

Data Set and Reference Models of EMPS

A. Janot * M. Gautier ** M. Brunot *

* ONERA, 2 Avenue Edouard Belin, 31055 Toulouse, France (e-mail:

Mathieu.Brunot@onera.fr and Alexandre.Janot@onera.fr).

** LS2N, 1 rue de la noe, 44321 Nantes, France (e-mail:

maxime.gautier@ls2n.fr)

Abstract:

The following report provides a complete description of the Electro-Mechanical Positioning System (EMPS) which is a standard configuration of a drive system for prismatic joint of robots or machine tools. First, an overview of EMPS is presented: its main components, the continuous-time inverse and direct dynamic models suitable to describe its nonlinear dynamics and the controller used to pilot it are described in details. Second, the trajectories used to excite the EMPS, the data set available in .mat file format, the identification goals and the challenges are introduced. Fourth, two source codes allowing the user to run the standard identification method adopted in robotics and simulate the EMPS are presented. The EMPS has a pure integrator, is piloted by a controller consisting of two nested loops and exhibits an asymmetrical friction. This constitutes a challenge when performing identification of such electro-mechanical systems.

1. EMPS OVERVIEW, MODELLING AND CONTROL

1.1 EMPS Overview

The EMPS illustrated Fig. 1 is a standard configuration of a drive system for prismatic joint of robots or machine tools. It is connected to a dSPACE digital control system for easy control and data acquisition using Matlab and Simulink software. Its main components are:

- A Maxon DC motor equipped with an incremental encoder. As we shall see later, this DC motor is position-controlled with a PD controller.
- A Star high-precision low-friction ball screw drive positioning unit. Its extremity is equipped with an incremental encoder that is not used in this benchmark.
- A load in translation. An accelerometer is mounted on it and is not used in this benchmark.

The measurements provided by the encoder mounted at the extremity of the ball screw and the accelerometer are not used because such measurements are not available in most industrial robots, see e.g. (Khalil and Dombre, 2004).

1.2 EMPS Modelling

The Inverse Dynamic Model (IDM) of a robot expresses the joint torque/force as a function of the joint position, velocity and acceleration, (Khalil and Dombre, 2004). From Newton's law, the IDM of EMPS is given by

$$\tau_{idm}(t) = M\ddot{q}(t) + F_v\dot{q}(t) + F_c\text{sign}(\dot{q}(t)) + offset, \quad (1)$$

where q , \dot{q} and \ddot{q} are respectively the joint position, velocity and acceleration; τ_{idm} is the joint torque/force; M is the inertia of the arm; F_v and F_c are respectively the viscous

and Coulomb frictions; *offset* is an offset of measurements of τ_{idm} . M , F_v and F_c are referred to as the *dynamic* parameters.

Equation (1) is linear in relation to the dynamic parameters,

$$\tau_{idm}(t) = [\ddot{q}(t) \quad \dot{q}(t) \quad \text{sign}(\dot{q}(t)) \quad 1] \boldsymbol{\theta} = \boldsymbol{\phi}(t)\boldsymbol{\theta}, \quad (2)$$

where $\boldsymbol{\phi}(t)$ is the (1×4) observation matrix of basis functions of the IDM and $\boldsymbol{\theta} = [M \quad F_v \quad F_c \quad offset]^T$ is the (4×1) vector of the 3 dynamic parameters plus the offset.

The Direct Dynamic Model (DDM) of a robot expresses the joint acceleration as a function of the joint torque/force, position and velocity, (Khalil and Dombre, 2004). In the case of the EMPS, we have

$$\ddot{q}(t) = \frac{\tau_{idm}(t)}{M} - \frac{F_v}{M}\dot{q}(t) - \frac{F_c}{M}\text{sign}(\dot{q}(t)) - \frac{offset}{M}. \quad (3)$$

Again, the DDM is linear in relation to a set of parameters that is a nonlinear combination of the dynamic parameters i.e.

$$\ddot{q}(t) = [\tau_{idm}(t) \quad -\dot{q}(t) \quad -\text{sign}(\dot{q}(t)) \quad -1] \boldsymbol{\theta}_{ddm} \quad (4) \\ = \boldsymbol{\phi}_{ddm}(t)\boldsymbol{\theta}_{ddm},$$

where $\boldsymbol{\phi}_{ddm}(t)$ is the (1×4) observation matrix of basis functions of the DDM and $\boldsymbol{\theta}_{ddm} = [1/M \quad F_v/M \quad F_c/M \quad offset/M]^T$ is the (4×1) vector of the nonlinear combination of the 3 dynamic parameters plus the offset. In the case of EMPS, the 3 dynamic parameters and the offset are divided by M .

In (Brunot et al., 2015), a comparison between the IDM and the DDM is made for the identification of the EMPS. However, because the DDM is usually nonlinear with respect to the dynamic parameters for multi degrees-of-freedom robots, it is rarely used (Swevers et al., 2007) and (Gautier et al., 2013).

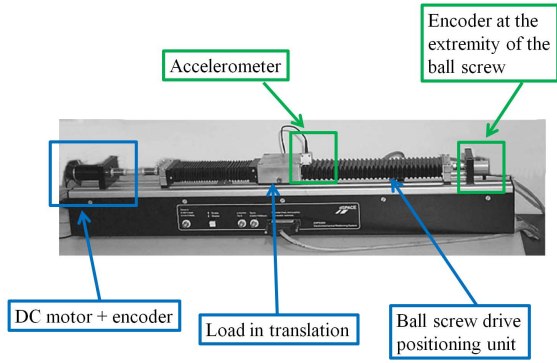


Fig. 1. EMPS prototype

Finally, the DDM can be written as a transfer-function (TF) like expression as usually done in robotics, (Gautier et al., 2013)

$$q(t) = \frac{b_0}{s(s + a_0)} (\tau_{idm}(t) - d(t)), \quad (5)$$

where $s = j\omega$ is the Laplace's variable; $b_0 = 1/M$; $a_0 = F_v/M$; $d(t) = F_c \text{sign}(\dot{q}(t)) + \text{offset}$ a perturbation encompassing the Coulomb and offset effects.

1.3 EMPS Control

As well enlightened by relation (5), the EMPS model (1) has a pure integrator and cannot, therefore, be identified in open loop. The system is thus driven by a Proportional-Derivative (PD) controller consisting of an outer loop controlling the position and an inner loop controlling the velocity. The control signal ν is given by

$$\nu(t) = k_p k_v (q_r(t) - q(t)) - k_v \dot{q}(t), \quad (6)$$

where k_p is the proportional gain and k_v is the derivative gain. With the bandwidth of the position loop equal to 20 Hz, the gains have been chosen such as $k_p = 160.18$ 1/s and $k_v = 243.45$ V/(m.s⁻¹). In Gautier et al. (2013), the authors have shown that a PD control is enough to identify the dynamic parameters of robots because an excellent tracking is not needed.

The control signal, coming from the control law, is linked to τ_{idm} by the following relation

$$\tau_{idm}(t) = g_\tau \nu(t), \quad (7)$$

where g_τ is the drive gain of the EMPS. Although the drive gain is usually given by the manufacturer, it can be identified with special tests (Gautier and Briot, 2014). In this case, it has been estimated to $g_\tau = 35.15$ N/V.

2. DESCRIPTION OF DATA SET, EXCITATION OF EMPS, IDENTIFICATION GOALS AND CHALLENGES

2.1 Excitation of EMPS

According to (Gautier, 1991), M , F_v , F_c and *offset* constitute the set of *base* parameters i.e. the set of structurally identifiable. Hence, if the EMPS is appropriately excited, then these parameters must be well identified. Of course, if M , F_v , F_c and *offset* are well excited, so are $1/M$, F_v/M , F_c/M and *offset/M*.

To excite the dynamic parameters, bang-bang accelerations are applied to the EMPS. Such trajectories are common in robotics, see (Khalil and Dombre, 2004), because they allow the user identifying the inertia (or mass), gravity and friction parameters. Indeed, inertia (or mass) is identified while the acceleration is varying (or constant) over a period and gravity and friction parameters are identified while the velocity is constant over a period with a maximum angular range. Finally, to assess the excitation of the system, the condition number of the observation is considered, (Gautier, 1997). In this case, the condition number being close to 30, the dynamic parameters can be considered as well excited.

2.2 Description of Data Set

In most applications, the available information is the measurements of the joint position denoted as q_m and the control signal denoted as ν_m . As previously mentioned, all data are measured by a dSPACE digital control system in order to ensure quality and periodicity.

The sampling frequency is 1kHz while the duration is 25s approximately. q_m is measured by an incremental encoder whose the resolution is 12500 counts per revolution. Since this encoder is working in quadrature count mode, its resolution is finally $12500 \cdot 4 = 50000$ counts per revolution. Such a resolution ensures an extremely good quality of q_m . Concerning ν_m , it is considered as the output of the controller and is directly measured by the dSPACE card.

To compute the velocity involved in the control law (6), the motor position is first filtered by a FIR filter. Hence, at time k , on has

$$\dot{q}_m(k) = \frac{q_{fm}(k) - q_{fm}(k-1)}{dt}$$

with

$$q_{fm}(k) = 0.5 \cdot q_m(k) + 0.5 \cdot q_m(k-1)$$

where dt is the sampling time (1ms here); and $q_{fm}(k)$ the filtered motor position at time k .

The **EMPS_DATA** mat file contains all the data required to identify the EMPS. The names of variables are the following:

- **qm**: the vector of motor position expressed in the load side i.e. in m ;
- **qg**: the vector of reference of motor position expressed in the load side i.e. in m ;
- **vir**: the vector of motor force expressed in the load side i.e. in N ;
- **t**: the vector of time expressed in seconds;
- **kp**: the gain of the outer loop of position in 1/s;
- **kv**: the gain of the inner loop of velocity in V/(m.s⁻¹);
- **gtau**: the drive gain in N/V.

It is important to stress that all the data were not treated and they can be, therefore, considered as *raw*. Furthermore, the input of the EMPS is the force i.e. **vir** MATLAB variable while the output is the position i.e. **qm** MATLAB variable.

To validate the estimates, cross-test validations must be executed. To run cross-test validations, the EMPS is still excited with bang-bang accelerations but pulses are also

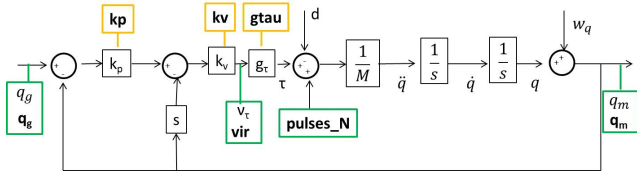


Fig. 2. Diagram of EMPS and signals

directly applied. The **EMPS_DATA_PULSE** mat file contains all the data required to execute the cross-test validations. The pulses applied to the EMPS are stored in the variable **pulses_N** given in *N*.

The complete diagram of the EMPS is illustrated in Fig. 2 where the variables of signals stored in the mat file are framed in green while the variables of gains are framed in orange. Finally w_q is the noise caused by the encoder.

2.3 Identification goals

In robotics, the main goal is to identify the physical parameters involved in the IDM i.e. M , F_v and F_c in order to develop accurate model-based controllers such as the computed torques, see (Khalil and Dombre, 2004). In addition, identifying these physical parameters allows the user having a direct physical assessment of the results.

Always in robotics, another goal is the enhancement of the mechanical or electrical design, e.g. to enhance the haptic rendering and/or backdrivability, to increase the stiffness of the structure, to lower the effect of frictions... It is interesting to notice that identifying the EMPS (or robots) to develop a *simple* controller (e.g. PID controller) is not usual because the Computer-Aided Design (CAD) values of dynamical parameters provided by manufacturers are enough to design such controllers in order to get *rough* performances.

2.4 Challenges

Despite its simplicity, it is really difficult to use common toolboxes to identify the EMPS:

First, like any electro-mechanical systems, the EMPS exhibits a pure integrator which means that it must be identified with a closed-loop process.

Second, because the controller consists of two nested loops, it cannot be written with the following form commonly adopted in toolboxes

$$\nu(t) = C(s)(q_{ref}(t) - q(t)),$$

where $C(s)$ is the transfer function of the controller; and $q_{ref}(t)$ the position reference.

Third, the IDM of EMPS includes a nonlinear effect: the friction. As we shall see later, the friction model which is first assumed to be symmetric is, in fact, asymmetric. Finally, these three features make the usual toolboxes rather ineffective if it is aimed to obtain accurate dynamic models. This will be emphasized in a later section.

3. THE STANDARD METHOD FOR ROBOT IDENTIFICATION: IDIM-LS

This section presents the standard approach employed to identify the dynamic parameters of robots termed

the *Inverse Dynamic Identification Model with Least-Squares estimation (IDIM-LS)*. Such an approach has been validated on several industrial robots and prototypes see e.g. (Gautier, 1997), (Khalil and Dombre, 2004), (Swevers et al., 2007), (Gautier et al., 2013), (Janot et al., 2014a) and the references therein.

3.1 Joint velocity and acceleration estimation

The joint velocities and accelerations have to be calculated from q_m in order to build the observation matrix ϕ as described in (Gautier, 1997). q_m is firstly filtered to obtain \hat{q} . From this filtered position, the derivatives can be calculated with finite differences. The filter type and the cut-off frequency, ω_{f_q} , are selected such as $(\hat{q}, \hat{\dot{q}}, \hat{\ddot{q}}) \approx (q, \dot{q}, \ddot{q})$ in the range $[0, \omega_{f_q}]$. The filter, which is usually a Butterworth one, is applied in both forward and reverse directions to avoid lag introduction. The rule of thumb for the cut-off frequency is $\omega_{f_q} \geq 5\omega_{dyn}$, where ω_{dyn} is the natural frequency of the highest mode of the closed-loop system. The combination of the Butterworth filter and the central differentiation is referred to as the *bandpass* filtering process.

3.2 High-frequency ripples rejection

The IDIM differs from the IDM by an error term $v(t)$, resulting of perturbations coming from measurement noise and modelling errors. The IDIM is then given by:

$$\tau(t) = \tau_{idm}(t) + v(t) = \phi(t)\theta + v(t). \quad (8)$$

In practice, the torque is perturbed by high-frequency ripples which are rejected by the controller. Those ripples are removed with a parallel lowpass filtering of each basis function at the cut-off frequency $\omega_{F_p} \geq 2\omega_{dyn}$. The choice of ω_{F_p} is involved to keep enough information while avoiding the high frequency noise. Since there is no more useful information beyond the cut-off frequency, the data are also re-sampled by keeping one sample over n_d , i.e. n_d is the decimation factor. This combination of parallel filtering and re-sampling is referred to as the *decimate* process. After data acquisition and parallel filtering, we obtain

$$\begin{aligned} \tau_{F_p}(t) &= F_p(z^{-1})\tau(t) \\ &= \phi_{F_p}(\hat{q}(t), \hat{\dot{q}}(t), \hat{\ddot{q}}(t))\theta + v_{F_p}(t), \end{aligned} \quad (9)$$

where z^{-1} is the backward shift operator and F_p is the decimate filter applied to each element of the observation matrix, the torque signal and the error term.

3.3 IDIM-LS estimates

The IDIM-LS estimates are given by

$$\hat{\theta}_{LS} = \left(\Phi_{F_p}^T \Phi_{F_p} \right)^{-1} \Phi_{F_p}^T \mathbf{y}_{F_p}, \quad (10)$$

where Φ_{F_p} is the $(N \times 4)$ observation matrix resulting from the sampling of $\phi_{F_p}(\hat{q}(t), \hat{\dot{q}}(t), \hat{\ddot{q}}(t))$; \mathbf{y}_{F_p} is the $(N \times 1)$ vector of measurements resulting from the sampling of τ_{F_p} ; and N is the number of samples obtained after sampling and decimation.

If the two following conditions hold

- $\phi_{F_p}^T \phi_{F_p}$ is full column rank,
- $\mathbb{E} \left[\phi_{F_p}^T v_{F_p} \right] = 0$,

where $\mathbb{E} [\cdot]$ denotes the mathematical expectation, then the IDIM-LS (10) are unique and asymptotically unbiased.

3.4 A source code to run the IDIM-LS method

The **Script_IDIM_LS** m file presents a simple source code allowing the user running the IDIM-LS method. The code is commented and divided into 8 main parts:

- data loading;
- parameters of filters, both Butterworth and decimate;
- construction of the vector measurements;
- data filtering;
- construction of the observation matrix;
- decimation process;
- calculation of the IDIM-LS estimates;
- plotting and displaying the results.

Once the code executed, something like that must be displayed on the MATLAB command window:

```
Parameter M1 :      95.1089      0.1085      0.1141 %
Parameter FV1 :    203.5034      1.1460      0.5631 %
Parameter FC1 :     20.3935      0.1012      0.4964 %
Parameter OF1 :     -3.1648      0.0444      1.4021 %
```

```
Absolute value of the diagonal element of R
Parameter M1 :      64.1376
Parameter FV1 :     13.9005
Parameter FC1 :     69.1706
Parameter OF1 :    157.4483
Condition number of the observation matrix: 25.9883
```

```
Error norm : 108.7569
Relative error (%): 4.0834
Deviation of error: 2.1843
```

Then, a direct comparison between the reconstructed and measured force is plotted, see Fig.3, and a cross test validation is also performed by using a data set which was not used to run the IDIM-LS method, see Fig.3. In robotics, it is common to assess the quality of the identification results by computing the relative error given by

$$rel_{err} = 100 \cdot \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|}{\|\mathbf{y}\|}, \quad (11)$$

where \mathbf{y} is the $(N \times 1)$ vector of measurements; and $\hat{\mathbf{y}}$ is the $(N \times 1)$ vector of estimation. For instance, in the case of the IDIM-LS method, one has $\hat{\mathbf{y}} = \Phi_{F_p} \hat{\boldsymbol{\theta}}_{LS}$ and $\mathbf{y} = \mathbf{y}_{F_p}$ when performing direct validations while one has $\hat{\mathbf{y}} = \Phi_{F_p}^{CTV} \hat{\boldsymbol{\theta}}_{LS}$ and $\mathbf{y} = \mathbf{y}_{F_p}$ when performing cross test validations; $\Phi_{F_p}^{CTV}$ is the $(N \times 4)$ observation matrix constructed with the set of data which was not used to run the identification process.

To execute the cross-test validation, the **Script_IDIM_LS_CTV** m file is provided. It loads the **EMPS_DATA_PULSE** mat file and reconstructs the force with the new set of data and the IDIM-LS estimates. Once the code executed, something like that must be displayed on the MATLAB command window:

```
Validating the IDIM-LS estimates with cross-test validation
Absolute value of the diagonal element of R
```

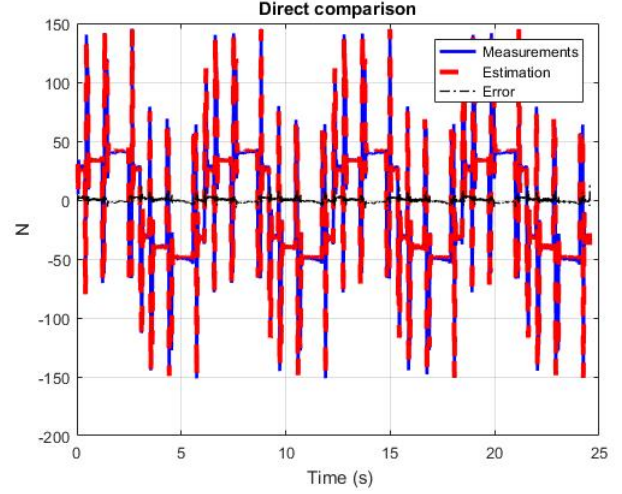


Fig. 3. Direct comparison between the reconstructed and measured forces

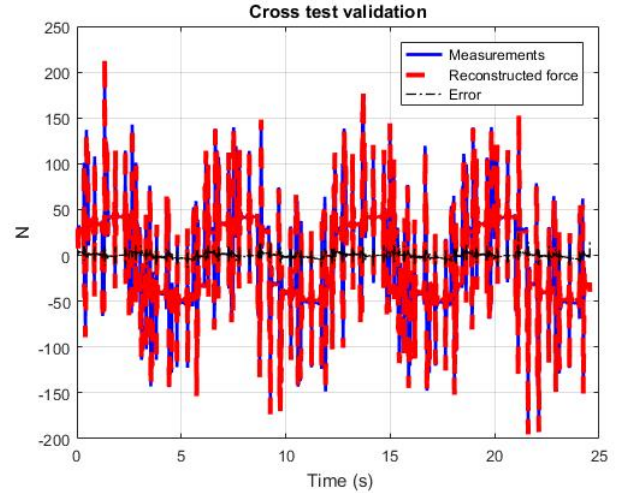


Fig. 4. Cross-test validation: comparison between the force reconstructed with a new set of data and the IDIM-LS estimates and the measured one

```
Parameter M1 :      72.1282
Parameter FV1 :     13.9021
Parameter FC1 :     69.1914
Parameter OF1 :    157.4499
Condition number of the observation matrix: 26.0217
```

```
Error norm : 169.4876
Relative error (%): 5.9824
Deviation of error: 3.4039
```

Then a comparison is plotted, see Fig. 4.

Finally, the whole system is simulated with the estimated values of the dynamic parameters (i.e. with $\hat{\boldsymbol{\theta}}_{LS}$) and the simulated force, position, velocity and acceleration are compared with the measured ones, see Fig.5. Again, all the relative errors between the simulated data and the measured ones are calculated with (11), e.g. when comparing the measured and the simulated positions, one has $\hat{\mathbf{y}} = \mathbf{q}_s$ and $\mathbf{y} = \mathbf{q}_m$ where \mathbf{q}_s is the simulated position.

3.5 Two source codes to simulate the EMPS

Two files allowing the simulation of the EMPS are provided. The simulation makes use of the DDM given by (3) and this is clearly emphasized by Fig. 2.

The first one is the **Simulation_EMPS** m file which calls the **Simulink_EMPS_Rigide** Simulink file generated with the MATLAB 2016b version. To properly simulate the EMPS, the following commands must not be modified

```
% Maximum voltage (saturation)
viradm = 10;
% For filtering the position
filterq1 = [0.5000 0.5000];
% Sampling
tec = t(10)-t(9);
% Duration
tf = t(end);
```

Once the code executed, something like that must be displayed on the MATLAB command window:

```
Simulation of EMSP with DATA_EMPS.mat

Simulation duration: 0h,0m,2.65441s.

Relative errors with EMPS_DATA.mat :
Position      : 0.013752%
Velocity      : 0.22911%
Acceleration  : 8.6036%
Force         : 5.848%
```

Finally, a direct comparison between the simulated and measured data is plotted, see Fig.5. According to the relative errors calculated with 11, the IDIM-LS estimates can be considered as good enough since these relative errors do not exceed 10%.

The second one is the **Simulation_EMPS_Validation** m file which calls the **Simulink_EMPS_Rigide_Pulse** Simulink file generated with the MATLAB 2016b version. This m file is used to perform cross-test validations by comparing the simulated data with the measured one with a trajectory that was not used to run the identification process. This m file loads the **EMPS_DATA_PULSE** mat file described in section 2.2.

Once the code executed, something like that must be displayed on the MATLAB command window:

```
Simulation of EMSP with DATA_EMPS_PULSES.mat

Simulation duration: 0h,0m,3.35169s.

Relative errors with EMPS_DATA_PULSES.mat :
Position      : 0.0080248%
Velocity      : 0.47888%
Acceleration  : 19.8152%
Force         : 8.9409%
```

Finally, a direct comparison between the simulated and measured data is plotted, see Fig.6. In this case, only the relative error between the measured and simulated acceleration calculated with 11 has significantly increased whereas the others do not exceed 10%. This rise may be explained by a flexibility that was accidentally excited by the pulses whereas the model does not account for it.

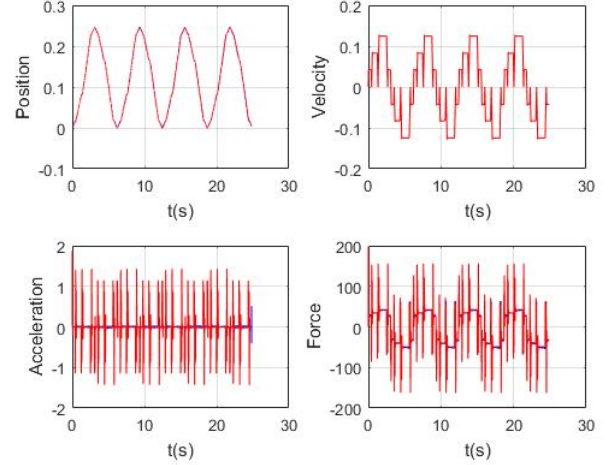


Fig. 5. Direct comparison between the simulated and measured data

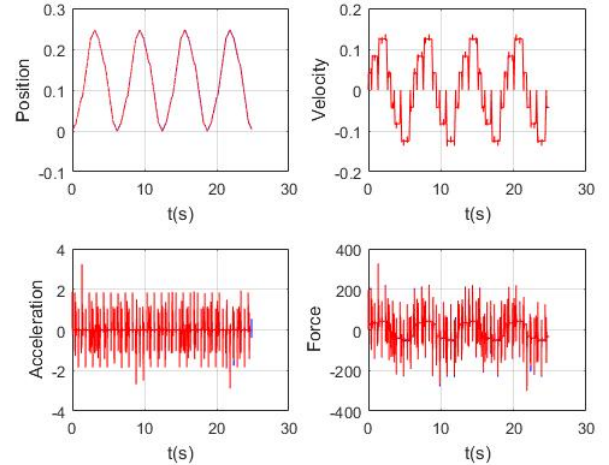


Fig. 6. Cross-test validation between the simulated and measured data

3.6 Comparison with toolboxes

We tried to identify a dynamic model with the CAPTAIN and the CONTSID toolboxes by assuming that it can be expressed with a simple transfer function, poor results were obtained:

- if $\nu(t)$ (i.e. **vir** Matlab variable) is considered as the input and $q_m(t)$ as the output (i.e. **qm** Matlab variable), a rel_{err} of 74.6 % (!) with a R_T^2 of 0.5 are obtained;
- if $\nu(t)$ (i.e. **vir** Matlab variable) is considered as the input and $\dot{q}_m(t) = \frac{d}{dt}(q_m(t))$ (calculated with the *diff* Matlab function) as the output, a rel_{err} of 47.3 % (!) with a R_T^2 of 0.8 is obtained.

For instance, a direct comparison between the measured velocity and the one estimated with the *rivebjid* function of the CAPTAIN toolbox by considering a linear model is illustrated in Fig.7. It is clear that the matching is not really good.

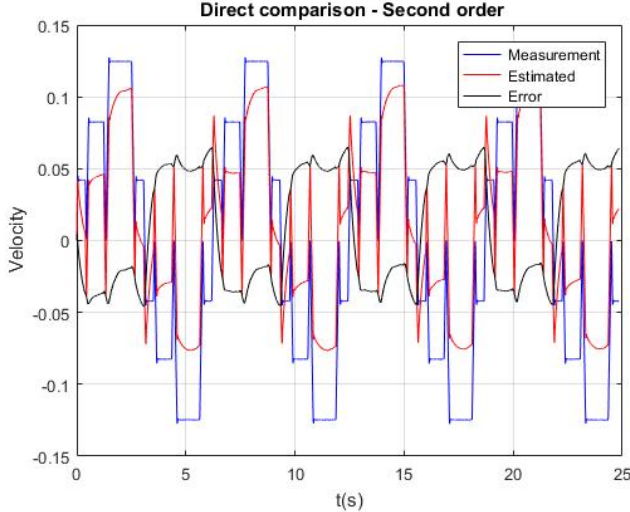


Fig. 7. Direct comparison between the measured velocity and estimated one obtained with the CAPTAIN toolbox

One notable exception is the identification of the whole closed loop i.e. by taking $q_r(t)$ as the input (i.e. **qg** Matlab variable), $q_m(t)$ as the output (i.e. **qm** Matlab variable) and a second-order structure. In this case, a rel_{err} of 0.005 % (!) with a R_T^2 of 1.0 are obtained. This result is not surprising. Indeed, it simply indicates that $d(t)$ has been well rejected by the controller which implies that the tracking is excellent. According to (Brunot et al., 2018a), this leads to a second-order system for the whole closed loop.

4. IMPROVING THE IDIM-LS METHOD

4.1 Considering other approaches

It is known that cautions must be taken when the LS estimation is used to identify a system operating in closed loop, see e.g. (Söderström and Stoica, 1983), (Ljung, 1999), (Gilson et al., 2011), (Young, 2011) among others. This explains why a tailor-made data filtering is employed to identify electro-mechanical systems.

Recently, a method based on Instrumental Variable approach was proposed in (Janot et al., 2014a), (Janot et al., 2014b) and then further improved in (Brunot et al., 2018b) and (Brunot et al., 2018a). This approach combines the IDM and DDM and has a good robustness against noises and/or data filtering and converges quickly. In (Gautier et al., 2013), the authors have proposed and validated an original *Input-Error* identification method. Indeed, because the system is identified in closed loop, it has been shown that the simulated position which is the usual output in classical *Output-Error* methods is not really sensitive to parameters' variations. Hence, the authors proposed to circumvent this problem by considering the simulated joint torques instead of the simulated joint positions. Like IV approaches, this input-error method has a good robustness against noises and data filtering and converges quickly. It is worth to note that it is possible to implement the IV and input-error methods mentioned above by using the Simulink file provided by the authors.

In (Olsen et al., 2002), the authors present an interesting Maximum Likelihood approach that is suitable for identification of industrial robots while in (Ramdani and Poignet, 2005) the authors have tested a set membership uncertainty method. A multi-objective genetic algorithm was used to identify the dynamic parameters of a hydraulic robot manipulator in (Montazeri et al., 2017). The authors suggested to employ the Linear Matrix Inequalities to identify the dynamics of a SCARA robot in (Calafiore and Indri, 2000).

Finally, in (Janot et al., 2017), the authors propose an original method by combining the IDM and a black-box identification method termed the *State-Dependent Parameter* identification method developed by P.C. Young, (Young, 2011) chapter 11. In so doing, he authors have shown that the friction model is asymmetric i.e. described by

$$\tau_{fric} = F_v^+ 0^+(\dot{q}) + F_c^+ \text{sign}(0^+(\dot{q})) + F_v^- 0^-(\dot{q}) + F_c^- \text{sign}(0^-(\dot{q})), \quad (12)$$

where $0^+(\dot{q}) = \dot{q} \cdot (1 + \text{sign}(\dot{q}))/2$ (resp. $0^-(\dot{q}) = \dot{q} \cdot (1 - \text{sign}(\dot{q}))/2$) is a mathematical operator which returns \dot{q} when $\dot{q} > 0$ (resp. $\dot{q} < 0$) and 0 otherwise; and F_v^+ and F_c^+ (resp. F_v^- and F_c^-) are the viscous coefficient and Coulomb coefficients for positive (resp. negative) velocities respectively. It is interesting to note that this asymmetry is due to a mechanical fatigue of the screw which is a phenomenon that can be hardly *a priori* accounted for.

4.2 An asymmetric friction model inserted into the IDM

The **Script IDIM_LS_Asym_Fric** m file presents a source code allowing the user running the IDIM-LS method with the asymmetric friction model (12) inserted into the IDM.

Once the code executed, something like that must be displayed on the MATLAB command window:

```
Parameter M1 : 95.1540 0.0827 0.0869 %
Parameter FV1p : 166.7061 1.2326 0.7394 %
Parameter FC1p : 20.1440 0.1088 0.5399 %
Parameter FV1m : 240.4236 1.2346 0.5135 %
Parameter FC1m : 20.6277 0.1091 0.5290 %
```

```
Absolute value of the diagonal element of R
Parameter M1 : 64.1376
Parameter FV1p : 9.8315
Parameter FC1p : 48.9895
Parameter FV1m : 9.8268
Parameter FC1m : 48.8267
Condition number of the observation matrix: 26.0197
```

```
Error norm : 82.8778
Relative error (%): 3.1117
Deviation of error: 1.6646
```

By considering the relative error, we can notice that the IDM was slightly improved.

REFERENCES

- Brunot, M., Janot, A., Carrillo, F., Garnier, H., Vandanjon, P.O., and Gautier, M. (2015). Physical parameter identification of a one-degree-of-freedom electromechanical system operating in closed loop. In *17th IFAC Symposium on System Identification*, 823–828.
- Brunot, M., Janot, A., Young, P.C., and Carrillo, F. (2018a). An improved instrumental variable method for

- industrial robot model identification. *Control Engineering Practice*, 74, 107–117.
- Brunot, M., Janot, A., Young, P.C., and Carrillo, F. (2018b). An instrumental variable method for robot identification based on time variable parameter estimation. *Kybernetika*, 54(1), 202–220.
- Calafiore, G. and Indri, M. (2000). Robust calibration and control of robotic manipulators. In *American Control Conf.*, 2003–2007.
- Gautier, M. (1991). Numerical calculation of the base inertial parameters of robots. *Journal of robotic systems*, 8(4), 485–506.
- Gautier, M. (1997). Dynamic identification of robots with power model. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, 1922–1927. IEEE.
- Gautier, M. and Briot, S. (2014). Global identification of joint drive gains and dynamic parameters of robots. *Journal of Dynamic Systems, Measurement, and Control*, 136(5), 051025.
- Gautier, M., Janot, A., and Vandanjon, P.O. (2013). A new closed-loop output error method for parameter identification of robot dynamics. *IEEE Transactions on Control Systems Technology*, 21(2), 428–444.
- Gilson, M., Garnier, H., Young, P.C., and Van den Hof, P.M. (2011). Optimal instrumental variable method for closed-loop identification. *IET control theory & applications*, 5(10), 1147–1154.
- Janot, A., Vandanjon, P.O., and Gautier, M. (2014a). A generic instrumental variable approach for industrial robot identification. *IEEE Transactions on Control Systems Technology*, 22(1), 132–145.
- Janot, A., Vandanjon, P.O., and Gautier, M. (2014b). An instrumental variable approach for rigid industrial robots identification. *Control Engineering Practice*, 25, 85–101.
- Janot, A., Young, P.C., and Gautier, M. (2017). Identification and control of electro-mechanical systems using state-dependent parameter estimation. *International Journal of Control*, 90(4), 643–660.
- Khalil, W. and Dombre, E. (2004). *Modeling, identification and control of robots*. Butterworth-Heinemann.
- Ljung, L. (1999). *System identification: theory for the user*. PTR Prentice Hall, Upper Saddle River, NJ.
- Montazeri, A., West, C., Monk, S.D., and Taylor, C.J. (2017). Dynamic modelling and parameter estimation of a hydraulic robot manipulator using a multi-objective genetic algorithm. *International Journal of Control*, 90(4), 661–683.
- Olsen, M.M., Swevers, J., and Verdonck, W. (2002). Maximum Likelihood Identification of a Dynamic Robot Model: Implementation Issues. *The International Journal of Robotics Research*, 21(2), 89–96.
- Ramdani, N. and Poignet, P. (2005). Robust dynamic experimental identification of robots with set membership uncertainty. *IEEE/ASME Trans. on Mechatronics*, 10(2), 253–256.
- Söderström, T.D. and Stoica, P.G. (1983). *Instrumental variable methods for system identification*, volume 57. Springer.
- Swevers, J., Verdonck, W., and De Schutter, J. (2007). Dynamic model identification for industrial robots. *IEEE Control Systems*, 27(5), 58–71.
- Young, P.C. (2011). *Recursive estimation and time-series analysis: an introduction for the student and practitioner*. Springer Berlin Heidelberg, 2nd edition.