Effectiveness of the *DIDIM* method with respect to the usual *CLOE* method. Application to the dynamic parameters identification of an industrial robot.

Anthony JUBIEN
IRCCyN (Institut de Recherche en Communications et Cybernétique de Nantes)
ONERA (the French Aerospace Lab)
Nantes (France)
anthony.jubien@irccyn.ec-nantes.fr

Maxime GAUTIER
L'UNAM
(L'Université de Nantes Angers le Mans)
IRCCyN
Nantes (France)
maxime.gautier@irccyn.ec-nantes.fr

Alexandre JANOT
ONERA
Toulouse (France)
Alexandre.janot@onera.fr

***Abstract*— The Usual Closed Loop Output Error (*CLOE*) method for dynamic parameters identification of robots has several drawbacks: slow convergence, sensitivity to initial conditions and the calculation of significant parameters is not easy-to-run. Recently a new *CLOE* method called as *DIDIM* for Direct and Inverse Identification Model needing only actual forces/torques data was validated on rigid robots. This method avoids the drawbacks of the usual *CLOE* method. With the *DIDIM* method, the optimal parameters minimize the 2-norm of the error between the actual forces/torques and the simulated ones. It is based on a closed-loop simulation of the robot using the direct dynamic model, the same structure of control-law and the same reference trajectory for both the actual robot and the simulated one. The *DIDIM* method simplifies dramatically the non-linear Least Squares problem by using the Inverse Dynamic Model in order to obtain an analytical expression of the simulated forces/torques which are linear in the parameters. This explains why the *DIDIM* method has a fast convergence. In this paper, the *DIDIM* method is compared with the usual *CLOE* method which uses the actual positions as output. Experiments are performed on a 6 degrees of freedom robot Stäubli TX40.**

***Keywords*— Identification, output error method, non linear least squares, robot, dynamic parameters.**

I. INTRODUCTION

Accurate dynamic robot models are needed to control and simulate their motions with precision and reliability. Identification of robots has been widely investigated in the last decades. The usual identification process is based on the Inverse Dynamic Model (*IDIM*) and the ordinary or weighted Least Squares (*LS*) estimation. This method, called *IDIM-LS* (Inverse Dynamic Identification Model with Least Squares), has been performed on several prototypes and industrial robots with good results [1]. This method needs the measurement of forces/torques and of positions. A well-tuned derivative band-pass filtering is necessary to estimate the velocities and accelerations and to obtain good results.

Recently, a new identification method, called *DIDIM*, needing only actual forces/torques data was validated on rigid robots [2]. It is a *CLOE* method which uses the forces/torques as output. The optimal parameters minimize the 2-norm error between the actual forces/torques and the simulated ones. A closed-loop simulation of the robot using the Direct Dynamic Model (*DDM*) assuming the same control law and the same reference trajectory for the actual and simulated robots is performed at each iteration of the algorithm. The gains of the simulated control law are updated according to the current estimates in order to simplify the non-linear *LS* problem: the simulated forces/torques is obtained with the Inverse Dynamic Model (*IDM*). This method is not sensitive to the initialization, converges in a few iterations and allows knowing the relevance of identified parameters.

To analyze the performance and the effectiveness of *DIDIM* method, a comparison with the usual *CLOE* method [3][4][5] is carried out. The usual *CLOE* method has several drawbacks compared with *DIDIM* method: slow convergence, sensitivity to initial conditions and difficult computation of the significant identified parameters. The optimal parameters minimize the 2-norm of the error between the actual positions and the simulated ones. Because of the number of parameters in the case of multi degrees of freedom (*dof*) robot (greater than 10), it is relevant to use non linear programming algorithms which do not need to calculate numerical derivative. As a consequence, in this paper, the non-linear *LS* problem is solved by using the Nelder-Mead Simplex algorithm [6]. The computation of the relevance of the identified parameters is not performed here.

This paper is divided into 7 sections. Section II describes the dynamic modeling. Section III presents the usual identification method called *IDIM-LS* while section IV presents the *DIDIM* method. Section V presents the usual *CLOE* method. Section VI is devoted to the experimental identification of the 6 *dof* TX40 robot. Finally section VII gives the conclusion.

978-1-4673-5769-2/13/$31.00 ©2013 IEEE

II. MODELING

A. Modified Denavit and Hartenberg notation

The kinematics of serial robot is defined using the Modified Denavit and Hartenberg (*MDH*) notation [7]. In this notation, the link j fixed frame is defined such that: the z_j axis is taken along joint j axis; the x_j axis is along the common normal between z_j and z_{j+1} ; α_j and d_j parameterize the angle and distance between z_{j-1} and z_j along x_{j-1} , respectively; θ_j and r_j parameterize the angle and distance between x_{j-1} and x_j along z_j , respectively. The parameter $\sigma_j = 0$ means that joint j is rotational and the parameter $\sigma_j = 1$ means that joint j is translational.

B. Inverse Dynamic Model

The *IDM* of a robot calculates the motor forces/torques τ_{idm} as a function the positions, velocities and accelerations. It can be obtained from the Newton-Euler or the Lagrangian equations [7]. It is given by the following relation:

$$\tau_{idm} = M(q)\ddot{q} + N(q,\dot{q}) \tag{1}$$

Where q , $\dot{q}$ and $\ddot{q}$ are respectively the $(n \text{ x } 1)$ vectors of joint positions, velocities and accelerations; $M(q)$ is the $(n \text{ x n})$ robot inertia matrix; $N(q,\dot{q})$ is the $(n \text{ x } 1)$ vector of centrifugal and frictions forces/torques. n is the number of moving links.

The choice of the modified Denavit and Hartenberg frames attached to each link allows a dynamic model that is linear in relation to a set of standard dynamic parameters χ_{st} [8]:

$$\tau_{idm} = IDM_{st}(q,\dot{q},\ddot{q})\chi_{st} \text{ with } \chi_{st} = \begin{bmatrix} \chi_{st1}^T & \chi_{st2}^T & \cdots & \chi_{stn}^T \end{bmatrix}^T \tag{2}$$

Where $IDM_{st}(q,\dot{q},\ddot{q})$ is the $(n \text{ x N}s)$ Jacobian matrix of τ_{idm} , with respect to the $(\text{N}s \text{ x } 1)$ vector χ_{st} of the standard parameters. χ_{stj} is composed of standard dynamic parameters of axis j :

$$\begin{aligned}\chi_{stj} = [&XX_j \quad XY_j \quad XZ_j \quad YY_j \quad YZ_j \quad ZZ_j \\ &MX_j \quad MY_j \quad MZ_j \quad M_j \quad Ia_j \quad Fv_j \quad Fc_j \quad \tau_{off_j}]^T\end{aligned} \tag{3}$$

Where:

- XX_j, XY_j, XZ_j, YY_j, YZ_j, ZZ_j are the six components of the robot inertia matrix of link j ; MX_j, MY_j, MZ_j are the components of the first moments of link j ; M_j is the mass of link j ; Ia_j is a total inertia moment for rotor and gears of actuator of link j ; Fv_j and Fc_j are the viscous and Coulomb friction parameters of joint j ; τ_{off_j} is an offset parameter which take account of the dissymmetry of Coulomb friction of joint j and of the motor current amplifier offset of joint j ; $Ns = 14 \text{ x } n$ is the number of standard parameters.

C. Direct Dynamic Model

The *DDM* can be obtained by writing the *IDM* equation (1) as follows:

$$M(q_{ddm},\chi)\ddot{q}_{ddm} = \tau_{ddm} - N(q_{ddm},\dot{q}_{ddm},\chi) \tag{4}$$

Where q_{ddm} , $\dot{q}_{ddm}$ and $\ddot{q}_{ddm}$ are the simulated positions, velocities and accelerations; τ_{ddm} is the input forces/torques of the *DDM*; $M(q_{ddm},\chi)$ and $N(q_{ddm},\dot{q}_{ddm},\chi)$ depend on an estimation the parameters χ .

III. *IDIM-LS* METHOD

Because of perturbations due to noise measurement and modeling errors, the actual force/torque τ differs from τ_{idm} by an error e , such that:

$$\tau = \tau_{idm} + e = IDM_{st}(q,\dot{q},\ddot{q})\chi_{st} + e \tag{5}$$

The vector $\hat{\chi}_{st}$ is the least squares (*LS*) solution of an over determined system built from the sampling of (5), while the robot is tracking exciting trajectories [9]:

$$Y = W_{st}\chi_{st} + \rho \tag{6}$$

Where: Y is the (rx1) measurement vector, W_{st} the (rxn_{st}) observation matrix, and ρ is the (rx1) vector of errors. The number of rows is $r = n * n_e$, where the number of recorded samples is n_e .

On some robots W_{st} is not a full rank matrix due to the kinematics structure of the system and the *LS* solution is not unique. Therefore it is necessary to determine the b base parameters of the robot also called a minimum set of inertial parameters [8][10]. They are obtained from the standard parameters by eliminating some of them which are regrouped to the others in linear relation. The system (6) is rewritten:

$$Y = W\chi + \rho \tag{7}$$

Where a subset W of b independent columns of W_{st} is calculated, which defines the vector χ of b base parameters with $b \le n_{st}$.

Standard deviations $\sigma_{\hat{\chi}_i}$, are estimated assuming that W is a deterministic matrix and ρ , is a zero-mean additive independent Gaussian noise, with a covariance matrix [11]:

$$C_{\rho\rho} = E(\rho\rho^{\mathrm{T}}) = \sigma_\rho^2 I_r \tag{8}$$

Where E is the expectation operator and I_r, the $(r \times r)$ identity matrix. An unbiased estimation of the standard deviation σ_ρ is the following:

$$\hat{\sigma}_\rho^2 = \|Y - W\hat{\chi}\|^2 / (r-b) \tag{9}$$

The covariance matrix of the estimation error is given by:

$$C_{\hat{\chi}\hat{\chi}} = E[(\chi - \hat{\chi})(\chi - \hat{\chi})^{\mathrm{T}}] = \hat{\sigma}_\rho^2 (W^{\mathrm{T}} W)^{-1} \tag{10}$$

The relative standard deviation $\%\sigma_{\hat{\chi}_{ri}}$ is given by:

$$\%\sigma_{\hat{\chi}_{ri}} = 100\sigma_{\hat{\chi}_i} / |\hat{\chi}_i|, \text{ for } |\hat{\chi}_i| \neq 0 \tag{11}$$

Where $\sigma_{\hat{\chi}_i}^2 = C_{\hat{\chi}\hat{\chi}}(i,i)$ is the i$^{\text{th}}$ diagonal coefficient of $C_{\hat{\chi}\hat{\chi}}$.

Calculating the *LS* solution of (7) from perturbated data in W and Y may lead to bias if W is correlated to ρ. Then, it is essential to filter data in Y and W before computing the *LS* solution. Velocities and accelerations are estimated by means of a band-pass filtering of the positions. To eliminate high frequency noises and torque ripples, a parallel decimation (decimate filter) is performed on Y and on each column of W. More details about data filtering can be found in [11] and [12].

IV. DIDIM: Direct And Inverse Dynamic Identification Model method

In this section, the method is briefly recalled. A complete presentation can be found in [2]. *DIDIM* is a *CLOE* method requiring only forces/torques data (see Fig 1). The output $y = \tau$, is the actual joint forces/torques τ, and the simulated output $y_s = \tau_{idm}$, is the simulated joint forces/torques. The signal $q_{ddm}(\chi, t)$ is the result of the integration of the linear implicit differential equation (4) (with robot simulated in closed loop with a tracking trajectory on q_r). The optimal solution $\hat{\chi}$ minimizes the following quadratic criterion:

$$J(\chi) = \|Y - Y_S\|^2 \tag{12}$$

Where $Y(\tau)$ and $Y_S(\tau_{idm})$ are vectors obtained by filtering and down sampling the vectors of samples of the actual forces/torques τ, and of the simulated forces/torques τ_{idm}, respectively.

This non-linear *LS* problem is solved by the Gauss-Newton regression. It is based on a Taylor series expansion of y_s, at a current estimate $\hat{\chi}^k$. Because the closed loop control keeps the same performances for the actual and for the simulated robot in relation to the estimation of parameters, the simulated positions, velocities and accelerations have little dependence on χ such as:

$$\left(q_{ddm}(\hat{\chi}^k), \dot{q}_{ddm}(\hat{\chi}^k), \ddot{q}_{ddm}(\hat{\chi}^k)\right) \simeq (q, \dot{q}, \ddot{q}) \tag{13}$$

Then the jacobian matrix can be approximated by:

$$\begin{aligned}\left(\partial(y_S)/\partial\chi\right)_{\hat{\chi}^k} &\approx IDM\left(q_{ddm}(\hat{\chi}^k), \dot{q}_{ddm}(\hat{\chi}^k), \ddot{q}_{ddm}(\hat{\chi}^k)\right) \\ &\approx IDM(q, \dot{q}, \ddot{q})\end{aligned} \tag{14}$$

Taking the approximation (14) of the jacobian matrix into the Taylor series expansion, it becomes:

$$y = \tau = IDM\left(q_{ddm}(\hat{\chi}^k), \dot{q}_{ddm}(\hat{\chi}^k), \ddot{q}_{ddm}(\hat{\chi}^k)\right)\chi^{k+1} + (o+e) \tag{15}$$

This is the Inverse Dynamic Identification Model, where $(q, \dot{q}, \ddot{q})$ are estimated with $(q_{ddm}, \dot{q}_{ddm}, \ddot{q}_{ddm})$ and $(o+e)$ is an error. An over-determined linear system is obtained after a sampling and a parallel decimation of (14):

$$Y(\tau) = W_\delta\left(q_{ddm}, \dot{q}_{ddm}, \ddot{q}_{ddm}, \hat{\chi}^k\right)\chi + \rho \tag{16}$$

The *LS* solution of (16) calculates $\hat{\chi}^{k+1}$, at iteration $k+1$. This process is iterated until: $\left(\|\rho_{k+1}\| - \|\rho_k\|\right) / \|\rho_k\| \leq \mathrm{tol}_1$.

Where tol_1 is a value ideally chosen to be a small number to get fast convergence with good accuracy.

A gains updating on the controller is performed at each iteration of algorithm to keep the performances between the simulated closed-loop and the actual one. It allows the algorithm is robust to initialization. The initialization proposed is the following:

$$\hat{\chi}^0 = 0 \text{ except for } Ia_j^0 = 1, j = 1, n \tag{17}$$

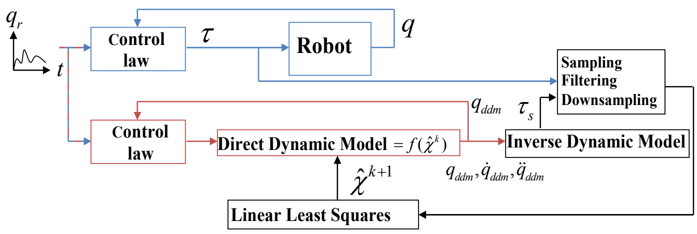

Fig 1. Scheme of *DIDIM* method

V. Usual CLOE method

In the usual CLOE method, the output is the actual joint positions q, and the simulated output is the simulated joint positions q_{ddm} (see Fig 2).

The closed-loop is preferable to open-loop because open-loop simulation is very sensitive to the initial state conditions and to the errors in algorithms which solve the differential equations. The signal $q_{ddm}(\chi, t)$ stay the result of the integration of the linear implicit differential equation (4) with robot simulated in closed loop.

Due to the number of parameters in case of multi *dof* robot, it is preferable to use non linear programming algorithms which don't need to calculate numerical derivative like Gauss-Newton method, then in this paper, the non-linear *LS* problem is solved by using the Nelder-Mead Simplex algorithm [6]. Only the *b* base parameters are presents on the *DDM* of robot.

Because the chosen non linear programming algorithm is sensitive to initialization, initialization (17) is not relevant. However the Nelder-Mead Simplex algorithm is less sensitive to initial conditions than other non linear programming algorithm (classical Gauss-Newton algorithm or Levenberg-Marquardts algorithm for example). Only initial inertias must be closes to the actual ones, the others parameters can be initialized at zero values:

$$\hat{\chi}^0 = 0 \text{ except for } Ia_j^0 = Ia_j^{ap}, j = 1,n \tag{18}$$

Where Ia_j^{ap} can be a *CAD* value or a robot manufacturer's value or the rotor inertia value of the motor of joint j.

The optimal parameters minimizes the following quadratic criterion:

$$J(\chi) = \|q - q_{ddm}\|^2 \tag{19}$$

Where q and q_{ddm} are the actual positions, and of the simulated positions, respectively. χ is composed of base parameters of robot.

This process is iterated until:

$$\left(\|\varepsilon_{k+1}\| - \|\varepsilon_k\|\right) / \|\varepsilon_k\| \le \text{tol}_2 \text{ with } \varepsilon_k = \|q_k - q_{ddmk}\| \tag{20}$$

Where tol_2 is a value ideally chosen to be a small number to get good accuracy.

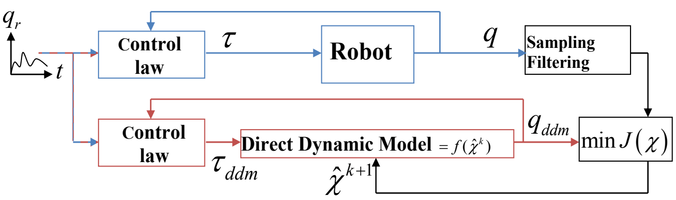

Fig 2. Scheme of usual CLOE method

VI. Experimental Validation

A. Experimental setup

The Staubli TX40 (see Fig 3) robot has a serial structure with 6 rotational joints. Its kinematics is defined using the *MDH* notation decribed in section II-A. The geometric parameters defining the robot frames are given in Table I.

Since all the joints are rotational θ_j equals the position variable q_j of joint *j*, except for joint two ($\theta_2 = q_2 - \pi/2$), joint three ($\theta_3 = q_3 + \pi/2$) and joint 6 ($\theta_6 = q_6 + \pi$), as shown in Table 1. All mechanical variables are given in *SI* unit in joint side. The robot is characterized by a coupling effect between the joint 5 and 6 [13]. It is described in the appendix. The TX40 has $N_s = 86$ standard dynamic parameters given by the $14x6$ usual standard parameters, plus f_{vm6} and f_{cm6}. The number of base parameters after model simplification is 60. The torques are calculated using the relation $\tau = v_\tau g_\tau$ with v_τ is the (6x6) matrix of the actual current references of the current amplifiers and g_τ is the (6x1) vector of the joint drive gain.

TABLE I MDH PARAMETERS OF THE TX40 ROBOT

j	α_j	d_j	θ_j	r_j
1	0	0	q_1	0
2	$-\pi/2$	0	$q_2 - \pi/2$	0
3	0	$d_3 (= 0.225m)$	$q_3 + \pi/2$	$rl_3 (= 0.035m)$
4	$\pi/2$	0	q_4	$rl_4 (= 0.225m)$
5	$-\pi/2$	0	q_5	0
6	$\pi/2$	0	$q_6 + \pi$	0

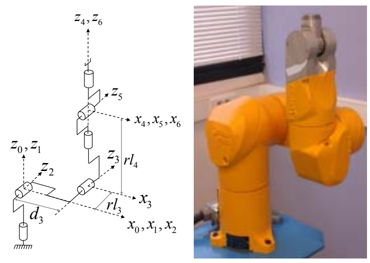

Fig 3. Link frame of the TX40 robot and picture of robot

B. DIDIM method

The identification with DIDIM method is performed. The algorithm is initialized with (17). The cut off frequency of decimate filter is fixed at 20Hz. At each iteration of the algorithm, the gains of controller are updating. The algorithm converges after 3 iterations (3 simulation of the *DDM*) in 18 (second). The identified values of base parameters are given in Table II.

C. Usual CLOE method

The identification with usual *CLOE* method is carried out. The *fminsearch* function of software *Matlab* is used (without modification). The algorithm is initialized with *a priori* rotor inertia of the motors and zero value for the others parameters:

$$\begin{aligned} &Ia_1^0 = Ia_2^0 = 0.362, Ia_3^0 = 0.081, Ia_4^0 = 0.014, \\ &Ia_5^0 = 0.025, Ia_6^0 = 0.0062 \; (Kg.m^2) \end{aligned} \tag{21}$$

The algorithm converges after 4230 simulations of algorithm. The number of simulation of *DDM* is 9428. Approximately 16 (hours) are necessary to identify all parameters on a dual-core 2(*Ghz*) CPU. The identified values of base parameters are given in Table II.

TABLE II IDENTIFIED PARAMETERS WITH DIDIM AND USUAL CLOE METHOD (RED: IDENTIFIED PARAMETERS WITH $\sigma_{\hat{\chi}_n^3}$ *(%)* < 20%)

Method	*DIDIM*		*CLOE*	Method	*DIDIM*		*CLOE*
Parameters	$\hat{\chi}^3$	$\sigma_{\hat{\chi}_r^3}$ *(%)*	Parameters	$\hat{\chi}^3$	$\sigma_{\hat{\chi}_r^3}$ *(%)*	$\sigma_{\hat{\chi}_r^3}$ *(%)*	$\hat{\chi}$
ZZ_{1R}	**1.24**	**1.1**	**1.19**	MX_4	-0.0276	22	2.38.10-5
Fv_1	**8.05**	**0.54**	**10.1**	MY_{4R}	-0.00556	142	0.000111
Fc_1	**7.07**	**1.8**	**8.16**	Ia_4	**0.0292**	**13**	**0.0219**
Off_1	0.296	23	0.405	Fv_4	**1.09**	**3.1**	**1.20**
XX_{2R}	**-0.473**	**2.2**	**-0.418**	Fc_4	**2.52**	**5.0**	**2.33**
XY_2	0.00320	157	0.0084	Off_4	-0.0825	78	-0.984
XZ_{2R}	**-0.156**	**3.5**	**-0.178**	XX_{5R}	0.00643	53	0.0222
YZ_2	-0.00618	68	0.161	XY_5	0.000369	433	-0.0013
ZZ_{2R}	**1.08**	**0.89**	**1.01**	XZ_5	0.00110	155	-0.000181
MX_{2R}	**2.20**	**2.0**	**2.12**	YZ_5	0.00215	73	0.0014
MY_2	0.117	33	-0.0469	ZZ_{5R}	0.00227	120	5.17.10-6
Fv_2	**5.55**	**0.87**	**5.77**	MX_5	-0.00496	110	0.000503
Fc_2	**8.16**	**1.4**	**7.49**	MY_{5R}	**-0.00322**	**15**	**-0.00173**
Off_2	0.858	62	-10.7	Ia_5	**0.00404**	**11**	**0.00483**
XX_{3R}	**0.135**	**7.7**	**0.112**	Fv_5	**1.81**	**2.3**	**1.90**
XY_3	0.00377	150	-0.0011	Fc_5	**3.10**	**3.5**	**3.36**
XZ_3	-0.00743	88	0.0235	Off_5	0.00264	267	-0.234
YZ_3	0.00839	48	0.0020	XX_{6R}	-0.00365	38	0.0028
ZZ_{3R}	**0.0117**	**7.2**	**0.0102**	XY_6	0.000453	159	0.000265
MX_3	0.00406	61	0.0538	XZ_6	-0.00223	45	0.0018
MY_{3R}	**-0.0592**	**2.1**	**-0.0606**	YZ_6	-0.00226	40	0.0018
Ia_3	**0.00875**	**7.9**	**0.0078**	ZZ_6	0.00333	35	0.000259
Fv_3	**1.94**	**1.8**	**2.03**	MX_6	0.00142	236	0.0101
Fc_3	**6.45**	**1.8**	**5.96**	MY_6	0.00629	50	0.0165
Off_3	0.00322	636	-0.152	Ia_6	**0.00918**	**16**	**0.0065**
XX_{4R}	0.00381	129	0.0031	Fv_6	**0.637**	**3.4**	**0.760**
XY_4	-0.00386	57	0.0029	Fc_6	**2.30**	**5.1**	**2.30**
XZ_4	-0.00706	38	-0.0019	$Fvm6$	**0.593**	**2.8**	**0.662**
YZ_4	-0.00689	43	0.0031	$Fcm6$	**2.01**	**4.8**	**1.98**
ZZ_{4R}	0.000450	754	1.60.10-7	Off_6	0.173	38	0.0285

D. Discussion

A study of the effectiveness of the *DIDIM* method compared to the usual *CLOE* method is performed. It is recalled that the first method uses the joint forces/torques as output while the second method uses the joint positions as output.

The run-time of *DIDIM* method is low, 18 (seconds), to identify a 6 *dof* TX40 robot. The algorithm requires only 3 iterations and 3 simulations of *DDM*. This is because the *DIDIM* method simplifies dramatically the non-linear least squares problem using the *IDM* in order to obtain an analytical expression of the simulated forces/torques which are linear in the parameters. The usual *CLOE* method needs more computation-time 16 (hours) which is 3000 times longer than the *DIDIM* method.

The non-linear *LS* problem for the usual *CLOE* method is solved by using the Nelder-Mead simplex algorithm. It is sensitive to the initialization compared with *DIDIM* and it only converges to a local optimum and no to a global optimum if the initial conditions are not good [14]. However, if only initial inertias must be closes to the actual ones, the others parameters can be initialized at zero values.

The calculation of standard deviations of base parameters allows determining their significance. With *DIDIM* method, the calculation is easy because this method is based on linear *LS*. But, with the usual *CLOE* method, the calculation of standard deviations is very involved. No information is available on significance of identified parameters. The computation of sensitivity functions for each base parameter could be a solution to resolve this problem. However it needs some computation time and it is difficult to implement.

In *DIDIM* method, the sensitivity functions are approximated by the *IDM.* So it is preferable to minimize the quadratic error between the actual and measured torque than to minimize the quadratic error between the actual and measured position.

Only the *DIDIM* base parameters well identified ($\sigma_{\hat{\chi}_{ri}}^{3}(\%)$ < 20%) are close than identified with *CLOE* method. The other parameters are pretty much different due to fact they have a little impact on positions and forces/torques (see Table II, parameters in red).

This study illustrates the great interest of *DIDIM* method relative to usual *CLOE* method

VII. CONCLUSION

In this paper a study of the effectiveness of *DIDIM* method compared to a usual *CLOE* method was performed. Due to many differences between these methods, *DIDIM* has many advantages:

-*DIDIM* reduces the computation time significantly,

-it is robust to initialization compared to usual *CLOE* method,

-it avoids the measurement of joint positions,

-it computes simply the relevance of identified parameters.

Therefore *DIDIM* method is more efficient than usual *CLOE* method.

APPENDIX: KINEMATIC COUPLING EFFECT

The coupling between the joints 5 and 6 on the TX40 robot is the following [13]:

$$\begin{bmatrix} \dot{q}_{r5} \\ \dot{q}_{r6} \end{bmatrix} = \begin{bmatrix} K5 & 0 \\ K6 & K6 \end{bmatrix} \begin{bmatrix} \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}, \quad \begin{bmatrix} \tau_{c5} \\ \tau_{c6} \end{bmatrix} = \begin{bmatrix} K5 & K6 \\ 0 & K6 \end{bmatrix} \begin{bmatrix} \tau_{r5} \\ \tau_{r6} \end{bmatrix} \tag{22}$$

Where $\dot{q}_{rj}$ is the velocity of the rotor of motor j, $\dot{q}_j$ is the velocity of joint j, $K5$ is the transmission gain ration of axis 5 and $K6$ is the transmission gain ration of axis 6, τ_{cj} is the motor torque of joint j, taking into account the coupling effect, τ_{rj} is the electro-magnetic torque of the rotor of motor j. The coupling between joints 5 and 6 also adds the effect of the inertia of rotor 6 and new viscous and Coulomb friction parameters fvm_6 and fcm_6, to both τ_{c5} and τ_{c6}:

$$\begin{aligned} \tau_{c5} &= \tau_5 + Ia_6\ddot{q}_6 + fvm_6\dot{q}_6 + fcm_6 sign(\dot{q}_6) \\ \tau_{c6} &= \tau_6 + Ia_6\ddot{q}_5 + fvm_6\dot{q}_5 + fcm_6\left(sign(\dot{q}_5 + \dot{q}_6) - sign(\dot{q}_6)\right) \end{aligned} \tag{23}$$

$$\begin{aligned} &\text{with } \tau_j = Ia_j\ddot{q}_j + fv_j\dot{q}_j + fc_j sign(\dot{q}_j) \\ &Ia_5 = K5^2 Ja_5 + K6^2 Ja_6 \text{ and } Ia_6 = K6^2 Ja_6 \end{aligned} \tag{24}$$

Where Ja_j is the moment of inertia of rotor j.

REFERENCES

[1] J. Hollerbach, W. Khalil, et M. Gautier, « Model Identification », in *Springer Handbook of Robotics*, Springer, 2008.

[2] M. Gautier, A. Janot, et P.-O. Vandanjon, « A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics », IEEE Transactions on Control Systems Technology, vol. 21, no 2, p. 428‑444, DOI 10.1109/TCST.2012.2185697. 2012.

[3] F. J. Carrillo, A. Baysse, et A. Habbadi, « Output error identification algorithms for continuous-time systems operating in closed-loop », IFAC Symposium on System Identification, 2009, p. 408‑413.

[4] I. D. Landau, B. D. O. Anderson, et F. De Bruyne, « Closed-loop output error identification algorithms for nonlinear plants », IEEE Conference on Decision and Control, 1999, p. 606‑611.

[5] M. Gautier, A. Janot, A. Jubien, et P. O. Vandanjon, « Joint stiffness identification from only motor force/torque data », IEEE Conference on Decision and Control and European Control Conference, 2011, p. 5088‑5093.

[6] J. C. Lagarias, J. A. Reeds, M. H. Wright, et P. E. Wright, « Convergence Properties of the Nelder – Mead Simplex Method in Low Dimensions », *Society for Industrial and Applied Mathematics*, vol. 9, p. 112‑147, 1998.

[7] W. Khalil et E. Dombre, *Modeling, Identification and Control of Robots*, 3rd edition. New York: Taylor and Francis Group, 2002.

[8] M. Gautier et W. Khalil, « Direct calculation of minimum set of inertial parameters of serial robots », *IEEE Transactions on Robotics and Automation*, p. 368‑373, 1990.

[9] M. Gautier et W. Khalil, « Exciting trajectories for the identification of base inertial parameters of robots », IEEE Conference on Decision and Control, 1991, p. 494‑499.

[10] H. Mayeda, K. Yoshida, et K. Osuka, « Base parameters of manipulator dynamic models », *IEEE Transactions on Robotics and Automation*, vol. 6, n° 3, p. 312–321, 1990.

[11] M. Gautier, « Dynamic identification of robots with power model », IEEE International Conference on Robotics and Automation, 1997, vol. 3, p. 1922‑1927.

[12] M. T. Pham, M. Gautier, et P. Poignet, « Identification of joint stiffness with bandpass filtering », IEEE International Conference on Robotics and Automation, 2001, vol. 3, p. 2867‑2872.

[13] M. Gautier et S. Briot, « Global identification of drive gains parameters of robots using a known payload », IEEE International Conference on Robotics and Automation, 2012, p. 2812‑2817.

[14] E. Tohme, R. Ouvrard, J.-C. Abche, J.-C. Trigeassou, T. Poinot, et G. Mercère, « A Methodology to Enhance the Convergence of Output Error Identification Algorithms », European Control Conference, 2007, p. 5721‑5728.