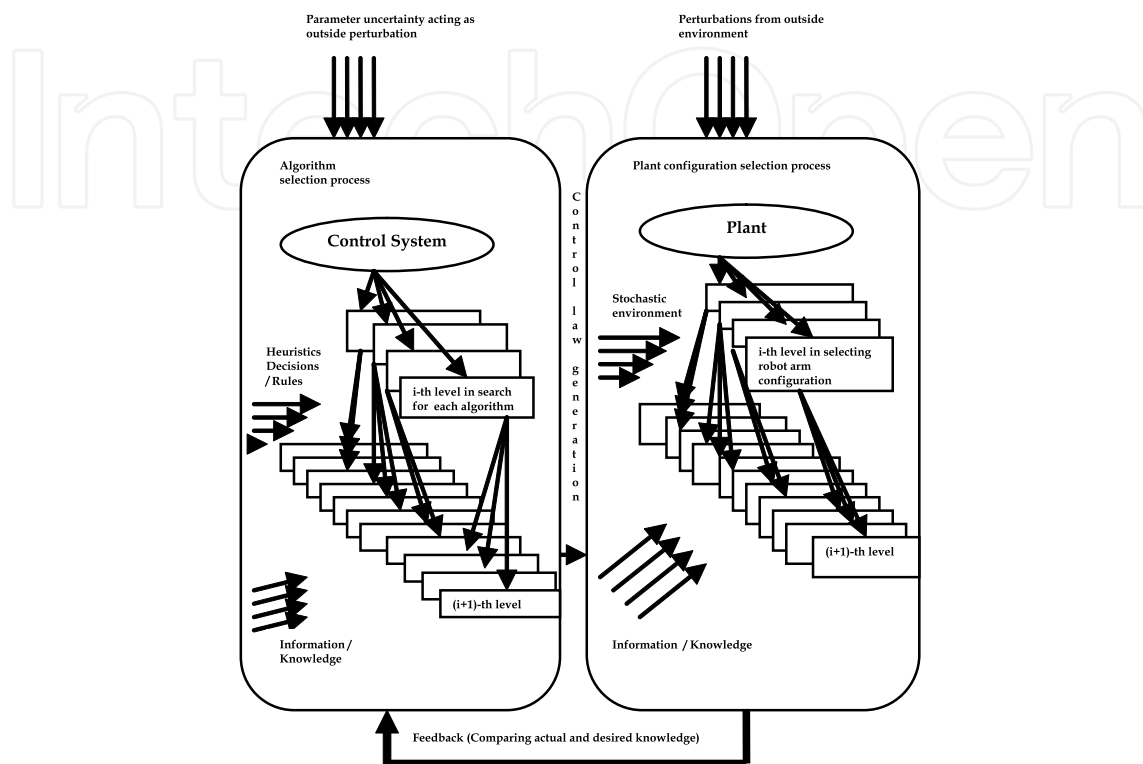

Fig. 2. Controller-plant matching problem

3.4.3 Reasoning strategy
- *Forward chaining*: hypothesis-driven.
- *Backward chaining*: goal-driven.
- *Hybrid chaining*: combining both forward and backward chaining.
- *Blackboards*: for keeping set of hypotheses of partial and final solutions.

3.4.4 Knowledge issues
- *Knowledge management* : browsers, editors, workspaces, workspaces security.
- *Knowledge validation* : "what if's" simulation capabilities.
- *Knowledge building tools* : human interface quality, KB construction quality, natural language environment.
- *Knowledge debugging* : levels of tracing, rules reporting, quality in entry and knowledge management.

3.4.5 Explanation, truth and uncertainty
- *Explanation of reasoning*: why's, natural language explanations, messages, variables values representation.
- *Truth maintenance* : forward update, backward update.

- *Uncertainty management* : certainty factors, fuzzy-oriented management.

3.4.6 Miscellaneous
- *Interface with outside world*: data acquisition, data bases, other specialized software interfacing.
- *Other performance*: stand-alone off-line, real-time performance, networking, other advanced special features.

4. RA Standard control problem

4.1 Brief history
On the control side, we concentrate on some classes of control methods such as adaptive control and passivity-based control. The development of RA control algorithms has gone through at least three historical phases. The first is the model reference adaptive control and self-tuning control followed by the passivity approach and then by the soft computing methods. We report here the first two phases while the soft computing methods have been described in Section 3.1.3 above..

4.1.1 Model Reference Adaptive Control (MRAC) and Self-Tuning Control (STC)
The first phase (1978-1985) concentrated its efforts on the approximation approach. The methods developed during this period are well-documented in the literature and some review papers have been written for that period (*e.g.* Hsia, 1986). Researches were concentrated on issues expanded below.
a. *Model reference adaptive control approach* (MRAC) guided by the minimization of the error between the actual system and some conveniently chosen model of it. At the methodological level, this represents a traditional example of supervised learning based on comparison between the actual and desired outputs while trying to minimize the error between desired and actual values.
b. *Self-tuning control* based on performance criteria minimization.

4.1.2 Parametrization approach
The methods developed during the second period that followed with some time overlaps with the previous period, concentrated on the parameterization approach. The methods developed within this period can be further separated in two broad classes, namely inverse dynamics and passivity-based control.
a. *Inverse dynamics*
   The first set of methods treats the inverse dynamics-based control or computed torque method. It relies on the exact cancellation of all the nonlinearities in the system. In the ideal case, the closed-loop system is decoupled and linear. Stability in this case is based on the Lyapunov direct method. A dynamical system is said to be stable in the sense of Lyapunov if it has the characteristics that when it loses an un-restored energy over time, then it will stabilize at some final state, called the attractor. In Lord Kelvin's terms this means that conservative systems in the presence of dissipative forcing elements will decay to a local minimum of their potential energy. However, finding a function that gives the precise energy of a given physical system can be extremely difficult. On the

other hand, for some systems (*e.g.* econometric and biological systems), the Lyapunov function has no physical meaning.

b.   Passivity-based control

The second set of methods deals with passivity-based control. The aim is to find a control law that preserves the passivity of the rigid RA in closed-loop. Stability here is based on the Popov hyperstability method (Popov, 1973). One of the main motivations for using these control laws, as far as stability is concerned, is that they avoid looking for complex Lyapunov functions - a bottleneck of the Lyapunov-based design. These laws also lead, in the adaptive case, to error equations where the regressor is independent of the joint acceleration. The difficult issue of inertia matrix inversion is also avoided. At the opposite of inverse dynamics methods, passivity-based methods do not look for linearization but rather for the passivity of the closed-loop system. Stability is granted if the energy of the closed-loop system is dissipated. The resulting control laws are therefore different for the two previous classes.

4.2 Issues in adaptive and passivity RA control

From the vast literature on adaptive control, only a small portion is applicable to RA control. One of the first approaches to adaptive control, based on the assumption of decoupled joint dynamics, is presented in (Craig, 1988). In general, multi-input multi-output (MIMO) adaptive control provides the means of solving problems of coupled motion, though nonlinear robot dynamics with rapidly changing operating conditions complicate the adaptive control problem involved, even if there are also advantages when compared with the adaptive control of linear systems. Specialized literature has appeared in the field, *e.g.*, the interesting tutorial reported in (Ortega & Spong, 1989). As far as adaptive control is concerned, some methods assume that acceleration is available for measurement and that the inertia matrix inverse is bounded. Others avoid at least the boundedness constraint (*e.g.* Amestegui *et al.*, 1987) while passivity-based control avoids both limitations. We propose to classify the specialized contributions in the field as follows:

*a.   Parameter estimation: s*uch as the linear estimation models suitable for identification of the payload of a partially known robot, going back to (Vukabratovic *et al.*, 1984).

b.   *Direct adaptive control* of robot motion as studied by :

    1.   (Craig et al., 1987) in conjunction with model reference adaptive control (MRAC). Here stability is studied using strictly positive real transfer functions (SPR-TF).

    2.   (Slotine and Li, 1987) in conjunction with the so-called "MIT rule". Here the regulator is independent of the acceleration measurement and linear in the parameters.

    3.   Johansson has still improved the work of (Craig et al., 1987) in terms of stability. This method avoids matrix inversion and SPR-TF requirements (Johansson, 1990).

c.   *Decentralized control* for adaptive independent joint control as proposed by (Seraji, 1989).

d.   *Control and stability analysis* such as passivity-based control developed by (Landau and Horowitz, 1989).

4.3 RA dynamics

A standard mathematical model is needed for any RA control problem. The RA dynamics are modeled as a set of *n* linked rigid bodies (Craig, 2005). The model is given by the following standard ordinary differential equation in matrix form.

$$\tau(t) = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + V(\dot{q}) \qquad (1)$$

Time arguments are omitted for simplicity. The notations used have the following meaning:

q : joint angular position, *nx1* real vector.

$\dot{q}$: joint angular velocity, *nx1* real vector.

$\ddot{q}$: joint angular acceleration, *nx1* real vector.

$\tau(t)$: joint torque, *nx1* real vector.

$M(q)$: matrix of moment of inertia or inertia matrix, *nxn* real matrix.

$C(q,\dot{q})\dot{q}$: Coriolis, centrifugal and frictional forces. C is *nxn* real matrix.

$G(q)$: gravitational forces. G is an *nx1* real vector describing gravity.

$V(\dot{q})$: *nx1* real vector for viscous friction. It is neglected in our forthcoming treatment.

4.4 RA PID control

Proportional integral and derivative (PID) control is one of the simplest control schemes. It has been successfully used for the last six decades, or so, in many diversified applications of control. Despite its simplicity, PID is still active as an applied research field. In February 2006, a special issue of *IEEE Control Systems Magazine* has been devoted to the subject to account for its importance and actuality. Insofar as automatically-tuned PIDs (or autotuners) are concerned, commercial products became available around the early eighties. Since the Ziegler-Nichols rules of thumb developed in the 1940's, many attempts have been made in the "intelligent" choice of the three gains (*e.g.* Åström *et al.* 1992). The intelligent approach also helps in explanation of control actions usage. Indeed, in many real-life applications, explanation of control actions is desirable, *e.g.*, why derivative action is necessary. On the numerical level, the PID control *u(t)* is given by:

$$u(t) = K_p e(t) + K_v \dot{e}(t) + K_i \int_0^t e(n)dn \qquad (2)$$

$$e(t) = q(t) - q_d(t) \qquad (3)$$

$$\dot{e}(t) = \dot{q}(t) - \dot{q}_d(t) \qquad (4)$$

Equation (1) describes the control *u(t)*. K_p, K_i, K_v are the gains for the proportional (P), integral (I) and derivative (D) actions, respectively.

Equation (3) defines the position error *e(t)*, *i.e.*, the difference between the actual system position *q(t)* and the desired position *qd(t)*.

Equation (4) defines the velocity error and is simply the time-derivative of the error given in Equation (3) above. Equation (4) describes the difference between the actual system velocity and the desired velocity. The PID scheme block-diagram is given in Figure 3.

4.5 RA adaptive control
4.5.1 Purpose of adaptive control

The general adaptive controller design problem is as follows : given the desired trajectory *qd(t)*, with some (perhaps all) manipulator parameters being unknown, derive a control law for the actuator torques and an estimation law for the unknown parameters such that the

manipulator output *q(t)* tracks the desired trajectories after an initial adaptation process. Adaptive control laws may be classified on the basis of their control objective and the signal
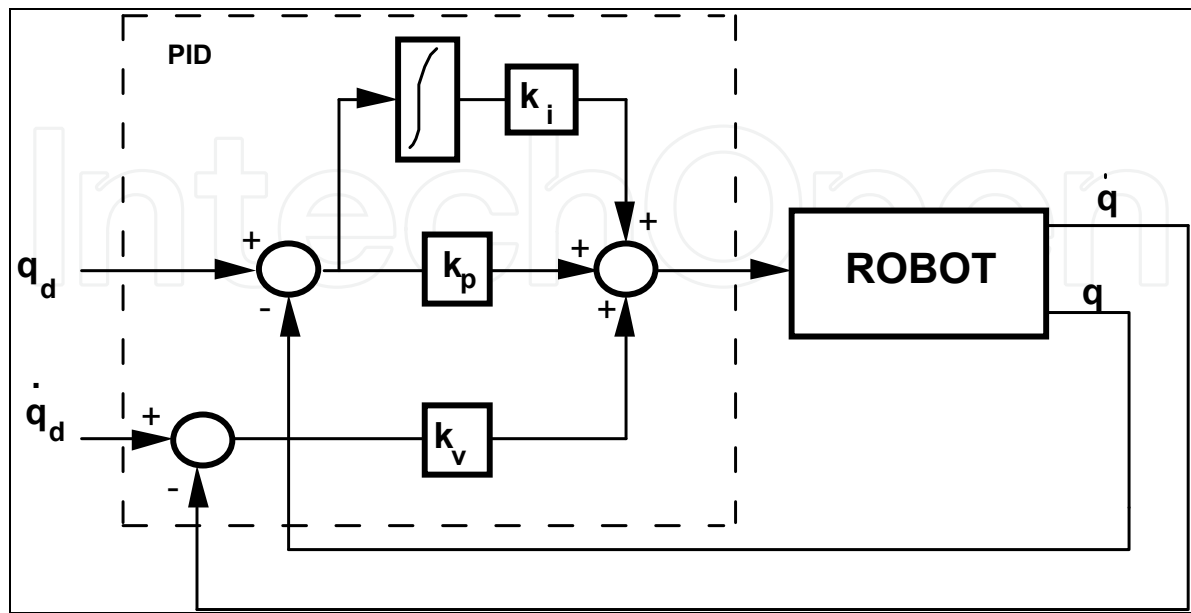

Fig. 3. RA PID Control

that drives the parameter update law. This latter can either be driven by the error signal between the estimated parameters and the true parameters (prediction or parametric error) or by the error signal between the desired and actual outputs (tracking error). Stability investigations are at the basis of acceptability of the proposed scheme.

4.5.2 Example of adaptive control scheme

As an example, the method due to (Amestegui *et al.*, 1987) compensates the modeling errors by a supplementary control $\delta\tau$. First, the computed torque approach is used whereby the linearizing control is obtained by a suitable choice of the torque. This amounts to simply replacing the acceleration $\ddot{q}$ by the control *u* in (1) above resulting in:

$$\tau(t) = M(q)u + C(q,\dot{q})\dot{q} + G(q) + V(\dot{q}) \tag{5}$$

Combining (1) and (5) yields:

$$M(q)(\ddot{q} - u) = 0 \tag{6}$$

Which amounts to *n* decoupled integrators $(\ddot{q} = u)$. In this case, the control *u* can be expressed in terms of the desired acceleration as a PD compensator.

Now compensate the modeling errors by a supplementary control $\delta\tau$ and neglect viscous friction.

$$\tau(t) = M_0(q)(u) + C_0(q,\dot{q})\dot{q} + G(q) + \delta\tau \tag{7}$$

Using the linear parametrization property, we obtain:

$$M_0(q)(u - \ddot{q}) + \delta\tau = \psi(q,\dot{q},\ddot{q})\Delta\theta \tag{8}$$

The compensating control is then given by :

$$\delta\tau = \psi(q,\dot{q},\ddot{q})\Delta\hat{\theta} \tag{9}$$

and the estimated parametric error vector is solution of :

$$\Delta\dot{\hat{\theta}} = -\Gamma\psi^T(q,\dot{q},\ddot{q})\hat{M}_0(q)(u - \ddot{q}) \tag{10}$$

In the previous equations, the following notations are used:
$\psi(q,\dot{q},\ddot{q})$ represents the regressor matrix, of appropriate dimensions.
The parametric error vector:

$$\Delta\theta = \theta_0 - \theta \tag{11}$$

where θ is the actual parameter vector and
θ_0 a constant and linear vector with respect to the nominal robot model.
$\Delta\hat{\theta}$ is the estimate of $\Delta\theta$ and

$$\Gamma = diag(\gamma_1,\gamma_2...,\gamma_n) \tag{12}$$

is a positive-definite diagonal matrix with $\gamma_i > 0$, representing the adaptation gain for the gradient parametric estimation method. Note that this last scheme avoids the inversion of the inertia matrix. It reduces the calculations complexity. However the measurement of the acceleration is always required. The block-diagram is given in Figure 4.
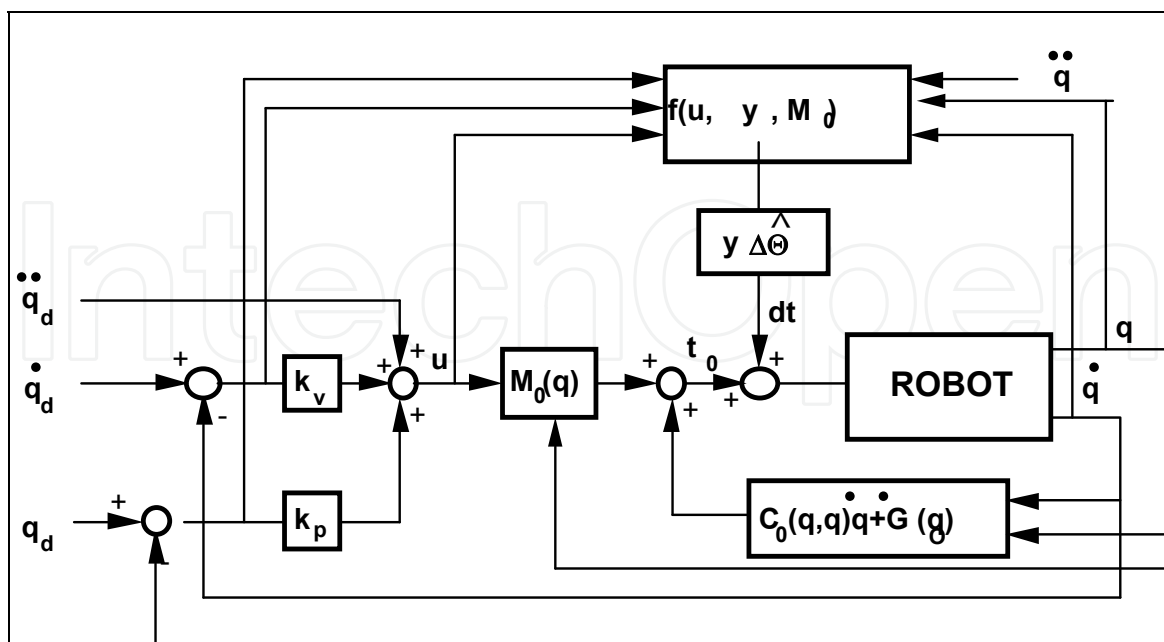
NB : In Figure 4, the following notations are used: $y = \psi$; $t = \tau$ $t_0 = \tau_0$

Fig. 4. Amestegui's adaptive compensation scheme

4.6 RA robust control

Robust control approach considers adding a correcting term to the control signal. This compensates the parametric error. This supplementary signal gives better tracking and makes the system more robust with respect to parametric error. We can classify the robust methods as Lyapunov-based methods, variable structure methods and non-chattering high gains methods.

4.6.1 Lyapunov-based methods

This class of methods is based on the Lyapunov direct method and is based on (Spong and Vidyasagar, 2006). The main problem encountered by Lyapunov-based class of RA control algorithms is the so-called *chattering effect* which results from commutation of the supplementary signal. This behavior creates control discontinuities. Research efforts have been accomplished that cater for this undesirable chattering effect. The algorithm proposed by (Cai and Goldenberg 1988) is a tentative answer to the problem of chattering. The issue of chattering represents a predilection area for the applicability of KB methods, since chattering can be modeled using human expertise.

4.6.2 Variable structure methods

Variable structure methods, such as the one proposed by (Slotine, 1985) are based on high-speed switching feedback control where the control law switches to different values according to some rule. This class of methods drives the nonlinear plant's trajectory onto an adequately designed sliding surface in the phase space independently of modeling errors. In (Chen and Papavassilopoulos, 1991) four position control laws have been analyzed and compared for a single-arm RA dynamics with bounded disturbances, unknown parameter, and unmodeled actuator dynamics. Although very robust to system's disturbance and simplifying the complexity of control laws implementation, these methods suffer from undesirable control chattering at high frequencies.

4.6.3 Non-chattering high gains methods

The non-chattering high gains class of methods is based on the singular perturbation theory and considers two time scales. This class avoids the chattering effect (Samson, 1987). However, robustness in this case is guaranteed by the choice of a nonlinear gain which is calculated from the *a priori* knowledge of the parametric uncertainties and from the model chosen for control calculation. The resulting control can be considered as a regulator which automatically adapts the gains in accordance with the displacement errors (Seraji, 1989) and uses high gains only when these are needed, for instance when displacement error is large.

4.6.4 Example of robust control scheme

In this case, the parameters are not known but their range of variations is known. The basic idea of this method is to add a compensating term to the control which is obtained from an *a priori* estimated model. This compensation term takes into account the parameters bounds and tries to compensate the difference between the estimated and the real parameters of the robot. This makes possible an improved trajectory tracking and provides robustness with respect to the parametric errors. Several schemes of RA robust control have been studied

and compared (Abdallah *et al.*, 1991). As an example, only one robust algorithm is described here, whose control law is given by :

$$\tau(t) = M_0(q)(u + \delta u) + C_0(q,\dot{q})\dot{q} + G_0(q) \tag{13}$$

where
* M_0, C_0 and G_0 are the *a priori* estimates of M, C and G, respectively.
* δu is the compensating control supplement.
* u is given by a PD compensator of the form:

$$u(t) = \ddot{q}_d(t) - K_p e(t) - K_v \dot{e}(t) \tag{14}$$

The additional control δu is chosen so as to ensure robustness of the control by compensating the parametric errors. Stability must be guaranteed. A reformulation of this control gives:

$$\dot{x} = Ax + B(\delta u + \eta(u,q,\dot{q})) \tag{15}$$

$$E_1 = Cx \tag{16}$$

where A, B, C and x are given by

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad C = \begin{bmatrix} \alpha & I \end{bmatrix} \quad x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{17}$$

with α is a diagonal constant positive-definite matrix of rank n, and

$$\eta(u,q,\dot{q}) = E(q)\delta u + E_1 u + M^{-1}(q)\Delta H(q,\dot{q}) \tag{18}$$

$$E(q) = M^{-1}(q)M_0(q) - I \tag{19}$$

$$\Delta H(q,\dot{q}) = [C_0(q,\dot{q}) - C(q,\dot{q})]\dot{q} + [G_0(q) - G] \tag{20}$$

Stability is granted only if the vector $\eta(u,q,\dot{q})$ is bounded. These bounds are estimated on the worst-case basis. Furthermore, under the assumption that there exists a function ρ such that:

$$\|\delta u\| < \rho(e,\dot{e},t) \tag{21}$$

$$\|\eta\| \leq \rho(e,\dot{e},t) \tag{22}$$

the compensating control δu can be obtained from :

$$\delta u = \begin{cases} -\rho(e,\dot{e},t)\dfrac{E_1}{\|E_1\|} & if \quad \|E_1\| \neq 0 \\ 0 \quad if & \|E_1\| = 0 \end{cases} \tag{23}$$
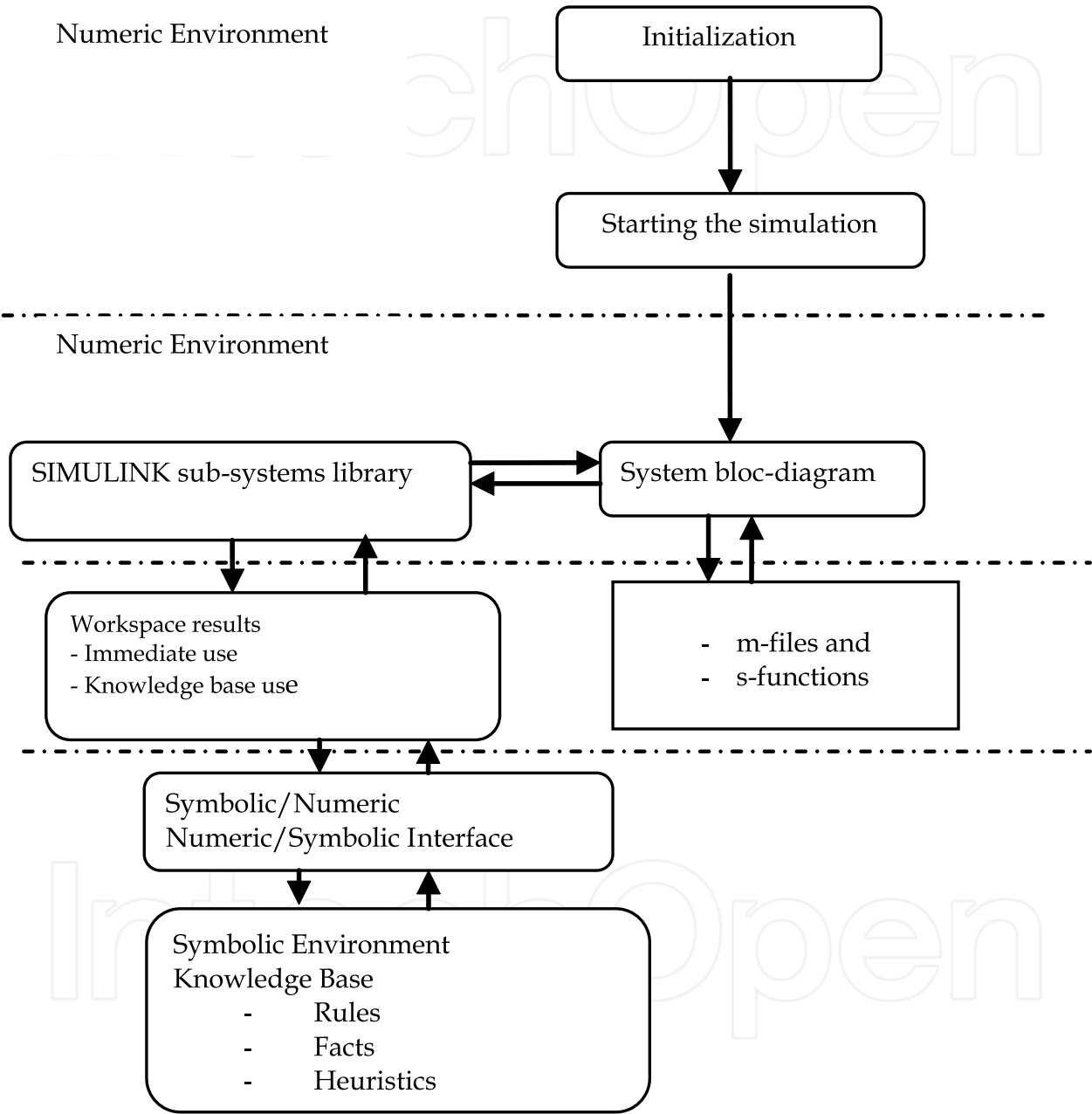
This last control δu presents a chattering effect due to the discontinuities in (23). This phenomenon can cause unwanted sustained oscillations. Another control has been proposed which reduces these unwanted control jumps, (Cai and Goldenberg, 1988) as given in equation (24).

$$\delta u = \begin{cases} -\rho(e,\dot{e},t)\dfrac{E_1}{\|E\|} & if & \|E_1\| > \varepsilon \\[2em] \dfrac{-\rho(e,\dot{e},t)}{\varepsilon}E_1 & if & \|E_1\| \leq \varepsilon \end{cases} \tag{24}$$

The robust control scheme is represented in Figure 5.

Fig. 5. Spong and Vidyasagar's robust control algorithm

5. Implementation

5.1 Basic architecture

The basic architecture is described in Figure 6. The general menus are described in Figure 7. The main program is started from the Matlab™ workspace window. Simulation triggers the Simulink™ environment and results can be obtained under the Matlab™ graphics window or in the Simulink™ environment (*e.g.* through scopes). Results can also be stored in *.MAT data files to be later handled by the knowledge base, through the interface.

The overall system is written in the Matlab™/ Simulink™ environment (http://www.mathworks.com). One of the main reasons for this choice is the possibility of interfacing it with the developed knowledge base using higher programming language, such as Microsoft Visual C++™ (MVC++™), under Windows™. The knowledge base is developed under a commercial expert system generator that supports interfacing with external MVC++™ executable programs. The other fundamental reason is the Matlab™ control systems library functions and specialized toolboxes, *e.g.* control systems toolbox and identification toolbox needed for adaptive control. Although, many languages / environments can be identified as suitable for the solution to our RA problem, we do not know, however whether any of these is interfaceable with the chosen expert system generator.

Numeric Environment

Initialization

Starting the simulation

Numeric Environment

SIMULINK sub-systems library ⟷ System bloc-diagram

Workspace results
- Immediate use
- Knowledge base use

- m-files and
- s-functions

Symbolic/Numeric
Numeric/Symbolic Interface

Symbolic Environment
Knowledge Base
- Rules
- Facts
- Heuristics

Fig. 6. Implemented Architecture

Initialization

Menu 1: Type of Robot
    1-   SCARA
    2-   PLANAR
    3-   RZERO
    4-   Close

(1, 2, 3)

4

End

Robot Parameters Introduction

7

Menu 2: Control Choice
    1-   PID
    2-   Known Parameter Case
              2.1 Computed Torque
              2.2 Passivity
    3-   Unknown Parameter Case
              3.1 Adaptive Computed Torque
              3.2 Passive adaptive
    4-   Robust Control
    5-   Block Diagram
    6-   Curve Plotting
    7-   Close

(1, 2, 3, 4)

5

6

Control Parameters
Introduction

System s-blocks

Menu 3: Curve
    1-   Positions + errors
    2-   Velocities + errors
    3-   Control law
    4-   Adaptive Parameters
    5-   Close

SIMULATION

Calculation of response
time and responses
errors

5

(1,2,3,4)

Curves Plotting

Saving in .MAT files

Fig. 7. Exploitation Environment

5.2 Plant configuration

Some of the available RA configuration have been used in the implementation, as examples. PLANAR, SCARA and RZERO are chosen because they are widely used and they represent different classes of configurations, as described in Figure 8,9,10 below. Of course the system is open to other configurations through the Matlab™ environment.
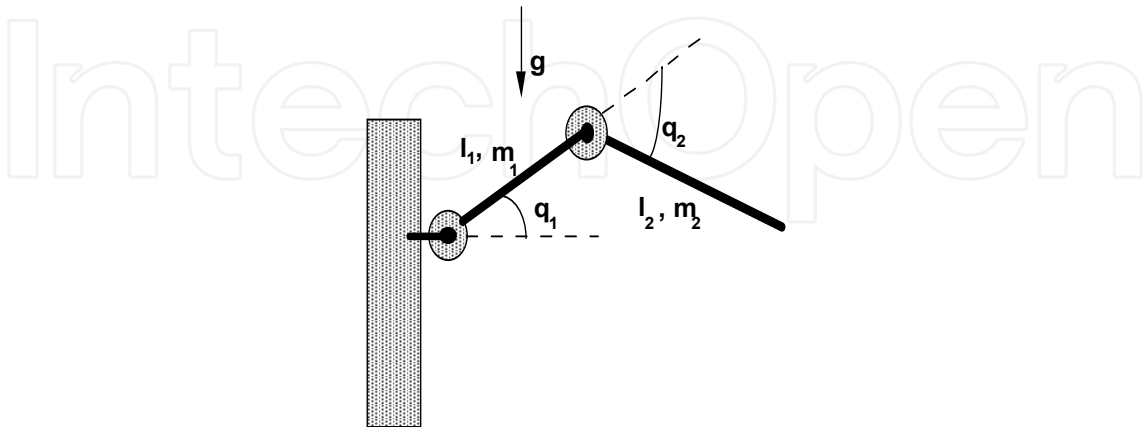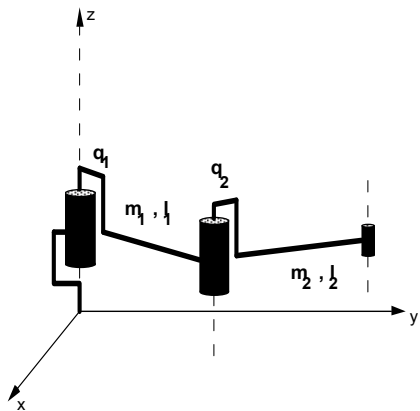

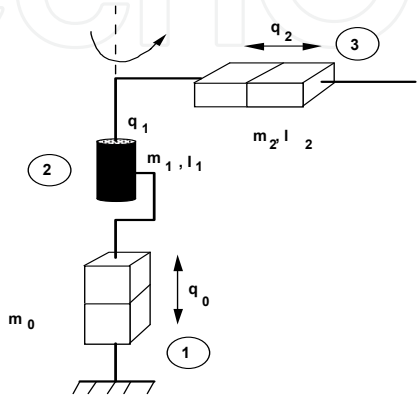Fig. 8. PLANAR Robot

Fig. 9. SCARA Robot

Fig. 10. RZERO Robot

5.3 Knowledge organization: from worlds to objects

Knowledge organization is handled by the inferential (exploration) environment. The knowledge is organized in different levels :

5.3.1 Global *vs.* local search

i.   *At the global level* : this is done through the partitioning of the KB in coherent thematic sets of rules (each of these sets is called a *world*). These worlds can be hierarchically organized offering the possibility of describing global knowledge (ascending worlds, fathers worlds) and local knowledge (descending worlds, descendants worlds).

ii. *At the local level* : this is done through the expertise structuring using a network of classes and objects. A class is defined as an abstract object maker. Objects represent a declarative knowledge described by sets of particular data (called attributes) and the corresponding attributes values. Rules allow description of the expert knowledge using objects and / or classes. They are expressed in the conventional IF-THEN form. We only operate the KB as a stand-alone module to test its behaviour against that of human experts for further refinement.

5.3.2 The existing worlds

Worlds are coherent sets of rules and represent independent and encapsulated entities ensuring a high degree of knowledge modularity and maintenance. A world can be created according to the type of knowledge that is handled (*e.g.* set of rules dealing with PID controller). Hierarchical representation is available. This allows the organization of knowledge from the more general to the more specific (top-down fashion).

i.   *The meta-level nucleus (MLN)*

The meta-level nucleus (MLN) represents the world that governs the navigation from one world to the other (or from one individual KB to the other). It is placed at the highest level of the hierarchy. All remaining worlds are sub-worlds of the MLN. All the rules of the MLN world (called a father world) are applicable in all the other inheriting sub-worlds (called descendants). Its description is given by the following structure:

```
World name : MLN
Father World : None
Descendants Worlds : %List of all the other remaining worlds%
```

ii. *Pruning worlds*

Based on the data and specifications provided by the user, the pruning worlds help in guiding the search towards the specialized individual knowledge base (IKB) as soon as possible. This pruning process is efficient in concentrating on the specific knowledge of interest at this or that particular step of the reasoning process. Once these worlds are selected, they become active while all other worlds remain inactive. The pruning worlds consist of the following :

```
W1 World : DynamicModelKnown
          % Describing the rules applicable for this case%
     W11 Sub-World : SpeedSlow
                    W111 Sub-sub-World : ParametersKnown
                    W112 Sub-sub-World : ParametersUnknown
                    W113 Sub-sub-World : ParametersOthers
                    W114 Sub-sub-World :ParametersNoImportance
     W12 Sub-World : SpeedHigh
```

```
        (Same sub-worlds as in W11 above)
        W13 Sub-World : SpeedUnknown
        (Same sub-worlds as in W11 above)
        W14 Sub-World : SpeedOthers
        (Same sub-worlds as in W11 above)
        W15 Sub-World : SpeedNoImportance
        (Same sub-worlds as in W11 above)


W2 World : DynamicModelUnknown
            % Describing the rules applicable for this case%

        W21 Sub-World : SpeedSlow
                        W211 Sub-sub-World : ParametersUnknown
                        W212 Sub-sub-World : ParametersOthers
                        W213 Sub-sub-World :ParametersNoImportance
        W22 Sub-World : SpeedHigh
        (Same sub-worlds as in W21 above)
        W23 Sub-World : SpeedUnknown
        (Same sub-worlds as in W21 above)
        W14 Sub-World : SpeedOthers
        (Same sub-worlds as in W21 above)
        W24 Sub-World : SpeedNoImportance
        (Same sub-worlds as in W21 above)


W3 World : DynamicModelOthers
            % Describing the rules applicable for this case%
        W31 Sub-World : SpeedSlow
                        W311 Sub-sub-world : ParametersKnown
                        W312 Sub-sub-World : ParametersUnknown
                        W313 Sub-sub-World : ParametersOthers
                        W314 Sub-sub-World :ParametersNoImportance


        W32 Sub-World : SpeedSlow
        (Same sub-worlds as in W31 above)
        W33 Sub-World : SpeedUnknown
        (Same sub-worlds as in W31 above)
        W14 Sub-World : SpeedOthers
        (Same sub-worlds as in W31 above)
        W34 Sub-World : SpeedNoImportance
        (Same sub-worlds as in W31 above)


W4 World : DynamicModelNoImportance
            % Describing the rules applicable for this case%
        W41 Sub-World : SpeedSlow
                        W411 Sub-sub-World : ParametersKnown
                        W412 Sub-sub-World : ParametersUnknown
                        W413 Sub-sub-World : ParametersOthers
                        W414 Sub-sub-World :ParametersNoImportance
        W42 Sub-World : SpeedSlow
        (Same sub-worlds as in W41 above)
        W43 Sub-World : SpeedUnknown
        (Same sub-worlds as in W41 above)
        W14 Sub-World : SpeedOthers
        (Same sub-worlds as in W41 above)
        W44 Sub-World : SpeedNoImportance
        (Same sub-worlds as in W41 above)
```

For each sub-world *Wij* (*i* = 1 to 4 ; *j* = 1 to 4), there correspond a sub-sub-world for RA parametric description. These pruning worlds give a preliminary guide to a world (corresponding to the chosen algorithm) where the initial search is to be started. If the results given by this algorithm are satisfactory then choose this algorithm as a solution. Otherwise, either fine-tune the obtained solution within the same world (or other eventual specialized sub-worlds) or go back to the meta-level nucleus (MLN) for further search.

iii.   The worlds describing the RA algorithms

For each RA algorithm, we have developed a world. Each of these worlds can be considered as an independent KB (IKB). Some of the worlds have very few rules. Each IKB can obviously be incremented, provided the expertise is available. We have considered worlds and sub-worlds partially describing the following algorithms.

```
World PID
      Sub-worlds : Basic PID, Gravitational PID, Adaptive PID, Robust PID.
World Computed Torque (known parameters)
      Sub-worlds : PD control, Predictive control
World Compensators
      Sub-worlds  :  Spong's  adaptive  compensator,  Amestegui's  adaptive
      compensator
World Adaptive Control
      Sub-worlds : linearized adaptive, passive adaptive
World Robust Control
      Sub-worlds : Robust PID, large gains, variable structure control (VSC)
```

5.4 Example : Fuzzy rule involving fuzzy attributes in its conclusion.

If the user does not know the RA parameters but knows the dynamic model and that the RA is slow, then a tentative algorithm is the passive adaptive or the linear adaptive. In the conclusion, we can therefore translate this by a certainty factor (CF) of 50 meaning that either algorithm can be used with a degree of equal certainty. The CF can of course be changed according to the available knowledge and refined expertise. This rule can be expressed by :

```
WORLD : MLN % New world %
DESCENDANTS WORLDS % Here is a list of all worlds %

Rule TryPassvAdaptCF60  % This is the name of the rule %
CHAINING : forward
PRIORITY : 40  % can be changed from 0 to 100 %
CONTENT
      IF Guide.DynamicModelKnown_VelocitySlow = TRUE
      AND RA.Parameters = "don't know"
      AND Algorithm.AlgoActivation = "Activable"
      THEN TryAlgorithm.PassivAdaptivFuzzy = TRUE CF 50
      AND Guide.PassivAdaptivCF50 = TRUE
```

Other situations can be described in a similar manner.

6. Conclusion

We have described some foundational steps to solve the RA control using knowledge base systems approach. More specifically, this research work reports some features of KBC

approach as applied to some RA control algorithms spanning PID through adaptive, and robust control. As such, this research represents an early contribution towards an objective evaluation of the effectiveness of KBC as applied to RA control. A unification of the diversified works dealing with RAs, while concentrating on KBC as an alternative control method, is therefore made possible. The adopted knowledge base systems approach is known for its flexibility and conveys a solution better than that provided by numerical means alone since it incorporates codified human expertise on top of the algorithms. The fundamental constraints of the proposed method is that it requires an elicitation of human expertise or extensive off-line trials to construct this expertise. This expertise codification has a direct impact on the size of the KB and on the rapidity of the user-defined problem solution. Like any KBS method, the proposed procedure also requires a diversified coverage of the working domain during the elicitation stage to obtain a richer KB. As a consequence, the results report only some aspects of the overall issue, since these describe only a fragment of the human expertise for a small class of control algorithms. Much work is still required on both sides, *i.e.,* robotics and KBS in order to further integrate these two entities within a single one while meeting the challenges of efficient real-life applications.

7. References

Abdallah, C.; Dawson, D.; Dorato, P. & Jamshidi, M. (1991). Survey of robust control for rigid robots, *IEEE Control Systems Magazine*, Vol. 11, No. 2 (February 1991) page 24–30, ISSN: 0272-1708.

Amestegui, M.; Ortega, R. & Ibarra, J.M. (1987). Adaptive linearizing and decoupling robot control : a comparative study of different parametrizations, *Proceedings of 5th Yale Workshop on Applications of Adaptive Systems Theory*, 1987, New Haven, CN, USA.

Åström, K.J. (1989). Towards intelligent control, *IEEE Control Systems Magazine*, Vol. 9, no.3, pp. 60-64, ISSN: 0272-1708.

Åström, K.J. & McAvoy, T.J. (1992). Intelligent control. *J. Process Control,* Vol. 2, no. 3, pp. 115-127, ISSN: 0959-1524.

Åström, K.J.; Hang, C.C.; Persson P. & Ho W.K. (1992). Towards intelligent PID control, *Automatica*, Vol. 28, No. 1 (January 1992) page 1-9, ISSN: 0005-1098.

Cai, L. & Goldenberg, A.A. (1988). Robust control of unconstrained maneuvers and collision for a robot manipulator with bounded parameter uncertainty. *Proceedings IEEE Conference on Robotics and Automation,* pp. 1010–1015, Vo. 2, 1988, Philadelphia, IEEE, NJ, USA.

Calangiu, G.A.; Stoica, M. & Sisak, F. A knowledge based system designed for making task execution more efficient for a robot arm. *IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD),* pp. 381-387, 24-26 June 2010, Budapest, Hungary.

Cellier, F.E.; Schooley, L.C.; Sundareshan, M.K. & Ziegler, B.P. (1992). Computer aided design of intelligent controllers : challenges of the nineties. In M.Jamshidi and C.J Herget (Eds.). *Recent Advances in Computer-Aided Control Systems Engineering.* Elsevier, Science Publishers, Amsterdam, Holland.

Chen, L.W. & Papavassilopoulos, G.P. (1991). Robust variable structure and adaptive control of single-arm dynamics. *Proceedings 30th Conference on Decision and Control,* pp. 367-372, Brighton, UK, 11–13 December 1991, IEEE, NJ, USA.

Corke, P.I. (1996). A robotics toolbox for MATLAB, *IEEE Robotics and Automation Magazine*, Vol. 3, No. (March 1996) page 24-32, ISSN: 1070-9932.

Craig, J.J. (2005). *Introduction to Robotics: Mechanics and Control* , 3rd Ed., Pearson Prentice Hall, ISBN: 0201-54361-3, Upper Saddle River, NJ, USA.

Craig, J.J. (1988). *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, ISBN: -201-10490-3, Reading, MA, USA.

Craig, J.J., Hsu, P. & Sastry, S. (1987). Adaptive control of mechanical manipulators. *International Journal of Robotics Research,* Vol. 6, No. 2 (June 1987) page 16-28, ISSN: 0278-3649.

Egerstedt, M.; Frazzoli, E. & Pappas, G. (2006). Special section on symbolic methods for complex control systems, *IEEE Transactions On Automatic Control.* Vol. 51, No. 6 (June 2006) page 921-923, ISSN: 0018-9286.

Eldeib, H.K. & Tsai S. (1989). Applications of symbolic manipulation in control system analysis and design. *Proceedings of the IEEE Symposium on Intelligent Control* page 269-274, 1989, Albany, NY, USA.

Etxebarria, V. (1994). Animation of a simple planar robotic arm. *Proceedings of the European Simulation Multiconference (ESM'94)*, pp. 809-813, Barcelona, Spain, 1-3 June 1994.

Hamdi-Cherif, A. Towards robotic manipulator grammatical control, Invited Book Chapter In: Suraiya Jabin (Ed.) *"Robot Learning"*, SCIYO Pub., pp. 117-136, October 2010, ISBN: 978-953-307-104-6.

Hamdi-Cherif, A. & Kara-Mohammed (*alias* Hamdi-Cherif), C. (2009). Grammatical inference methodology for control systems. *WSEAS Transactions on Computers,* Vol. 8, No. 4 (April 2009) page 610-619, ISSN: 1109-2750.

Handelman, D.A.; Lane, S. H. & Gelfand, J. (1990). Integrating neural networks and knowledge-based systems for intelligent robotic control, *IEEE Control Systems Magazine*, Vol. 10, no.3, pp. 77-87, ISSN: 0272-1708.

Hanlei, W. (2010). On the recursive implementation of adaptive control for robot manipulators, *29th Chinese Control Conference (CCC10)*, pp. 2154 – 2161, 29-31 Jul. 2010, Beijing, China.

Hsia, T.C. (1986). Adaptive control of robot manipulators: a review. *IEEE International Conference on Robotics and Automation,* San Fransisco, CA, USA, 1986, IEEE, NJ, USA.

Johansson, R. (1990). Adaptive control of robot manipulator motion. *IEEE Transactions on Robotics and Automation*, Vol. 6 No. 4 (August 1990) pp. 483-490, ISSN: 1042-296X.

Klavins, E. (2007). Programmable self-assembly. *IEEE Control Systems Magazine.* Vol. 27, No. 4 (August 2007) page 43-56. ISSN: 0272-1708.

Kwan, C.; Dawson, D.M. & Lewis, F.L. (2001). Robust adaptive control of robots using neural network: global stability. *Asian Journal of Control*, Vol. 3, No. 2 (June 2001) page 111-121, ISSN: 1561-8625.

Landau, I.D. & Horowitz R. (1989). Applications of the passive systems approach to the stability analysis of the adaptive controllers for robot manipulators. *International Journal of Adaptive Control and Signal Processing*, Vol. 3, No. 1 (January 1989) pp. 23-38, ISSN: 0890-6327.

Lewis, F.L.; Dawson, D.M. & Abdallah, C.T. (2003). *Robot Manipulator Control: Theory and Practice, Control Engineering Series,* 2nd Ed., CRC Press, Taylor & Francis Group, ISBN: 978-0824740726, New York, USA.

Martins, J. F.; Dente, J.A.; Pires, A.J. & Vilela Mendes, R. (2001). Language identification of controlled systems: modeling, control, and anomaly detection. *IEEE Transactions On Systems Man and Cybernetics– Part C: Applications And Reviews* Vol. 31, No. 2 (April 2001) page 234-242, ISSN: 1094-6977.

Merchán-Cruz, E.A. & Morris, A.S. (2006). Fuzzy-GA-based trajectory planner for robot manipulators sharing a common workspace, *IEEE Transactions on Robotics*, Vol. 22, No. 4 (August 2006) page 613-624, ISSN: 1042-296X.

Meystel, A. (1991). Multiresolutional feedforward / feedback loops. *Proceedings of the IEEE International Symposium on Intelligent Control*, 13-15 August 1991, pp. 85-90, Arlington, Virginia, USA.

Meystel, A. (1994). Multiscale models and controllers. *Proceedings of the IEEE / IFAC Joint Symposium on Computer-Aided Control Systems Design*, Tucson, Arizona, 7-9 March 1994, pp. 13-26.

Ortega, R. & Spong, M.W. (1989). Adaptive motion control of rigid robots: A tutorial, *Automatica*, Vol. 25, No. 6 (November 1989) page 877–888, ISSN: 0005-1098.

Polyakov, V.; Ghanadan R. & Blackenship G.L. (1994). Symbolic numerical computation tools for nonlinear and adaptive control. *Proceedings of IEEE-IFAC Joint Symposium on CACSD*, pp. 117-122, Tucson, AZ, USA, 7-9 March 1994, IEEE, NJ, USA.

Popov, V.M. (1973). *Hyperstability of Control Systems.* Springer Verlag, ISBN: 0387063730, Berlin, Germany.

Quigley, M.; Gerkey B.; Conley, K.; Faust, J.; Foote, T., Leibs, J.; Berger, E.; Wheeler, R. & Ng, A. (2009). ROS: an open-source Robot Operating System. http://www.cs.stanford.edu/people/ang/papers/icraoss09-ROS.pdf

Rao, M. (1992) *Integrated System for Intelligent Control. LNCIS*, Vol. 167, Springer-Verlag, New York, USA.

Siciliano, B.; Sciavicco, L.; Villani, L. & Oriolo, G. (2009). *Robotics Modeling, Planning and Control.* Springer, ISBN 978-1-84628-641-4, e-ISBN 978-1-84628-642-1, Series published under ISSN 1439-2232, London, UK.

Samson, C. (1987). Robust control of a class of nonlinear systems and applications to robotics. *International Journal of Adaptive Control and Signal Processing*, Vol. 1, No. 1 (January 1987) page 49-68, ISSN: 0890-6327.

Saridis, G.N. (1987). Machine-intelligent robots : A hierarchical control approach. In T. Jorjanides and B. Torby (Eds.) *Expert Systems and Robotics.* NATO Series F, pp. 221-234.

Seraji, H. (1989). Decentralized adaptive control of manipulators: theory, simulation and experimentation. *IEEE Transactions on Robotics and Automation*, Vol. 5 No. 2 (April 1989) Page 183-201, ISSN: 1042-296X.

Slotine, J.J.E. (1985). The robust control of robot manipulators. *International Journal of Robotics Research.* Vol. 4, No. 2 (June 1985) page 465-492, ISSN: 0278-3649.

Slotine, J.J.E. & W. Li (1987). On the adaptive control of robot manipulators. *International Journal of Robotics Research,* Vol. 6, No. 3 (September 1987) page 49-59, ISSN: 0278-3649.

Spong, M.W.; Hutchinson, S. & Vidyasagar M. (2006). *Robot Modeling and Control*, Wiley, ISBN: 0471649902, New York, USA.

Vukabratovic, M.; Stoic D. & Kirchanski, N. (1984). Towards non-adaptive and adaptive control of manipulation robots. *IEEE Transactions on Automatic Control.* Vol. 29, No.9 (September 1984) page 841-844, ISSN: 0018-9286.

Yae, K.H.l; Lin, T.C. & Lin S.T. (1994). Constrained multibody library within EASY5. *Simulation*, Vol. 62, No. 5 (May 1994) page 329-336, ISSN (Online): 1741-3133, ISSN (Print): 0037-5497.

Robot Arms

Edited by Prof. Satoru Goto

Robot arms have been developing since 1960's, and those are widely used in industrial factories such as welding, painting, assembly, transportation, etc. Nowadays, the robot arms are indispensable for automation of factories. Moreover, applications of the robot arms are not limited to the industrial factory but expanded to living space or outer space. The robot arm is an integrated technology, and its technological elements are actuators, sensors, mechanism, control and system, etc.

How to reference

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

Aboubekeur Hamdi-Cherif (2011). Knowledge-Based Control for Robot Arm, Robot Arms, Prof. Satoru Goto (Ed.), ISBN: 978-953-307-160-2, InTech, Available from: http://www.intechopen.com/books/robot-arms/knowledge-based-control-for-robot-arm

INTECH
open science | open minds