7

Towards Robotic Manipulator Grammatical Control

Aboubekeur Hamdi-Cherif[1,2]
[1]Qassim University, Computer College, PO Box 6688 – 51452 Buraydah
[2]Permanent Address: Université Ferhat Abbas Setif (UFAS)
Faculty of Engineering, 19000 Setif
[1]Saudi Arabia
[2]Algeria

1. Introduction

The present research work reports the effectiveness and appropriateness of grammatical inference (GI) as an alternative control method. Specificity is made on robotic manipulator (RM). RM 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 RM industrial applications. RM control is a much diversified field. As a result, it makes concentrated research a difficult task. While RM 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 other methods such as those provided by GI. Indeed, the GI efforts as applied to control at large remain quite isolated (Martins *et al.* 2006). Our fundamental aim is to contribute to the integration of GI within RM control, considered within a larger control methodology (Hamdi-Cherif, 2009). As a subfield of machine learning, GI attempts to learn structural models, such as grammars, from diverse data patterns, such as speech, artificial and natural languages, amongst others. GI broadest aim is therefore consistent with the overall goal of machine learning defined as a computational methodology that provides automatic means of improving tasks from experience. As a general computational method, GI is the process whereby a language is automatically generated from positive and eventually negative examples of sentences. Inductive inference of formal languages, as defined by (Gold, 1967), have been studied in the case of positive data, *i.e.,* when the examples of a given formal language are successive elements of some arbitrary enumeration of the elements of the language. After Gold's negative result, a theorem due to (Angluin, 1980) characterizes when an indexed family of nonempty recursive formal languages is inferrable from positive data. In oracle-based GI, the inferred language is done with the help of a teacher who answers questions concerning the language to be inferred. Useful accounts of GI methods and algorithms can be found in (Sakakibara, 1996; Cicchello & Kremer, 2003). Grammar-based classifier system as a universal tool for grammatical inference has been studied by (Unold, 2008). For the specificity of the present work, GI is understood as the learning of the behavior of a dynamical system, namely an RM. This behavior is translated into the syntax of a language to be learned. 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 (such as computed torque, adaptive and robust methods), exemplified by RM control, while concentrating on GI as an alternative control method. We describe the epistemological characteristics of a framework that is believed to integrate two distinct methodological fields of research (Klavins *et al.* 2006), *i.e.*, theory of formal languages and machine learning where GI is rooted, on the one hand, and control theory, from where RM control originated, on the other hand. Blending research from both fields results in the appearance of a richer community. Emphasis is now made on RM 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 problem is formulated within the general area of symbolic control. Section 3 briefly describes related works with three different lines of research - conventional control, GI in control, and software systems. Section 4 summarizes RM control in standard mathematical terms. Section 5 deals with GI as an alternative control method. Results are summarized in Section 6 followed by a conclusion.

2. Problem formulation

2.1 From machine learning to GI

The objective of machine learning is to produce general hypotheses by induction, which will make predictions about future instances. The externally supplied instances are usually referred to as training set. Generally speaking, machine learning explores algorithms that reason from externally supplied instances, also called inputs, examples, data, observations, or patterns, according to the community where these appear (Mitchell, 1997). To induce a hypothesis from a given training set, a learning system needs to make assumptions, or biases, about the hypothesis to be learned. A learning system without any assumption cannot generate a useful hypothesis since the number of hypotheses that are consistent with the training set is practically infinite and many of these are totally irrelevant. Because GI is considered as a sub-field of machine learning, it therefore inherits this fundamental characteristic (de la Higuera, 2005).

2.2 GI formalism

2.2.1 Formal grammar

A *formal string grammar* or simply *grammar* G has four components (Cohen, 1991):

- A set of symbols N called non-terminals.
- A set of symbols T, called terminals with the restriction that T and N are disjoint.
- A special non-terminal symbol S, called a start symbol.
- A set of production rules P.

In other words, a formal grammar is a set of rules that tells whether a string of characters (*e.g.* a sentence in a natural language), constructed from the starting symbol, can be expressed in the form of terminals (words), *i.e.,* in the general accepted structure of a sentence.

2.2.2 Inference

Inference or inductive learning is a generalization process which attempts to identify a hidden function, given a set of its values. As mentioned above, in formal languages settings, learning the syntax of a language is usually referred to as *grammatical induction* or *grammar inference* (GI); an important domain for both cognitive and psycholinguistic domain as well as science and engineering. We are concerned with the problem of constructing a grammar from some given data. These latter, whether sequential or structured, are composed from a finite alphabet, and may have unbounded string-lengths. In relatively simple situations, induction considers a deterministic finite-state automaton, or DFA, that takes strings of symbols as input, and produces a binary output, indicating whether that string, or sentence, is a part of the DFA's encoded language. In this particular example, GI builds a model of the hidden DFA internal structure, based only on pairs of input sentences and classifications, or outputs. From a control theory point of view, this is a classical input-output *identification* problem. The most successful GI algorithms produced so far are heuristic in nature. For our implementation example, we will concentrate only on one of these algorithms, namely *ILSGInf* (Hamdi-Cherif, 2007). An overview of some out of many other algorithms can be found in (de la Higuera, 2005).

2.3 Motivation for grammatical control approach

Any grammar codes for the class of all possible syntactical patterns that belong to the language produced by the grammar. The basic idea is to design a parser (or classifier) that recognizes strings accepted by the grammar. There is a mapping signals-to-strings. Each signal is quantized and each value is given a terminal symbol. Under normal operations, signals are compatible with the grammar. Once the grammar is learnt, it is used as a reference by the nominal system. If at a later time, there is some faulty output from the dynamical system then the faulty generated signals are translated as "odd" strings, reporting anomaly detection. The basic procedure is described in Figure 1. An input of non-terminals is used for both the nominal and actual dynamical systems. An error is evaluated between the strings generated by both systems. Two modes are possible. In the open-loop mode, the grammar generates the working patterns imposed by the external input command. If this error exceeds some threshold, a fault is reported. A closed-loop control is used when the control U is generated for an output y to be within some prescribed values.

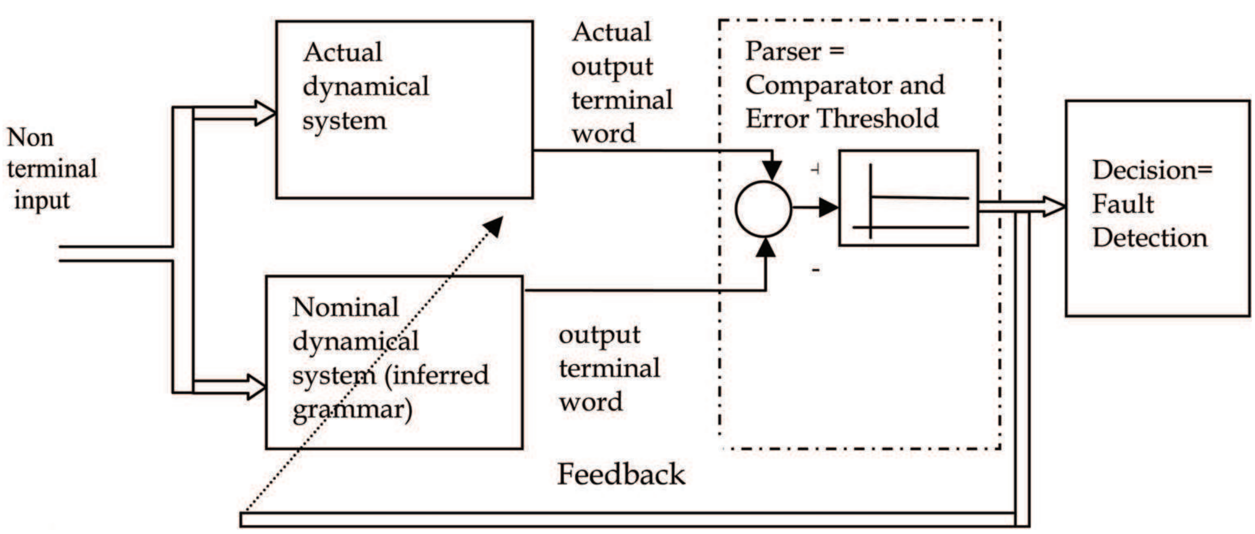


Fig. 1. Grammatical control used in open-loop/closed-loop modes

2.4 From hybrid control to grammatical control

The closest application field of GI-oriented control is the area of symbolic control, developed within the larger area of hybrid control. In the early sixties, the discipline of hybrid control referred to controlled systems using both discrete and continuous components. This discipline spanned a substantial area of research from basic switched linear systems to full-scale hybrid automata. In short, symbolic control methods 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. However, a new line of research in hybrid systems has also 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. For our concern, we will consider symbolic control as an integration of pure control and formal language theory. As a result, symbolic control addresses questions at the highest level, *i.e.*, at the level of symbols, and as such is as close to computer science and discrete mathematics as much as to classic control theory. At the same time, symbolic control provides faithful descriptions of the continuous level performance of the actual system, and as such, provides a formal bridge between its continuous and the discrete characteristics (Egerstedt *et al.,* 2006).

2.5 GI in self-assembly of robotic systems

The second motivation for using GI as a robot control method is its applicability to self-assembly. It is easy to see that assembling shapes into a given pattern can be seen as a "language" where the elementary shapes are the "words" and the obtained pattern correspond to a "sentence" or "string" obeying some specific rules or "grammar" for generating grammatically correct sentences. The process of self-assembly can therefore be seen as the automatic generation of a language. Since assembling geometrical shapes into some desired shape can be viewed as a set of sentences of a language, it is therefore not surprising to address this issue from the standpoint of grammars. Specifically, graph grammars are used as an emerging field that is believed to be promising in self-assembly (Klavins, 2007). One of the strigent questions in robotic self-organized systems is to know whether it is possible to synthesize a set of local controllers that produce a prescribed global behavior that is sufficiently robust to uncertainties about the environmental conditions (Hamdi-Cherif, 2009).

3. Related works

Related works are described under three different lines of research, namely pure control, GI approach to control, and software applications.

3.1 Conventional RM control

On the control side, we concentrate on some classes of control methods such as adaptive control and passivity-based control. From the vast literature on adaptive control, only a small portion is applicable to RM 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 (*e.g.* Craig *et al.*, 1986). 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 & 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).

3.2 Grammatical control

GI as applied to robot control at large is relatively a new area of research. As an indication, a rapid search in *IEEE* site (http://www.ieee.org) using *ieeexplore* search engines and keywords (formal language control + dynamical systems + grammatical inference) hits one journal paper and two conferences papers, all by the same team of authors (Martins, 2006). Moreover, the majority of other results deals with control systems in general and none with RM control. The closest works relied on graph grammars. For instance, in (Hasemann, 1994), new concepts for robot control architectures are presented. The key techniques used to model decision making and plan modification are fuzzy logic and graph grammars. Fuzzy logic is used to guide planning and graph grammars provide the framework for expanding plan components. Special emphasis is put on planning and monitoring for task level control. New features introduced are behavior switching, complete monitoring of plan execution and plan validity and rigorous explicit representation of activities and mutual dependencies within plans. Moreover, a behavior switching mechanism is proposed, which allows critical behaviors to interrupt or abandon a current less critical behavior. Graphs grammars have alternatively been used in self-assembly (*e.g.* Klavins, 2007). In (Burbidge *et al.*, 2009), an autonomous mobile robot requires an onboard controller that allows it to perform its tasks for long periods in isolation. Grammatical evolution is a recent evolutionary algorithm that has been applied to various problems, particularly those for which genetic programming has been successful. Indeed, evolutionary techniques such as genetic programming offer the possibility of automatically programming the controller based on the robot's experience of its environment. A method for applying grammatical evolution to autonomous robot control has been presented and evaluated in simulation for the Khepera robot.

3.3 Robot control: numeric and symbolic software

Few authors have addressed the issue of designing and developing software systems that cater for general-purpose RM 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 RM 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). Other environments tackled the general control problem from a CASCD standpoint (Hamdi-Cherif, 1994). Recent research directions aim at the development of operating systems for robots, not necessarily RM. 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 heterogenous 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 grammatical control approach to solve the RM control problem to enhance it.

4. RM classic control problem

4.1 Brief history

The development of RM control algorithms has gone through at least three historical phases.

4.1.1 Model reference adaptive control and self-tuning control

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 RM 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.

4.1.3 Soft computing approach
Later methods, in the third period, concentrated on soft computing methods such as:

a. *Neural networks (NNs).*
In (Kwan *et al.*, 2001), a desired compensation adaptive law-based neural network 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 RM 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 RMs' articulations and collision free motion is obtained by the corrective action of the collision-avoidance fuzzy units.

4.2 RM dynamics
A standard mathematical model is needed for any RM control problem. The RM 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.

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

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.

4.3 RM 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 PID's (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. In expert-systems approach, the knowledge elicited from human operators is codified and embodied within the knowledge base in the form of IF-THEN rules.
The PID control *u(t)* is given by:

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

$$e(t) = q(t) - q_d(t) \tag{3}$$

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

Equation (1) describes the control *u(t)*. *Kp, Ki, Kv* 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 $q_d(t)$.
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 2.

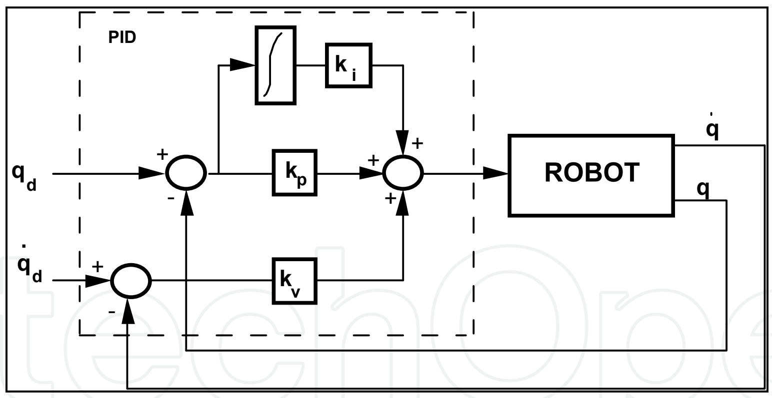


Fig. 2. RM PID Control

4.4 RM adaptive control

4.4.1 Purpose of adaptive control

The general adaptive controller design problem is as follows : given the desired trajectory $q_d(t)$, 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 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.

4.4.2 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 3.

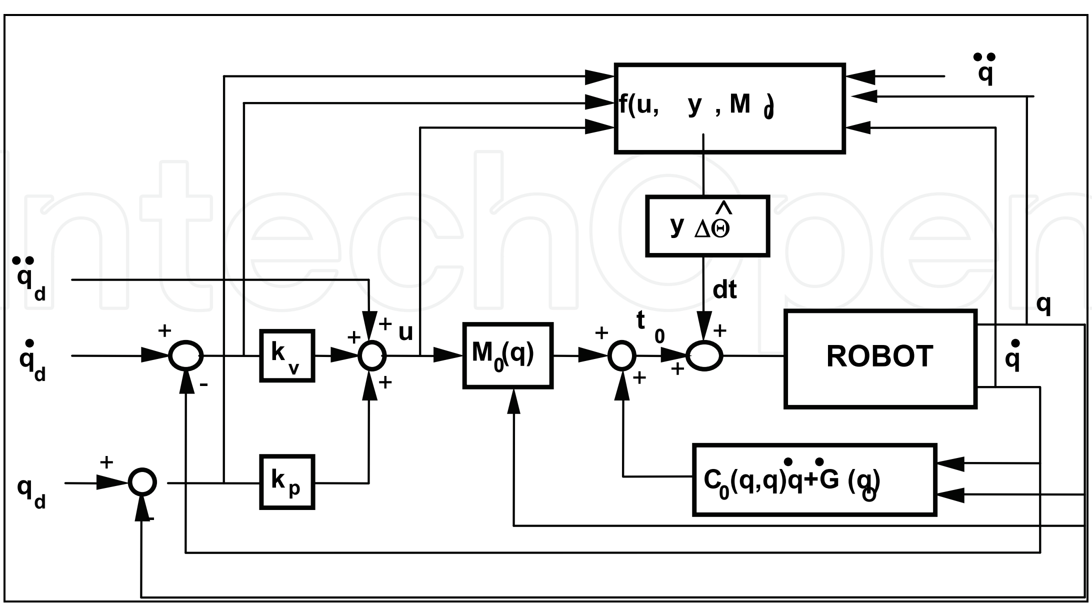


Fig. 3. Amestegui's adaptive compensation scheme

NB: In Figure 3, the following notations are used: $y=\psi$; $t=\tau$ $t_0=\tau_0$

4.5 RM 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.

4.5.1 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 RM 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 GI methods, since chattering can be modeled as an "odd" language departing from a nominal language, learned under normal operations.

4.5.2 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 RM 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.

4.5.3 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.

4.5.4 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 RM 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.
* $\boldsymbol{\delta u}$ 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 $\boldsymbol{\delta u}$ 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 $\boldsymbol{\delta u}$ 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 & Goldenberg, 1988) as given in equation (24).

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

The robust control scheme is represented in Figure 4.

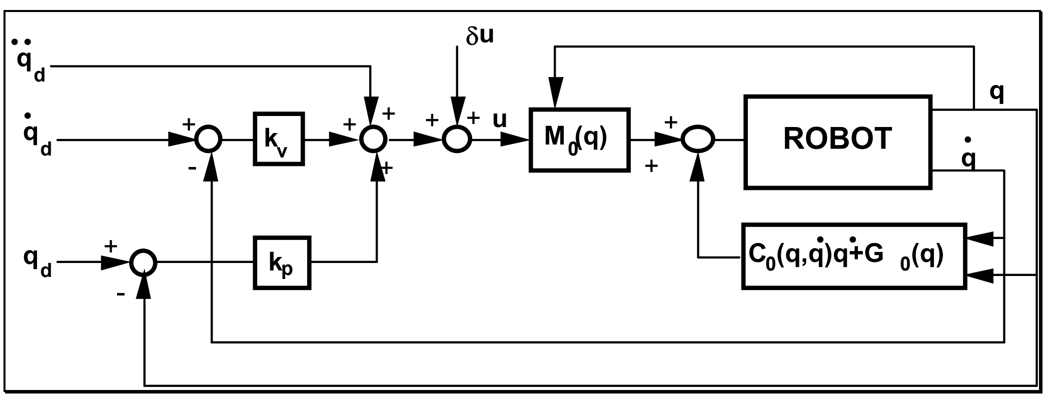


Fig. 4. Spong and Vidyasagar's robust control algorithm

4.6 Example of Implementation with Matlab/Simulink™

These implementations show two different classes of algorithms; one with adaptation and the other without.

5. GI for dynamical systems

5.1 Dynamical systems

A model for a controlled dynamical system has the general form

$$\dot{x}(t) = f(x(t),U(t)) \tag{25}$$

$$y(t) = h(x(y(t)) \tag{26}$$

or, considering it in a discrete-time form

$$x_{k+1} = f(x_k,U_k) \tag{27}$$

$$y_k = h(x_k) \tag{28}$$

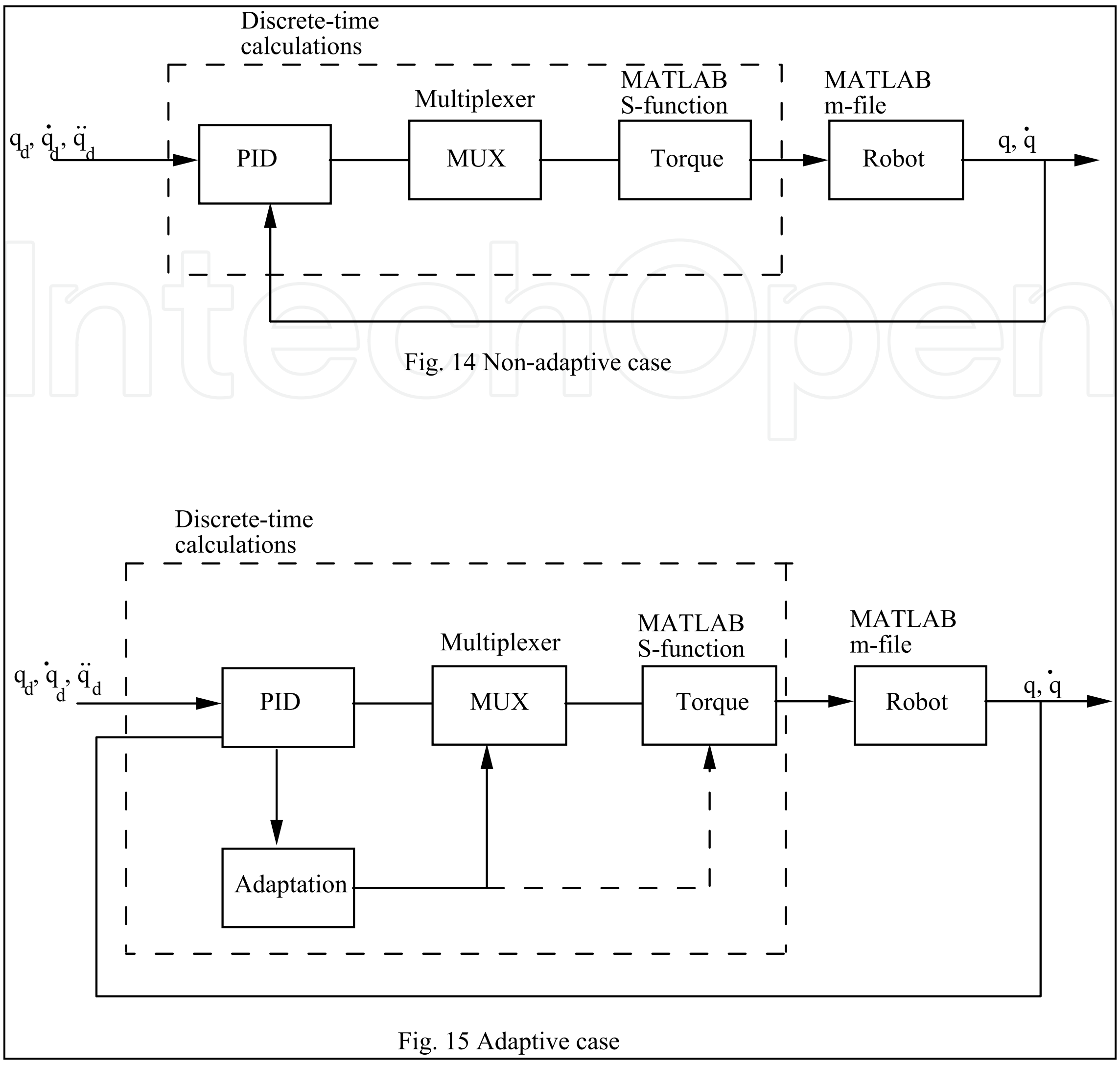


Fig. 5. RM classic control implementation with and without adaptation

where x is the state variable; y the output or observed variable; U the input or control variable; k denotes time in discrete case. Equations (25)-(28) also establish a functional relationship between the output variables at different times

$$y_{k+1} = g(x_k, U_k) \tag{29}$$

However, in most systems used in technology, including RM control, not all state variables are observable. Therefore, (29) does not provide a complete specification of the system. In general, specification of the dynamics in terms of the output variables requires a set of functional relationships involving many time steps in the past, namely:

$$\begin{aligned} y_{k+1} &= g_o(U_k) \\ y_{k+1} &= g_1(y_k, U_k) \\ y_{k+1} &= g_2(y_k, y_{k-1}, U_k) \\ y_{k+1} &= g_3(y_k, y_{k-1}, y_{k-2}, U_k) \\ y_{k+1} &= g_4(y_k, y_{k-1}, y_{k-2}, y_{k-3}, U_k) \end{aligned} \tag{30}$$

It is this structure which is required by dynamical considerations on actual controlled systems that leads in a natural way to the use of π-type productions, explained in the sequel.

5.2 Steps for using GI in control systems

To develop a grammatical description and a GI algorithm for controlled dynamical systems three steps are required (Martins *et al.*, 2006). First, the quantification of the variables are obtained, then the specification of the nature of the productions and finally a learning algorithm to extract the productions from the experimental data.

5.2.1 Quantification of the variables

Quantification refers to the choice of alphabets for the output (controlled) variable y and the control variable U. The objective is to generate the control U in order to maintain the output y within some prescribed values. A terminal alphabet T is associated to the output variable y and the nonterminal alphabet N to the control variable U. The feedback control law generates the required value of the input U so as to keep the controlled output y within a specified range. For so doing, a quantification of the variables is made, in a discrete way, dividing the variables range into equal intervals and associating each interval to a terminal symbol in the alphabet.

5.2.2 Production rules

π-type productions are defined by the human expert as some substitution rules of a given form. This human-supplied codification is necessary. A π-type production codes the evolution of the output variable, depending on its π past values and on the value of the control variable U. Therefore, there is a functional relationship between the dynamics of the system and the π-type productions. Note that a π-type production is usually written p-type. We prefer to represent it as π-type to avoid confusion with Proportional-control or P-type control action. An interesting line of research would be the use of knowledge-based systems approach to codify the human expertise and incorporate it with the final control system.

5.2.3 Learning

A learning algorithm is necessary to extract the productions from the experimental data. To obtain a sample of the language, a sequence of control signals is applied to the system in such a way that the output variable y takes values in a sufficiently wide region. The signal evolution is then quantified as described above, and a learning procedure is followed.

6. Results

For simplicity, we use a 2-symbol alphabet and show how the language is system generated by generalization, step by step.

6.1 Use of *ILSGINf*

ILSGINF is a heuristics-based inductive learning algorithm that induces grammars from positive examples. The main idea behind the algorithm is to take full advantage of the syntactic structure of available sentences. It divides the sentence into sub-sentences using partial derivatives PaDe's. Given a recognized sentence as reference, the parser is able to recognize part of the sentence (or sub-sentence(s)) while rejecting the other unrecognized part. Moreover the algorithm contributes to the resolution of a difficult problem in inductive learning and allows additional search reduction in the partial derivatives (PaDe's) space which is equal to the length of the sentence, in the worst case (Hamdi-Cherif, 2007). In the example, we suppose that all data are pre-processed from previous steps.

6.2 Example

6.2.1 ILSGInf results

We suppose that are given the following grammar for induction: G = (N, T P, S), where:

N = {S, A, B}, T ={b, *}, P = {S → AB, A → b, B→* A}

Let F= (b*b)*(b*b) be a global sentence to be parsed.
ILSGInf generates the following sub-sentences:

C_1 = (, C_2 = b * b, C_3 =), C_4 = *, C_5 = (, C_6 = b * b, C_7=)

Using the dotted (•) representation as in (Earley, 1970), *ILSGInf* gives the following results of sub-lists and sub-sentences:

	sub-list 0	*sub-list 1*	*sub-list 2*	*sub-list 3*
sub-sentence 1	$\mathbf{I}_{01}$ S → •AB, 0 A → • b, 0	$\mathbf{I}_{11}$ empty	$\mathbf{I}_{21}$ empty	$\mathbf{I}_{31}$ empty
sub-sentence 2	$\mathbf{I}_{02}$ S →• AB, 0 A → • b, 0	$\mathbf{I}_{12}$ A → b • , 0 S →A•B , 0 B →• +A, 1	$\mathbf{I}_{22}$ B →+•A, 1 A → • b , 2	$\mathbf{I}_{32}$ A → b • , 2 B →+A•, 1 S →AB•, 0
sub-sentence 3	$\mathbf{I}_{03}$ S →•AB, 0 A → • b , 0	$\mathbf{I}_{13}$ empty	$\mathbf{I}_{23}$ empty	$\mathbf{I}_{33}$ empty
sub-sentence 4	$\mathbf{I}_{04}$ S →•AB, 0 A → • b , 0	$\mathbf{I}_{14}$ empty	$\mathbf{I}_{24}$ empty	$\mathbf{I}_{34}$ empty
sub-sentence 5	$\mathbf{I}_{05}$ S →•AB, 0 A → • b , 0	$\mathbf{I}_{15}$ empty	$\mathbf{I}_{25}$ empty	$\mathbf{I}_{35}$ empty
sub-sentence 6	$\mathbf{I}_{06}$ S →• AB, 0 A → • b, 0	$\mathbf{I}_{16}$ A → b • , 0 S →A•B , 0 B→•+A,1	$\mathbf{I}_{26}$ B→ +•A, 1 A→ • b , 2	I_{36} A → b • , 2 B →+A•, 1 S →AB•, 0
sub-sentence 7	$\mathbf{I}_{07}$ S →•AB, 0 A→ • b , 0	$\mathbf{I}_{17}$ empty	$\mathbf{I}_{27}$ empty	$\mathbf{I}_{37}$ empty

Table 1. Progressive construction of sub-lists

6.1.2 Discussions

For the sub-sentences 1, 3, 4, 5 and 7, we note that:

i. I_{1x} (x=1,3,4,5,7) is empty. In this case, while no classical algorithm (*e.g.* Earley-like) proceeds further, the algorithm looks for other partial derivatives. Because sub-sentences are refused, then no transformation is needed.

ii. In sub-sentences 2, 6 all I_{3x} (x=2,6) are accepted. In each of these, we find an item of the form $S \rightarrow \alpha \bullet ,0$ which is $S \rightarrow AB \bullet ,0$. Then respective sub-sentences are totally accepted and transformed as S.

iii. Partial derivatives (PaDe's) of the global sentence (b*b)*(b*b) have the form: D = (S)*(S). Other partial derivatives of b*b are :
    b*A from item $A \rightarrow b \bullet ,2$ in I_{3x}, (x=2,6)
    bB from item $B \rightarrow {*}A \bullet ,1$ in I_{3x}, (x=2,6)
    A*b from item $A \rightarrow b \bullet ,0$ in I_{1x} (x = 2,6)
    AB from item $A \rightarrow b \bullet ,0$ in I_{1x} and I_{3x}, (x=2,6)

iv. Local sorting is done as follows: S, AB, bB, b*A, A*b.

7. Conclusion

We have described the foundational steps integrating robotic manipulator control and formal languages. More specifically, this research work reports some features of grammatical inference approach as applied to robotic manipulator control. As such, this research represents an early contribution towards an objective evaluation and a basic study of the effectiveness and usefulness of grammatical inference as applied to robotic manipulator control. Grammars and languages are used as supervising entities within control of robotic manipulators. A unification of the diversified works dealing with robotic manipulators, while concentrating on formal grammars as an alternative control method, is therefore made possible. The fundamental constraints of the proposed method is that it requires a choice of an appropriate quantification for the feature space. This choice has a direct impact on the size of the alphabets and the dimension and complexity of the grammars to be inferred. Like any machine learning method, the proposed procedure also requires a diversified coverage of the working domain during the learning stage to obtain rich generalization properties. As a consequence, the results report only some aspects of the overall issue, since these describe only the case of a small class of learnable languages. Much work is still required on both sides, *i.e.,* robotics and formal languages, for the development of fully-integrated systems that meet the challenges of efficient real-life applications.

8. 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.

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control*, Vol. 45, No. 2 (May 1980) page 117–135, ISSN: 0019-9958.

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.

Burbidge, R.; Walker, J.H. & Wilson, M.S. (2009). Grammatical evolution of a robot controller, *International Conference on Intelligent Robots and Systems, IROS 2009. IEEE/RSJ* pp. 357–362, ISBN: 978-1-4244-3803-7, St. Louis, MO, USA, 10-15 Oct. 2009, IEEE, New Jersey, USA.

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.

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.

Cicchello, O. & Kremer, S. C. (2003). Inducing grammars from sparse data sets: A survey of algorithms and results, *Journal of Machine Learning Research*, Vol. 4 (October 2003) page 603–632, ISSN: 1938-7228

Cohen, D.I.A. (1991). *Introduction to Computer Theory*. John Wiley & Sons, Inc., ISBN: 0-471-51010-6, New York, 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.

de La Higuera, C. (2005). A bibliographical study of grammatical inference. *Pattern Recognition*, Vol. 38 No. 9 (September 2005) page 1332-1348, ISSN: 0031-3203.

Earley, J., (1970). An efficient context-free parsing algorithm *Communications of the ACM,* Vol. 13, No. 2 (February 1970) pp. 94-102, ISSN: 0001-0782.

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.

Gold, E.M. (1967). Language identification in the limit, *Information and Control*, Vol. 10, No. 5 (1967) page 447–474, ISSN: 0019-9958.

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.

Hamdi-Cherif, A. (1994). The CASCIDA project – A computer-aided system control for interactive design and analysis. *Proceedings of IEEE / IFAC Joint Symposium On CACSD (CASCD'94)*, pp. 247-251, Tucson, AZ, 7-9 March 1994, IEEE, NJ, USA.

Hamdi-Cherif, C. & Hamdi-Cherif, A. (2007). ILSGInf : Inductive learning system for grammatical inference. *WSEAS Transactions on Computers*, Vol. 6, No. 7 (July 2007) page 991-996. ISSN: 1109-2750.

Hasemann, J.M. (1994). A robot control architecture based on graph grammars and fuzzy logic. *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94*, Vol.3, pp. 2123-2130, ISBN: 0-7803-1933-8, 12-16 Sep 1994, Munich, Germany.

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., Ghrist, R. & Lipsky, D. (2006). A grammatical approach to self-organizing robotic systems, *IEEE Transactions on Automatic Control.* Vol. 51, No. 6, (June 2006) page 949–962, ISSN: 0018-9286.

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.

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.

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

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.

Mitchell, T.M. (1997). *Machine Learning.* McGraw-Hill, ISBN: 0070428077, New York, USA.

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.

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

Sakakibara, Y. (1997). Recent advances in grammatical inference. *Theoretical Computer Science.* Vol. 185, No. 1 (October 1997) pp. 15–45, ISSN: 0304-3975.

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.

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.

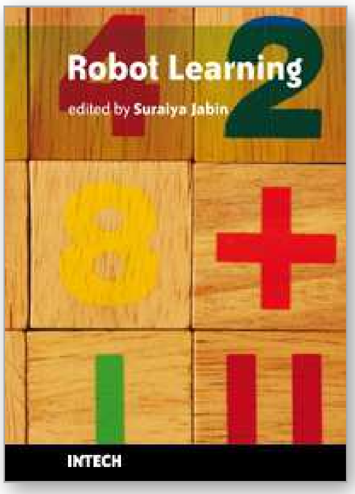
Slotine, J.J.E. & W. Li (1987). On the adaptive control of robot manipulators. *International Journal of Robotics Research,* Vol. 6, No. 3 (September 1987) page 49-59, ISSN: 0278-3649.

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

Unold, O., (2008). Grammar-based classifier system: a universal tool for grammatical inference. *WSEAS Transactions on Computers* Vol. 7, No. 10 (October 2008) page 1584-1593. ISSN: 1109-2750.

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 Learning
Edited by Suraiya Jabin

ISBN 978-953-307-104-6
Hard cover, 150 pages
Publisher Sciyo
Published online 12, August, 2010
Published in print edition August, 2010

Robot Learning is intended for one term advanced Machine Learning courses taken by students from different computer science research disciplines. This text has all the features of a renowned best selling text. It gives a focused introduction to the primary themes in a Robot learning course and demonstrates the relevance and practicality of various Machine Learning algorithms to a wide variety of real-world applications from evolutionary techniques to reinforcement learning, classification, control, uncertainty and many other important fields. Salient features: - Comprehensive coverage of Evolutionary Techniques, Reinforcement Learning and Uncertainty. - Precise mathematical language used without excessive formalism and abstraction. - Included applications demonstrate the utility of the subject in terms of real-world problems. - A separate chapter on Anticipatory-mechanisms-of-human-sensory-motor-coordination and biped locomotion. - Collection of most recent research on Robot Learning.

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

Aboubekeur Hamdi-Cherif (2010). Towards Robotic Manipulator Grammatical Control, Robot Learning, Suraiya Jabin (Ed.), ISBN: 978-953-307-104-6, InTech, Available from: http://www.intechopen.com/books/robot-learning/towards-robotic-manipulator-grammatical-control-

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

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