On the control of robot manipulator: A model-free approach

Hassane Abouaïssa, Samira Chouraqui

▶ **To cite this version:**

Hassane Abouaïssa, Samira Chouraqui. On the control of robot manipulator: A model-free approach. Journal of computational science, 2019, 31, pp.6 - 16. 10.1016/j.jocs.2018.12.011 . hal-03674957

HAL Id: hal-03674957

https://univ-artois.hal.science/hal-03674957v1

Submitted on 10 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the control of robot manipulator: A model-free approach

Hassane Abouaïssa[a], Samira Chouraqui[b]

[a] Univ. Artois, UR 3926, Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A)
Technoparc Futura
F-62400, Béthune, France

[b] Department of Computer Sciences,
University of Sciences and Technologies of Oran USTO'MB, Algeria

Abstract

The works presented in this paper deal with the control of a highly nonlinear and uncertain system using the new setting of "Model-Free Control" and its related intelligent Proportional, Integral and Derivative (iPID) regulators. Such approach that can be designed, using only the input and output data of the controlled device and the new algebraic method of identification, is applied to the control of multi/input- multi/output (MIMO) robot manipulators. Numerical simulations conducted for the robot PUMA 560 with 6° of freedom (6-DOF) show the effectiveness of the method and the easiness of the tuning of the gains parameters of the used regulators.

keyword:

Robot manipulator, PUMA 560, Control, Algebraic methods of identification, Model-free control,

1 Introduction

Designing controllers for industrial processes or guided vehicles, nonlinear functions are often encountered. For example, it is not unusual for the equations that govern some controlled process or plant to be nonlinear. Even with a plant that can be considered linear, if the response of the system is to be optimal in some way, practical limitations on the actuator may lead to a required drive signal that is highly nonlinear. Intentional nonlinearity may then be introduced to achieve the desired performance.

Nonlinear control design is generally difficult to apply to practical systems. Over the years, this fact has led to the application of linear methods to problems best suited to nonlinear design. The resulting response may be far less desirable than the optimal response, but the design methods used are far less complex than nonlinear approaches. Because of the complexities of nonlinear analysis, procedures such as describing function methods are not amenable to systems of third or higher order [1].

Multi-degree-of-freedom (M-DOF) robot manipulators become an integral part of industrial applications and being extensively employed in industries such as automobile, Medical, space exploration, search and rescue, underwater exploration and in military. Modern manipulators are designed complicatedly and need to do more precise, crucial and critical tasks. Therefore, the simple traditional control methods cannot be efficient, and advanced control strategies with considering special constraints are needed to establish.

The robot, as the plant to be controlled, is a multi-input/multi- output, highly coupled and nonlinear mechatronic system. The main challenges in the motion control problem are the both para- metric and dynamic complexities. Parametric uncertainties arise from imprecise knowledge of the kinematics and dynamics, while dynamic uncertainties occur from joint and link flexibility, actuator dynamics, friction, sensor noise, and unknown environment dynamics [2]. It is therefore, very difficult to implement real time control based on a detailed dynamic model of a robot. Moreover, these systems operate in an unstructured environment and are always subjected to environmental disturbances. A better solution to the complex control problem might result in using an intelligent control. One of the major tasks for an intelligent control system is to provide a desired or satisfactory system behavior under an unknown and/or uncertain controlled environment. In this paper, we focus on the application of intelligent system methodology in the design of a "non-math-knowledge" oriented or so-called dynamic "model-free-control" system, which has robust and adaptive behavior [3].

Among different approaches found in literature, this paper is interested to a strategy developed recently by M. Fliess and C. Join [4], known as Model-Free-Control (MFC) to control the 6-DOF PUMA 560 arm manipulator. This choice is justified by the simplicity of design and the ease of practical implementation for this control algorithm based on an elementary continuously updated local modeling via the unique knowledge of the input-output behavior, where the need of any mathematical model disappears. Although this concept is relatively new several successfully appli- cations in a number of practical studies have been implemented in many areas, essentially experimental greenhouses [5], wheeled autonomous vehicles [6], thermal processes [7] and among others the highway traffic [4,8]. To prove the effectiveness of this new algorithm in the field of robotics and nonlinear complex systems, the proposed Model Free Control (MFC) technique has been implemented in a highly coupled nonlinear and high dynamic process, such as a six degree of freedom PUMA 560 ar m manipulator, widely used in robotics research for which there is a substantial literature. Results obtained have been tested with a precise and widely used method in automation area known as computed torque controller (PD-Torque) [9], which is highlighted on the following sections of this paper.

The rest of the paper is organized as follows. Section 2 resumes the related works in the field of control strategies with different characteristics and complexities. For simulations purpose a description of the mathematical model of the studied robot "PUMA 560" and the application of the computed torque controller are detailed in Section 3. The same section relates some basics of the new concept of "Model-free control" and its implementation to the studied robot manipulator. Section 4, gives some diagrams of the studied system in Matlab/Simulink. The conducted numerical simulations are shown in Section 5. Section 6 provides a conclusion and some furthers works.

2 Related work

A number of control algorithms with different characteristics and complexities have been developed (see e.g., [10–15] and references therein). One of the most important powerful nonlinear robust controllers is computed torque controller (PD-Torque) [10]. Computed torque controller (PD-Torque) is a powerful nonlinear controller, which is widely used in control robot manipulator. It is based on Feedback linearization and computes the required arm torques using the nonlinear feedback control law. This controller works very well when all dynamic and physical parameters are known but when the robot manipulator has variation in dynamic parameters, in this situation the controller has no acceptable performance [16]. Since, it is difficult to obtain a good mathematical model of the process dynamics and to know the different disturbances acting

on the process, the Model-based control techniques such as PD-Torque controller cannot provide satisfactory results when applied to poorly modeled processes, which can operate in ill-defined environments. Research on computed torque controller is significantly growing on robot manipulator application, which has been reported in [9,10,14,15]. In order to overcome this problem, model-free control techniques that can be directly applied to complex processes have been given much attention in the control community during the last decade [17,18]. In nowadays, the best-known classical model free industrial process controller is the PID controller [19] because of its simplicity, robustness, high reliability and it can be easily implemented on any processor, nevertheless the tuning gain of such regulators is very difficult and classical PID controllers are usually not sufficiently powerful control tools. In recent years, there has been an increasing interest in the utilization of unconventional control strategies such as neural networks (NN), fuzzy logic, and genetic algorithm (GA) [20]. Fuzzy logic reasoning is based on human experience and expert knowledge [21,22]. Neural network reasoning depends on the extraction of hidden relationships in given datasets. It has the ability to learn from examples, drawing conclusions based on experiences [22,23]. Genetic algorithm is essentially an optimization technique based on the ideas of evolution in biological development. It has the ability to obtain systematically solutions in complex problems [24,25]. These control methods derive their advantages from the fact that they do not use any mathematical model of the system, however, sometimes these methods take quite a long time to find a coefficient that satisfies the requirement of the controlling task. In addition, lack of theoretical analysis and stability security makes industrialists wary of using the results in real industrial environments. A possible solution to avoid the identification of the model parameters to efficiently control a complex process like robot arm manipulators, a new technique in the framework of "model-free control" [4,26,27,28] is considered. The approach uses derivative estimations [29–31], which provides good results even if the measured signals are corrupted by noise. Thus, a non-physical model valid during a very short period of time is estimated and permits classic control design.

3 Control design

In this section, the process of controller design is presented. In this study, a model free controller (MFC) is used to control the six degree of freedom robot system. In order to compare the performance of the studied controller a detailed mathematical model is performed for 6DOF-PUMA 560 and implemented with Computed torque controller (PD-Torque) which is a powerful nonlinear controller widely used in control of robot manipulator. Results obtained have been compared with the newest model free control (MFC) method.

3.1 Robot manipulators dynamics

The system under study is a six-degree robot arm manipulator developed specifically for research purposes. An important initial step in analyzing, designing or controlling a complex mechanical system, such us a robot, is to construct a representative model of the system. The PUMA 560 has six revolute joints as shown in Fig. 1.

The dynamics of robot manipulators with rigid links describes the relationship between displacement, velocity and acceleration to force acting on robot manipulator, the application of the Euler-Lagrange equations to each link results in a system of coupled differential equations. In matrix form, it can be written as [10,11]:

$$M(q)\ddot{q} = V(q,\dot{q})\dot{q} + F(\dot{q})G(q) = \Gamma \tag{1}$$

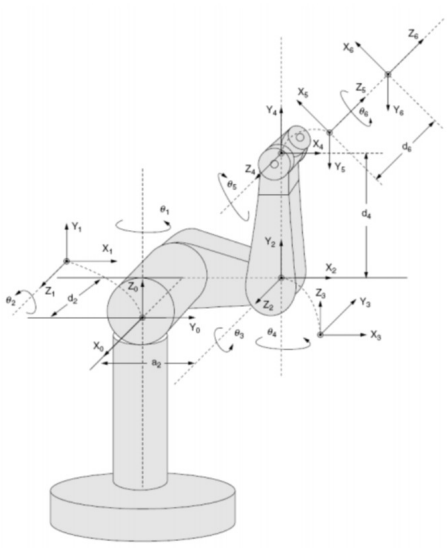

Figure 1: D–H notation for a six-degrees-of-freedom PUMA 560 robot manipulator [7]

where $M(q)$ is the inertia matrix, $V(q, \dot{q})$ represents the Coriolis/centripetal matrix, $F(\dot{q})$ are the friction terms, $G(q)$ is the gravity vector, and Γ is the control input vector. q, $\dot{q}$, $\ddot{q}$ are the $n \times 1$ vectors of the joint position, velocity and acceleration respectively.

By writing the velocity dependent term $V(q, \dot{q})$ in a different form, all the matrices become functions of only the manipulator position; in this case the dynamic equation is called configuration space equation and has the following form:

$$\Gamma = M(q)\ddot{q} = B(q)[\dot{q}.\dot{q}] + C(q)[\dot{q}^2] + G(q) \tag{2}$$

where, $B(q)$: $n \times n(n-1)/2$ matrix of Coriolis torques, $C(q) : n \times n$ matrix of Centrifugal torques, $[\dot{q}\dot{q}] : n(n-1)/2 \times 1$ vector of joint velocity products given by:

$$[\dot{q}_1\dot{q}_2, \dot{q}_1\dot{q}_3, \dot{q}_1\dot{q}_n, \dot{q}_2\dot{q}_3, \dot{q}_2\dot{q}_4, \ldots \dot{q}_{n-2}\dot{q}_n, \dot{q}_{n-1}\dot{q}_n]^T$$

$[\dot{q}^2] : n \times 1$ vector given by: $[\dot{q}_1^2, \dot{q}_2^2, \ldots, \dot{q}_n^2]$

To derive the model of the robot arm, Amstrong et al. [32] started by generating the kinetic energy matrix and gravity vector symbolic elements by performing the summation of either Lagrange's or the Gibbs-Alembert formulation; these elements are then simplified by combining inertia constants that multiply common variable expressions. The Coriolis and centrifugal matrix elements are then calculated in terms of partial derivatives of kinetic energy, and then reduced using four relations that hold on the partial derivatives. Finally, the necessary partial derivatives are formed, and the Coriolis and centrifugal matrices are found. A simplification step is then done by combining the inertia constants that multiply the common variable expressions. All the parameters and the explicit dynamic model of the PUMA 560 used in the simulation can be found in [33, 34].

3.1.1 Computed torque control for 6DOF-PUMA560

The Computed Torque Control (PD-Torque) is a controller used in wide range areas such as robotics, control process and aerospace applications, because it has an acceptable control performance and solve some main challenging topics in control such as resistivity to the external disturbance [9]. This controller rely detailed knowledge of the dynamic characteristic of the robot arm expressed in an accurate mathematical model.

From the dynamic model in Eq. (2), the computed torque control law is stated as follows:

$$\Gamma = M(q)\ddot{q}^{\star} + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}^2] + G(q) \tag{3}$$

where,
M represents a symmetric 6×6 inertia matrix

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} & 0 & 0 & 0 \\ m_{31} & m_{32} & m_{33} & 0 & m_{35} & 0 \\ 0 & 0 & 0 & m_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} \tag{4}$$

and
m_{11}=$I_{m1} + I_1 + I_3CC_2 + I_7SS_{23} + I_{10}SC_{23} + I_{11}SC_2 + I_{21}SS_{23}$
$+2\,[I_5C_2S_{23} + I_{12}C_2C_{23} + I_{15}SS_{23} + I_{16}C_2S23 + I_{22}SC_{23}]$
m_{12}=$I_4S_2 + I_8C_{23} + I_9C_2 + I_{13}S_{23}$
m_{13}=$I_8C_{23} + I_{13}S_{23} + I_{18}C_{23}$
m_{22}=$I_{m2} + I_2 + I_6 + 2\,[I_5.S_3 + I_{12}C_2 + I_{15} + I_{16}S_3]$
m_{23}=$I_5S_3 + I_6 + I_{12}C3 + I_16S_3 + 2I_{15}$
m_{33}=$I_{m3} + I_6 + 2I_{15}$
m_{35}=$I_{15} + I_{17}$
m_{44}=$I_{m4} + I_{14}$
m_{55}=$I_{m5} + I_{17}$
m_{66}=$I_{m6} + I_{23}$
$m_{21} = m_{12}$, $m_{31} = m_{13}$ and $m_{32} = m_{23}$
while the matrix of Coriolis torques B is:

$$B(q) = \begin{bmatrix} b_{112} & b_{113} & 0 & b_{115} & 0 & b_{123} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{214} & 0 & 0 & b_{223} & 0 & b_{225} & 0 & 0 & b_{235} & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{314} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_{412} & b_{413} & 0 & b_{415} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{514} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

and,

- b_{112}=$2[-I_3SC2 + I_5C_{223} + I_7SC_{23} - I_{12}S_{223} + I_{15}2SC_{23} + I_{16}C_{223} + I_{21}SC_{23} + I_{22}(1 - 2.SS_{23})] + I_{10}(1_2SS_{23}) + I_{11}(1 - 2SS_2)$
- b_{113} =$2[I_5C_2C_{23} - I_7SC_{23} - I_{12}C_2S_{23} + I_{15}2SC_{23} + I_{16}C_2C_{23} + I_{21}SC_{23} + + I_22(1 - 2SS_{23})] + I_10(1 - 2SS_{23})$
- b_{115} =$2[-SC_{23} + I_{15}SC_{23} + I_{16}C_2C_{23} + I_{22}CC_{23}]$

- b_{123} =2$[-I_8S_{23} + I_{13}C_{23} + I_{18}S_{23}]$
- b_{214} =$I_{14}S_{23} + I_{19}S_{23} + 2I_{20}S_{23}(1 - 0.5)$
- b_{223} =2$[-I_{12}S_3 + I_5C_3 + I_{16}C_3]$
- b_{225}=2 $[I_{16}C_3 + I_{22}]$
- b_{235}=2 $[I_{16}C_3 + I_{22}]$
- b_{314}=2$[I_{20}S_{23}(1 - 0.5)] + I_{14}S_{23} + I_{19}S_{23}$
- $b_{412} = -b_{214}$=-$[I_{14}S_{23} + I_{19}S_{23} + 2I_{20}S_{23}(1 - 0.5)]$
- $b_{413} = -b_{314}2[I_{20}S_{23}(1 - 0.5)] + I_{14}S_{23} + I_{19}S_{23}$
- b_{415}=$-I_{20}S_{23} - I_{17}S_{23}$
- b_{514}=$-b_{415} = I_{20}S_{23} + I_{17}S_23$

The Coriolis matrix C is:

$$C(q) = \begin{bmatrix} 0 & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{21} & 0 & c_{23} & 0 & 0 & 0 \\ c_{31} & c_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c_{51} & c_{52} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

and,

- c_{12}=$I_4c_2 - I_8S_{23} - I_{13}c_23 + I_{18}S_23$
- $c_{13} = 0.5b123$=$-I_8S_{23} + I_{13}C_{23} + I_{18}S_{23}$
- $c_{21} = -0.5b112$=$I_3SC2 + I_5C_{223} + I_7SC_{23} - I_{12}S_{223} + I_{15}2SC_{23} + I_{16}C_{223} + I_{21}SC_{23} + I_{22}(1 - 2.SS_{23})] + I_{10}(1_2SS_{23}) + I_{11}(1 - 2SS_2)$
- $c_{32} = -c_{23}$=$I_{12}S_3 - I_5C_3 - I_{16}C_3$
- $c_{51} = -0.5b_{115}$=$SC_{23} - I_{15}SC_{23} - I_{16}C_2C_{23} - I_{22}CC_{23}$
- $c_{52} = -0.5b_{225}$=$-I_{16}C_3 - I_{22}$

The gravity vector G is:

$$G(q) = \begin{bmatrix} 0 \\ g_2 \\ g_3 \\ 0 \\ g_5 \\ 0 \end{bmatrix} \tag{7}$$

- $g_2 = g_1C_2 + g_2S_{23} + g_3S_2 + g_4C23 + g_5S_{23}$
- $g_3 = g_2S_{23} + g_4C23 + g_5S_{23}$
- $g_5 = g_5S_{23}$

Table 1: Initial constants (kg/m^2)

$I_1 = 1.43 \pm 0.05$	$I_2 = 1.75 \pm 0.07$
$I_3 = 1.38 \pm 0.05$	$I_4 = 0.69 \pm 0.02$
$I_5 = 0.372 \pm 0.031$	$I_6 = 0.333 \pm 0.016$
$I_7 = 0.298 \pm 0.029$	$I_8 = -0.134 \pm 0.014$
$I_9 = 0.038 \pm 0.012$	$I_{10} = -0.0213 \pm 0.0022$
$I_{11} = -0.0142 \pm 0.0070$	$I_{12} = -0.011 \pm 0.0011$
$I_{13} = -0.00379 \pm 0.0009$	$I_{14} = 0.00164 \pm 0.07$
$I_{15} = 0.00125 \pm 0.0003$	$I_{16} = 0.00124 \pm 0.0003$
$I_{17} = 0.000642 \pm 0.0003$	$I_{18} = 0.000431 \pm 0.00013$
$I_{19} = 0.0003 \pm 0.0014$	$I_{20} = -0.000202 \pm 0.0008$
$I_{21} = -0.0001 \pm 0.0006$	$I_{22} = -0.000058 \pm 0.000015$
$I_{23} = 0.00004 \pm 0.00002$	$I_{m1} = 1.14 \pm 0.27$
$I_{m2} = 4.71 \pm 0.54$	$I_{m3} = 0.827 \pm 0.093$
$I_{m4} = 0.2 \pm 0.0016$	$I_{m5} = 0.179 \pm 0.014$
$I_{m6} = 0.193 \pm 0.016$	

Table 2: Gravitational constants ($N.m$)

$g_1 = -37.2 \pm 0.5$	$g_2 = -8.44 \pm 0.20$
$g_3 = 1.02 \pm 0.50$	$g_4 = 0.249 \pm 0.025$
$g_5 = -0.0282 \pm 0.0056$	

where,
$S_i = sin(\theta_i)$, $C_i = cos(\theta_i)$,$C_{ij} = cos(\theta_i + \theta_j)$, $S_{ijk} = sin(\theta_i + \theta_j + \theta_k)$,
$CC_i = cos(\theta_i)cos(\theta_i)$ and $Cs_i = cos(\theta_i)sin(\theta_i)$

Table. 1 and 2 contain the computed values for the constants appearing in the equations of forces of motion,

The expression of the acceleration term $\ddot{q}^\star$ in Eq. (3) is defined by Eq. (8) as below.

$$\ddot{q}^\star = \ddot{q}_d + K_p(q_d - q) + K_v(q_d - q) \tag{8}$$

where, $\ddot{q}_d$ is the vector of desired acceleration with respect to the joint coordinates of the robot, $\dot{q}_d$ represents the vector of desired velocity with respect to the joint coordinates of the robot. $\dot{q}$ is the vector of actual velocity with respect to the joint coordinates. q_d is the vector of desired position with respect to the joint coordinates. q is the vector of actual position with respect to the joint coordinates of the robot. K_p is the proportional gain matrix, K_v is the derivative gain matrix. The proportional and the derivative gains can be designed by knowing the physical characteristics of the system or by following empirical tuning methods like the Zeigler-Nichols method [34]. Thus, for every sampling period, from the position and velocity feedback information, the corrected accelerations estimated using Eq. (8) and thereby, the control torque can be computed using the control law Eq. (3) [9]. The computed torque signal is further communicated to the torque servo module for real-time execution. This process is carried out repeatedly, through the trajectory period, at regular sampling intervals, until the desired trajectory is completed.

3.2 Model-free control

As robots and their tasks become more complex, the effectiveness of model based control methods begins to decline. Firstly, the often highly nonlinear dynamic model of the robot becomes more difficult to establish with sufficient accuracy. Particularly, effects such as joint friction and link flexibility can make accurate models difficult to derive. Secondly, strategies such as computed torque control, which require that the robot's dynamic model be recalculated during every sample period can, place exceptional demands on computing resources, especially when rapid sampling is required. To avoid the drawbacks resulting from model uncertainties and/or model truncations, a very recent approach to nonlinear control that has been introduced by M. Fliess and C. Join [4] is proposed.

3.2.1 Genesis of the ultra-local model concept

For a sake of simplicity, consider a SISO[1] Systems. The ultra-local model is based on local modeling, continually updated, from the solely knowledge of input-output behavior (See e.g. [35]). For the unknown differential equation:

$$E\left(y, \dot{y}, \ldots, y^{(\iota)}, u, \dot{u}, \ldots, u^{(\kappa)}\right) = 0 \tag{9}$$

Notice that Eq. (9) can be linear or not, with u the system input and y the system output.

E is a sufficiently smooth function of its arguments. Assume also that for for an integer ν, $0 < \nu < 1$, $\frac{\partial E}{\partial y^{(\nu)}} \neq 0$. The implicit function theorem permits to write

$$y^{(\nu)} = \phi\left(t, y, \dot{y}, \ldots, y^{(\iota-1)}, y^{(\iota+1)}, \ldots, y^{\gamma}, u, \dot{u}, \ldots, u^{(\kappa)}\right) \tag{10}$$

Ultra-local model control consists in trying to estimate via the input and the output measurements what can be compensated by control in order to achieve a good output trajectory tracking. This implies the construction of a purely ultra-local model of the system which replaces (2) and that, can be written as:

$$y^{(\nu)}(t) = F(t) + \alpha u(t) \tag{11}$$

where

- y represents the output of the system,
- ν is the time derivative order, usually no more than 2 because it is sufficient to describe the behavior of the system.
- α is a parameter that can be defined by the practitioner such that αu and $y^{(\nu)}$ are of the same magnitude.
- F groups all the unknown signals (disturbance, noise, ...) and the imperfections of the model, it is estimated at each sample time.

Emphasis that in many industrial applications, the order of ν is often equal to 1 or 2. Assume that $\nu = 2$ in Eq. (11), which leads to the following expression:

$$\ddot{y} = F + \alpha u \tag{12}$$

The ultra-local model can be seen as an approximated model for system dynamics valid for a short period of time that allows a real-time update. This approach attempts the simplification

[1] Single-Input Single-Output

of the nonlinear control by discarding the need for a global and complex nonlinear model. The numerical value of F must be updated every short period T since Eq. (**??**) is valid for a short time window. Using the knowledge of the input u and the measurement of the output y, F can be estimated based on the online parameter identification techniques described in the following sections.

3.2.2 Estimation of F: first method

Assume that the first term in Eq. (12) may be well approximated by a piecewise constant function F_{est} which is valid during a short lapse of time. Rewrite, then the Eq.(12) in Laplace domain (See e.g. [36])

$$s^2Y(s) - sy(0) - \dot{y}(0) = \frac{F_{est}}{s} + \alpha U(s)$$

Take the fist derivative[2] with respect to s in order to get a rid of $\dot{y}(0)$

$$s^2\frac{dY(s)}{ds} + 2sY(s) - y(0) = -s^{(-2)}F_{est} + \alpha\frac{dU(s)}{ds}$$

Differentiating the above equation permits to eliminate the initial condition $y(0)$

$$s^2\frac{d^2Y(s)}{ds^2} + 4s\frac{dY(s)}{ds} + 2Y(s) = 2s^{(-3)}F_{est} + \alpha\frac{d^2U(s)}{ds^2}$$

Noise attenuation is achieved by multiplying both sides on the left by s^{-3}, since integration with respect to time is a lowpass filter [37].

$$s^{-1}\frac{d^2Y(s)}{ds^2} + 4s^{-2}\frac{dY(s)}{ds} + 2s^{-3}Y(s) = 2s^{(-6)}F_{est} + \alpha s^{-3}\frac{d^2U(s)}{ds^2}$$

It yields in the time domain, using the well known Cauchy rule (See e.g. [36]) the real-time estimate,

$$F_{est}(t) = \frac{60}{t^5}\int_0^t \left(t^2 - 6t\tau + 6\tau^2\right) y(\tau) - \frac{30\alpha}{t^5}\int_0^t (t-\tau)^2\,\tau^2 u(\tau)d\tau \tag{13}$$

where $\tau > 0$ might be quite small

Remark 3.1. *The above expression (17) can be digitally implemented resulting in a FIR filter. For this purpose, the method is modified and we integrate backwards in a small fixed length window T to provide a feasible implementation in real-time.*

3.2.3 Estimation of F: second method

If we close the loop, for example by an iP (intelligent proportional) controller

$$u = \frac{F_{est} - \ddot{y}^\star + K_p e}{\alpha} \tag{14}$$

where,

- $\ddot{y}^\star$ is the second derivative of the reference trajectory $y^\star$

[2]Derivative said "Algebraic": $\frac{d}{ds}$ corresponds, in time domain, to the multiplication by $-t$. In the same order, $\frac{d^n}{ds^n} \to (-1)^n t^n$.

- $e = y - y^\star$ is the tracking error
- K_p is a usual tuning gain

we obtain immediately the estimation value of F.

$$F_{est}(t) = \frac{1}{\tau}\left[\int_{t-\tau}^{t} ((\ddot{y}^\star - \alpha u - K_p e)d\tau)\right] \tag{15}$$

as stated in [35, 36], note the following facts

- $\tau > 0$ may be chosen quite small.
- Integrals (17) and (15) are low pass filters.
- These integrals may be replaced for implementation purpose by classic digital filters.

3.3 Application to the robot manipulator control design

3.3.1 Principle

Based on the numerical knowledge of F previously estimated, the control for sampling period k is calculated using Eq.(12) as a simple cancellation of the nonlinear terms F plus a closed loop tracking of a reference trajectory $y^\star$:

$$u = -\frac{F - \ddot{y}^\star + K_p e}{\alpha} \tag{16}$$

where $e = y - y^\star$ is the tracking error. According to the algebraic parameter identification methods, F can be approximated via the operational calculus mentioned in previous sections. The control scheme can be summarized as in Fig. reffig:2.

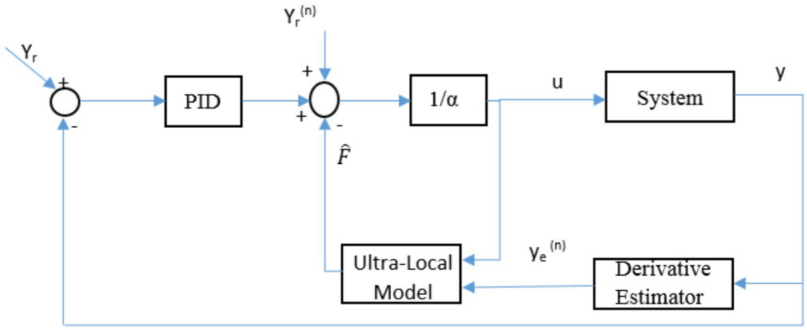

Figure 2: Model-free control scheme

3.3.2 Model-free control development for the PUMA6DOF manipulator

PUMA 560 is a six degree of freedom (6 DOF) arm manipulator; the outputs of the system are accelerations $(\ddot{q}_1, \ddot{q}_2, \ldots, \ddot{q}_6)$, velocities $(\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_6)$ and positions $(q_1, q_2, \ldots, q_6)$. According to the model-free control concept, we do not need any information about the system model. During a short time step ΔT, the system model is replaced by the ultra-local one as follow:

$$\ddot{y}_n(t) = F_n(t) + \alpha_n u_n(t), \quad n = 1, \ldots 6. \tag{17}$$

where, F_n represents the whole structural and unmodeled or time-varying part of the system which must be updated automatically. This parameter can be estimated either by the first or a second method depicted in Section 3. For example, the value of F is evaluated from u and $\ddot{y}$ at any time step as the following:

$$F_{n,est} = \ddot{y}_{n,est} - \alpha_n u_n(k-1) \tag{18}$$

where, $F_{n,est}$, $\ddot{y}_{n,est}(k)$ are the estimated values of F_n and $\ddot{y}^{\star}(k)$ at time step k, $u(k-1)$ is the command at time step $k-1$. Closing the loop with a linearizing control including a Proportional-Derivative action we obtain the intelligent proportional-derivative controller, or iPD,

$$u_n(k) = \frac{F_{n,est}(k)}{\alpha_n} + \frac{\ddot{y}_n^{\star}(k)}{\alpha_n} + K_{P_n} e_n(k) + K_{D_n} \dot{e}(k) \tag{19}$$

where $u_n(k)$ is the command at time step k; $y_n^{\star}(k)$ is the desired trajectory at time step k. The tracking error indicating how well the manipulator follows the desired trajectory is defined by

$$e_n(k) = y_n(k) - y_n^{\star}(k) \tag{20}$$

K_{P_n}, K_{D_n} are the classic PD tuning gains.

In this section a Simulink, which is a Block Diagram Simulation Tool used together with MATLAB, implementation of the arm manipulator and the proposed intelligent model free controller is presented. The Plant PUMA 560 was implemented in Simulink as represented in Fig. 3, ($q, \dot{q}$ and $\ddot{q}$) symbolize respectively the position, velocity and acceleration of the robot arm. $M(q)$, $B(q)$, $C(q)$ and $G(q)$ characterize inertia matrix, matrix of Coriolis torque, centrifugal torque matrix and gravity vector respectively.

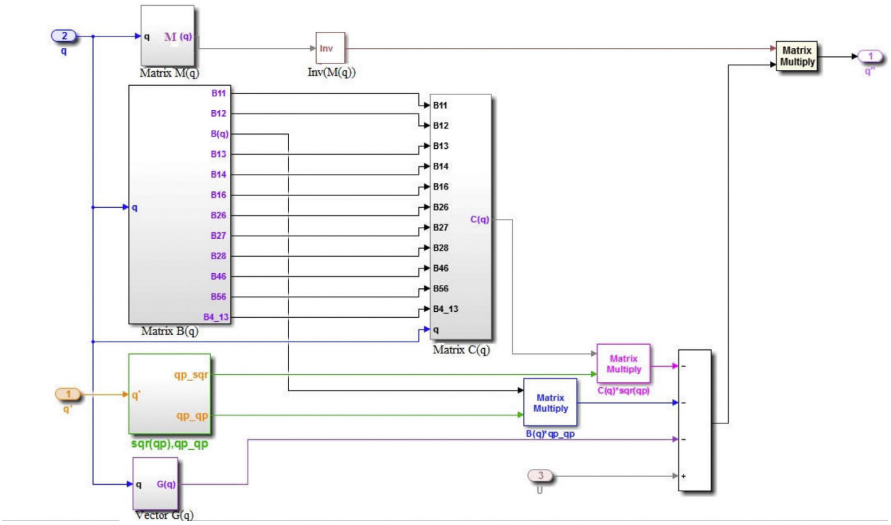

Figure 3: Simulink Block Diagram for PUMA560 arm.

The proposed MFC controller block scheme applied for the trajectory control of 6-DOF PUMA 560 robot arms is given by the Fig. 4 below. The blocks MFCs characterize the model free control of each joint q_n, $(n = 1, \ldots, 6)$, the value of u_n $(n = 1, \ldots, 6)$ is given in Eq. (19). For

simulations purpose the accelerations $\ddot{q}_n$ of the 6-DOF PUMA 560 have been calculated using Eqs. (2) and (19),

$$\ddot{q}_n = M^{-1}(q)\left\{\frac{F_{n,est}(k)}{\alpha_n} + \frac{\ddot{y}_n^{\star}(k)}{\alpha_n(k)} + K_{P_n}e_n(k) + K_{D_n}\dot{e}(k) - \left[B(q_n)\dot{q}_n\dot{q}_n + C(q_n)\dot{q}_n^2 + G(q_n)\right]\right\} \tag{21}$$

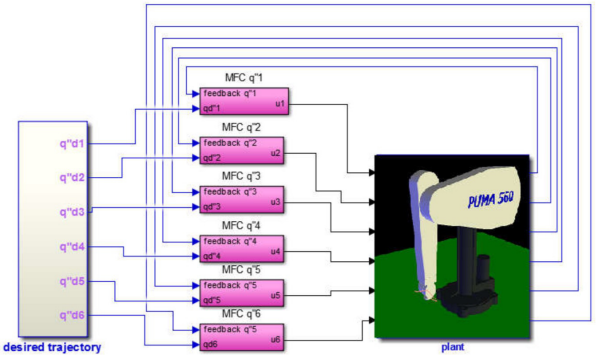

Figure 4: MFC structure for $PUMA560$ control

4 Simulation results

All simulations have been performed using MATLAB and SIMULINK [38] which are used widely in control applications. The control performances obtained with Model Free Controller (MFC) are compared with the performances achieved using computed torque control (PD-Torque). This controller allows the design of considerably more precise, energy efficient, lower feedback gains and complaint controls for robots but requires knowledge of all manipulator parameters and its payloads, which may not be realistic [39]. The obtained results using the 4th order Runge-Kutta solver with fixed step time $\Delta t = 0.001$ are shown below.

Results for the two controllers (PD-Torque and MFC) are shown in Figs. 5–8, in which the output of the closed loop system and the tracking errors are given. Using the integral of the accelerations leads to the time evolution of the speed of the studied device (See. Fig. 6). Fig. 7 illustrates the evolution of the manipulator. From the obtained different evolutions of the manipulator, one can plot the error tracking as depicted in Fig. 8. Fig. 9 illustrates the error (Radians vs. Seconds) between the computed torque controller and the Model Free Control. From Figs. 5–7 it can be seen that trajectories (red color) with the proposed MFC controller are driven to the desired trajectories (blue color) precisely and quickly. As we can see from figures, the MFC trajectories are very close to the desired trajectories than trajectories with PD-Torque controller, which demonstrate the validity of the proposed method to control highly nonlinear complex system. For example if we take the Fig. 5-a (joint 1 acceleration), the trajectory of acceleration with the model free control system (red color) is very close to the desired trajectory (blue color). On the other hand, the computed torque trajectory (Black color) outlying the desired trajectory. We can say the same thing about joints velocities and positions. For example, one can see in Fig. 7, all joints positions with model free controller (red color) are neighboring the reel (blue color) position than positions (black color) controlled by computed torque controller.

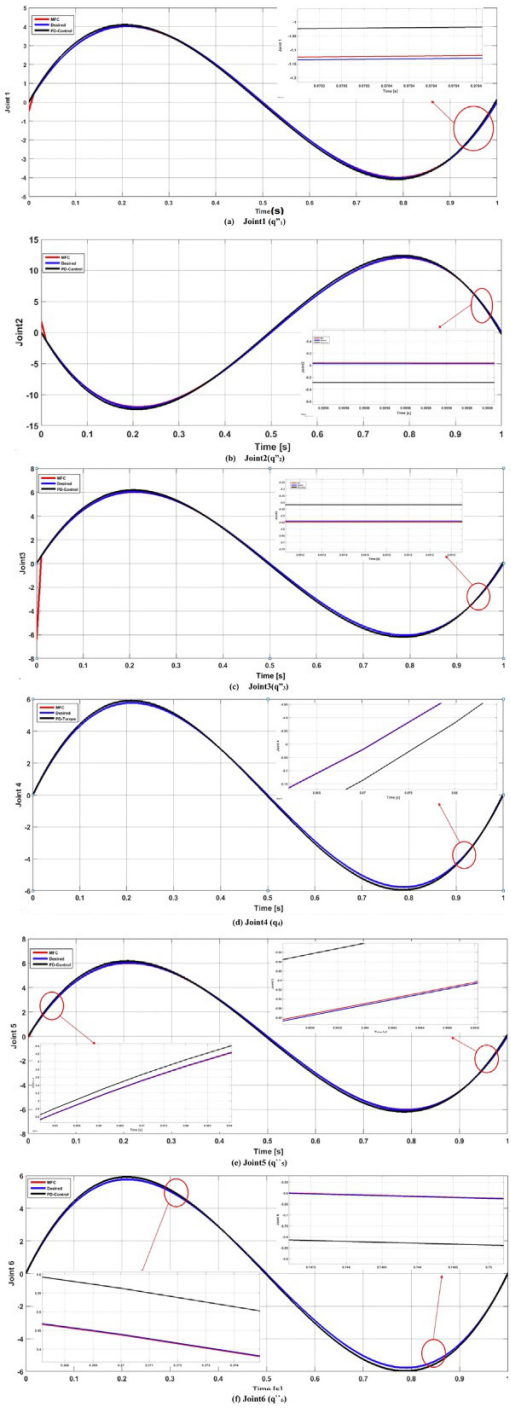

Figure 5: Time evolutions of the accelerations of the manipulator compared to the desired trajectory (Radians vs. Seconds).

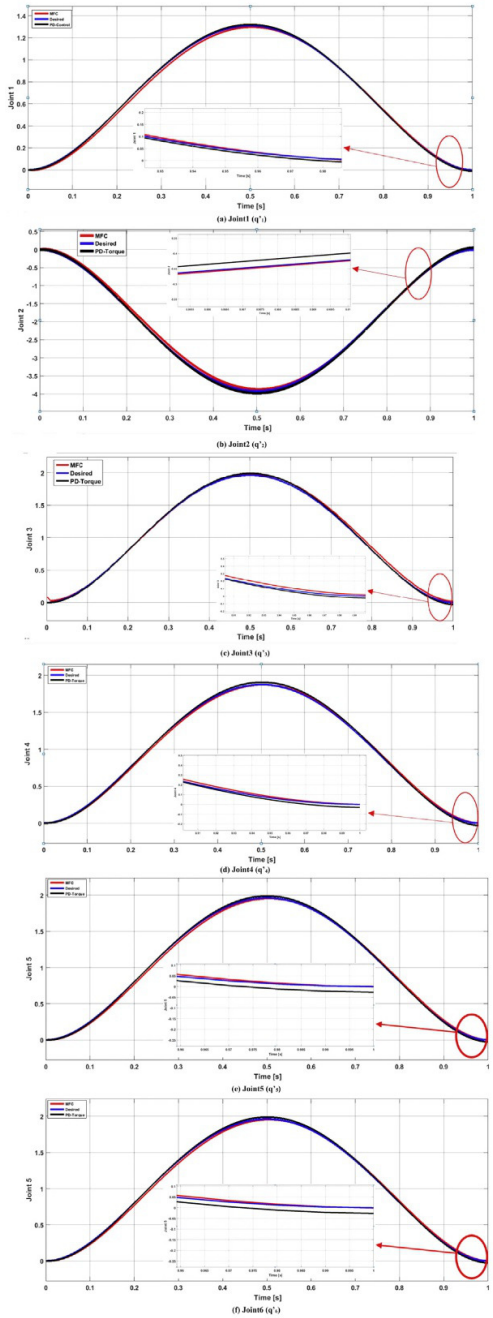

Figure 6: Time evolutions of the velocity of the manipulator compared to the desired trajectory (Radians vs. Seconds).

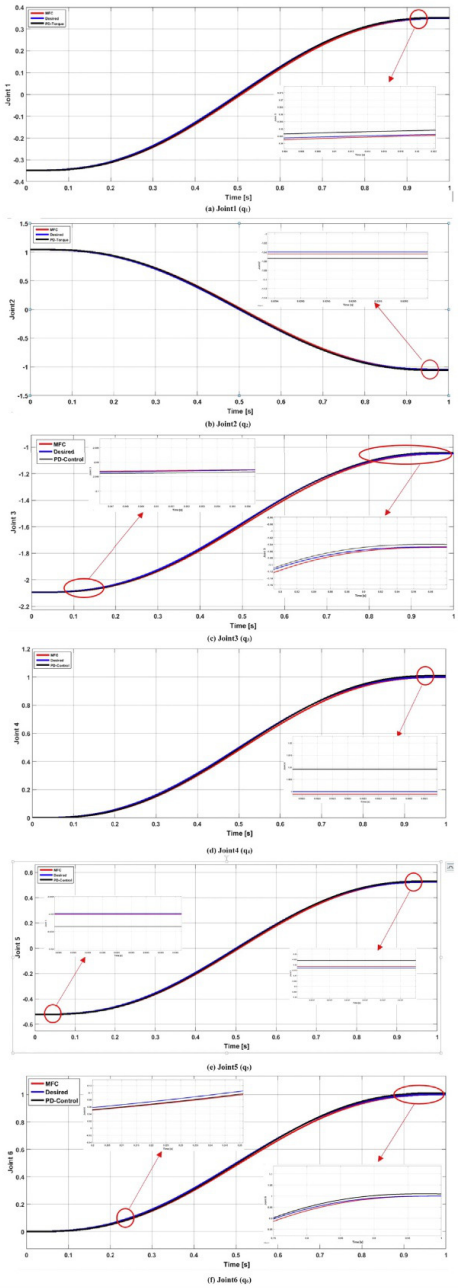

Figure 7: Time evolutions of the position of the manipulator compared to the desired trajectory (Radians vs. Seconds).

The errors between MFC controller trajectory and desired trajectory are shown in Fig. 8, it is shown from the figure that the acceleration tracking error converge to zero asymptotically. In Fig. 9-c, -d and -f, the error is equal to zero, which demonstrate the stability of proposed technique.

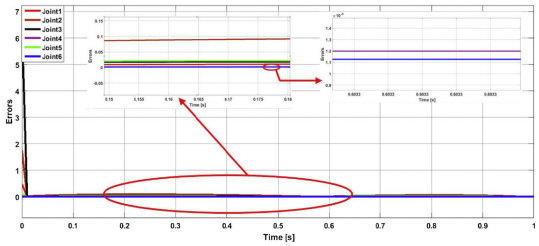

Figure 8: Trajectory tracking error (Radians vs. Seconds).

Computed torque controller (PD-Torque) algorithm is simple, but it requires high accuracy in system model and parameters [39]. Matrices and vectors M(q); C(q); D(q); and g(q) are required available and exact. It can be seen from the experimental results of Figs. 5–7 this controller gives good results and the trajectories fol- low the desired tracks. Nevertheless, the errors among the desired trajectory with the proposed controller and the conventional PD- Torque controller illustrated in Fig. 9 show that the errors with the proposed model free control are well below of those the PD-Torque controller, which demonstrate that the suggested controller can achieve superior tracking performance. In summary, through the simulations results, we have proved the efficiency of the proposed controller and we can conclude that the suggested controller is able to provide outstanding tracking performance robustly. Futures works will deal with a concrete.

5 Conclusions

The objective of this paper was to demonstrate the capability of the model free controller (MFC) to be able to command a highly nonlinear system. This controller is independently designed and not need any information about the system. The proposed model free controller MFC was used to control the PUMA 560 Robot, which is well-known industrial robot with six degrees of freedom. These systems are nonlinear, time varying, and dynamically cou- pled. According to the simulations results the proposed technique, is highly effective in controlling the robot arm. The free model MFC controller suggested in this research work were compared with an industrial used method, which is computed torque controller (PD- Torque), the results obtained have confirmed effectiveness of the studied technique.

To conclude, the model-free based control methods are easy to implement, as they do not require a precise model of the system and with simulations results we have proved the robustness and good performances of the proposed method with high tracking precision, this shows its high potential for application in industry.

References

[1] C.Y. Kuo, C.L. Yang, C. Margolin, Optimal controller design for nonlinear systems, IEE Proc. Control Theory Appl. 145 (1) (1998) 97–105.

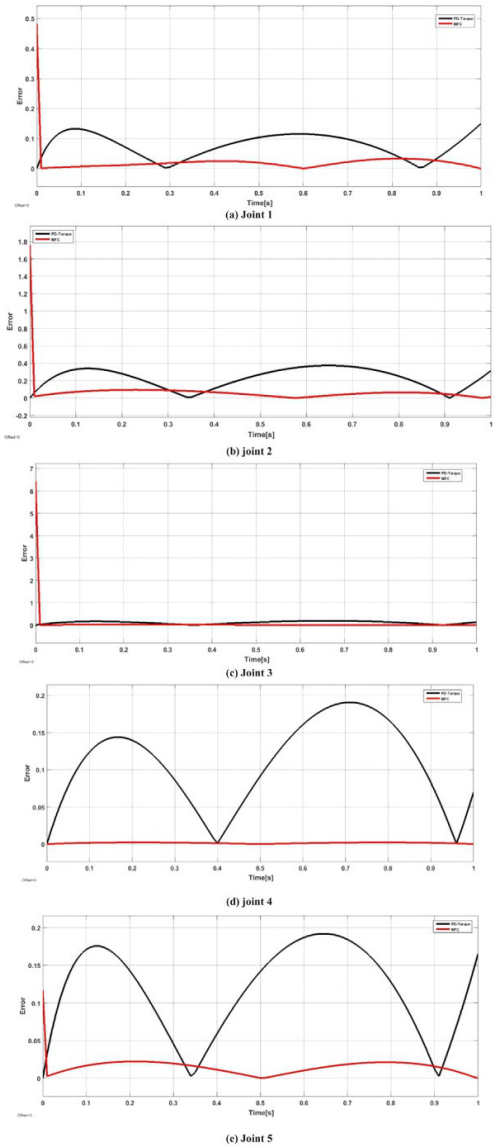

Figure 9: Fig. 9. Computed PD and MFC error (Radians vs. Seconds).

[2] M.W. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, John Wiley & Sons, New York, 2006.

[3] P.J. Antsaklis, K.M. Passino (Eds.), An Introduction to Intelligent and Autonomous Control, Kluwer Academic Publishers, Norwell, MA, 1993.

[4] M. Fliess, C. Join, Model-free control and intelligent PID controllers: towards a possible trivialization of nonlinear control, in: Proceeding of 15th IFAC Symp. System Identif., Saint-Malo, 2009.

[5] F. Lafont, J. Balmat, N. Passel, M. Fliess, A model-free control strategy for an experimental greenhouse with an application to fault accommodation, Comput. Electron. Agric. 110 (2015) 139–149.

[6] B.D.' Andrea-Novel, L. Menhour, M. Fliess, H. Mounier, Some remarks on wheeled autonomous vehicles and the evolution of their control design, Proceedings of the 9th IFAC Symposium on Intelligent Autonomous Vehicles (2016) 1–6.

[7] F.J. Carrillo, F. Rotella, Some contributions to estimation for mode-free control, in: In Proceedings of the 17th IFAC Symposium on System Identification, Beijing, China, 2015, pp. 19–21.

[8] H. Abouaïssa, M. Fliess, C. Join, On ramp metering: towards a better understanding of alinea via model-free control, Int. J. Control IJC 90 (5) (2017) 1018–1026.

[9] F. Piltan, M.H. Yarmahmoudi, M. Shamsodini, E. Mazlomian, A. Hosainpour, PUMA-560 robot manipulator position computed torque control methods using Matlab/Simulink and their integration into graduate nonlinear control and Matlab courses, Int. J. Rob. Automat. 3 (3) (2012) 167–3191.

[10] S.N. Cubero, Blind search inverse kinematics for controlling all types of serial-link robot arms, in: Mechatronics and Machine Vision in Practice, Springer, Berlin Heidelberg, 2008.

[11] W. Khalil, Dynamic modeling of robots using newton-euler formulation, in: Informatics in Control, Automation and Robotics, Springer, Berlin Heidelberg, 2011, pp. 3–20.

[12] Y. Li, Q. Xu, Adaptive sliding mode control with perturbation estimation and PID sliding surface for motion tracking of a piezo-driven micromanipulator, IEEE Trans. Control Syst. Technol. 18 (2010) 798–810, No. 4.

[13] F. Piltan, N. Sulaiman, M.H. Marhaban, A. Nowzary, M. Tohidian, Design of FPGA based sliding mode controller for robot manipulator, Int. J. Rob. Autom. 2 (3) (2011) 183–204.

[14] F. Piltan, M.R. Aghayari, M. Rashidian, A. Shamsodini, New estimate sliding mode fuzzy controller for robotic manipulator, Int. J. Robot. Autom. 3 (1) (2012) 45–58.

[15] S. Manzoor, R. ul Islam, A. Khalid, A. Samad, J. Iqbal, An opensource multi-DOF articulated robotic educational platform for autonomous object manipulation, Rob. Comput. Integr. Manuf. 30 (3) (2014) 351–362.

[16] S. Ajwad, M. Ullah, K. Baizid, J. Iqbal, A comprehensive state-of-the-art on control of industrial articulated robots, J. Balkan Tribological Assoc. 20 (2014) 499–521.

[17] M. Mirzadeh, G.H. Ahrami, M. Haghighi, A. Darveshi, S. Khezri, Intelligent model-reference method to control of industrial robot arm, Int. J. u- and e- Serv. Sci. Technol. 8 (2) (2015) 71, http://dx.doi.org/10.14257/ijunesst.2015.8. 2.07-80.

[18] U. Gogoi, Model Predictive Control of a Two Link Flexible Manipulator. Department of Electrical Engineering, National Institute of Technology, Rourkela, 2015.

[19] K.J. Astrom, T.H. agglund, PID Controllers: Theory, Design and Tuning, 2nd ed., Instrument Society of America, Research Triangle Park, NC, 1995.

[20] Gy. Mester, Intelligent mobil robot control in unknown environments, in: J.A. Tenreiro Machado, B. Pátkai, I.J. Rudas (Eds.), Intelligent Engineering Systems and Computational Cybernetics, Springer, Dodrecht, 2009, pp. 15–26.

[21] S. Heidari, F. Piltan, M. Shamsodini, K. Heidari, S. Zahmatkesh, Design new nonlinear controller with parallel fuzzy inference system compensator to control of continuum robot manipulator, Int. J. Control. Autom. 6 (4) (2013) 115–134.

[22] G. Bosque, I. Del Campo, J. Echanobe, Fuzzy Systems, Neural Networks and Neuro-Fuzzy Systems: A Vision on Their Hardware Implementation and Platforms Over Two Decades", 32, Engineering Applications of Artificial Intelligence, Elsevier Ltd, 2014, pp. 283–331, June.

[23] He Wei, Shuzhi Sam, Yanan Ge, Efe Chew Li, Yee Sien Ng, Neural network control of a rehabilitation robot by state and output feedback, J. Intell. Rob. Syst. 80 (1) (2015) 15–31.

[24] N. Thomas, D.P. Poongodi, Position control of DC motor using genetic algorithm based PID controller, Proceedings of the World Congress on Engineering 2 (2009) 1–3.

[25] T. Slavov, O. Roeva, Application of genetic algorithm to tuning a PID controller for glucose concentration control, WSEAS Trans. Syst. 11 (7) (2012) 223–233.

[26] M. Fliess, Commande sans modèle et commande à modèle restreint, eSTA 5 (2008) 1–23.

[27] M. Fliess, C. Join, M. Mboup, H. Sira-Ramirez, Vers une commande multivariable sans modèle, in: Proc. Conférence internationale francophone d'automatique (CIFA'06), 2006 http://hal.inria.fr/inria-00001139/fr/S.

[28] C. Join, J. Masse, M. Fliess, Commande sans modèle pour l'alimentation de moteurs: résultats préliminaires et comparaisons, In Proc. 2e Journées Identification et Modélisation Expérimentale (JIME'06), Poitiers (2006), URL http://hal.inria.fr/inria-00096695/fr/S.

[29] M. Mboup, C. Join, M. Fliess, A revised look at numerical differentiation with an application to nonlinear feedback control, 15th Mediterranean Conference on Control and Automation (MED'07) (2007), June URL/ http://hal.inria.fr/ inria-00142588/fr/S.

[30] M. Mboup, C. Join, M. Fliess, Numerical differentiation with annihilators in noisy environment, Numer. Algorithms 50 (4) (2008) 439–467, URL http://hal. inria.fr/inria-00319240/fr/S.

[31] M. Fliess, C. Join, H. Sira-Ramirez, Non-linear estimation is easy, Int. J. Modell. Identif. Control (IJMIC) 4 (1) (2018) 12–27, URL: http://hal.inria.fr/inria- 00158855/fr/S.2008.

[32] B. Armstrong, O. Khatib, J. Burdick, The explicit dynamic model and inertial parameters of the PUMA 560 arm, in: IEEE Conference, 2002, pp. 510–518.

[33] P.I. Corke, B. Armstrong-Helouvry, A search for consensus among model parameters reported for the PUMA 560 robot, in: IEEE Conference, 2002, pp. 1608–1613.

[34] J.G. Zeigler, et Nathaniel, B. Nichols, Optimum setting for automatic controllers, IEEE ASME 64 (1942) 759–768.

[35] H. Thabet, M. Ayadi, F. Rotella, Towards an ultra-local model of two-tank- system, in: Int. J. Dynamic Control, Springer, 2016.

[36] B.J.M.S. Join, C.M. Fliess, S. Rechdaoui-Guérin, S. Azimi, V. Rocher, A simple and efficient feedback control strategy for wastewater denitrification, 20th IFAC World Congress, IFAC (2017).

[37] H. Sira-Ramìrez, C. Garcià, A.J. Cortès-Romero, Luviano-Juárez, Algebraic Identification and Estimation Methods in Feedback Control Systems, Wiley, 2014.

[38] L. Žlajpah, MATLAB/simulink for robotics education and research, in: IEEE Int. Conf. on Robotics and Automation (Workshop), 2014.

[39] N.T. Duy, M. Seegar, J. Peters, Computed Torque Control with Nonparametric Regression Models, Proceeding of IEEE Conference on American Control, 2008, pp. 212–217.