An Autotuning Procedure for Motion Control Systems: Method and at-the-edge Implementation

Roberto Pagani*, Manuel Beschi*, Davide Colombo**, Giulia Facchini*, Antonio Visioli*
*Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Italy
**WEG Automation Europe S.r.l.
Email: roberto.pagani@unibs.it, manuel.beschi@unibs.it, dcolombo@weg.net,
g.facchini009@studenti.unibs.it, antonio.visioli@unibs.it

***Abstract*—Autotuning and system parameters monitoring are crucial aspects of modern motion control algorithms. At-the-edge controllers need to detect system changes and perform autotuning autonomously without requiring excessive computational burdens. Two algorithms to estimate the main dynamics of mechanical systems using integral figures of merit are proposed. The algorithms implementations are analyzed in offline and online modes. They have been tested in simulation, using an elastic model as a testbed, and in a hardware-in-the-loop PLC-controlled system. The results show the effectiveness of the methods compared to a recursive least square method.**

I. INTRODUCTION

There has been a growing demand for industrial devices to enhance their intelligence and connectivity in recent years [1]–[4]. This requirement stems from improving overall performance and efficiency in various industrial applications. In the field of motion control, controllers must be able to monitor and adapt to the performance and respond to changes in the system dynamics [5].

Extensive research has been conducted to develop computationally simple yet highly effective algorithms [6]–[8]. These algorithms serve the purpose of estimating the primary dynamics of the system. The obtained data plays a crucial role in diagnostics and controller autotuning. This ensures that the controller remains finely tuned over time and serves as a source of information related to plant variations. Developing computationally simple algorithms for estimating system dynamics brings several advantages to the industrial sector. Firstly, the simplicity of these algorithms allows for efficient implementation, reducing computational overhead and minimizing the required hardware resources. This makes them suitable for various industrial devices controlled by PLC-based controllers. Moreover, the effectiveness of these algorithms ensures an accurate estimation of the system dynamics, leading to reliable diagnostics and autotuning processes. The ability to identify and address issues promptly not only enhances operational efficiency but also improves the overall lifespan and reliability of the motion control system.

Mechanical systems employing brushless or asynchronous motors commonly rely on torque or current as the control variable, encoders measure angular position and velocity, and a Proportional-Integral-Derivative (PID) controller is used. However, these measured signals are inevitably influenced by noise, which can significantly affect the performance of estimation algorithms. To mitigate the impact of noise the use of integral figures of merit has been proposed in [9]. These figures of merit reduce the noise effect and enhance the estimation accuracy. This paper presents two novel algorithms that employ this approach in a mechanical system. The algorithms aim to estimate the primary dynamics of the system, represented by inertia and friction coefficients. By integrating the dynamic law during a ramp transient, a more robust and accurate estimation of system dynamics is provided.

The paper is organized as follows: Section II presents the system dynamics for rigid and elastic mechanical systems and the proposed estimation algorithms. Section III discusses the implementation issues in offline and online modes. Section IV shows the simulation and experimental results. Conclusions are in Section V

II. PROBLEM FORMULATION

The control architecture for speed control used in this work is defined by a feedback control scheme, in which two main elements are distinguished: the mechatronic system, which is the process to be controlled, and the control system, more in detail characterized by a PI regulator and a feedforward action.

A. Mathematical Models

A simple model of the system considers the motor rigidly connected to the load through a transmission with unitary efficiency and transmission ratio n. The purpose is to find a relationship between the input torque τ_m and the output speed of the motor shaft ω_m. Two effects are considered for modeling: inertia torque and friction. Inertia torque is linked to motor acceleration through the following expression:

$$\tau_{in} = J_{tot} \cdot \frac{d\omega}{dt} \tag{1}$$

where $J_{tot} = J_m + n^2 J_l$ [kgm^2] is the total inertia of the system, and J_l and J_m are the load and motor inertia, respectively. The classical friction models consist of different components taking care of many aspects of friction forces [10]. In this work, friction torque τ_{fric} is characterized by the two main components: τ_{stat} (Coloumb or static term) and τ_{dyn} (viscous or dynamic term).
The following expression describes static friction:

$$\tau_{stat} = T_{stat} \cdot \text{sgn}(\omega_m) \tag{2}$$

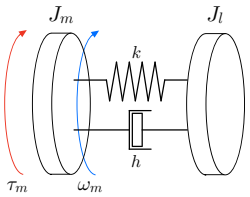


Fig. 1: Two-mass elastic servomechanism.

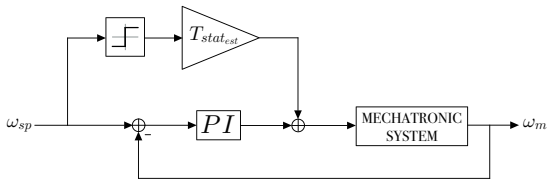


Fig. 2: PI controlled system with friction compensation.

where T_{stat} [Nm] is the static friction coefficient. This friction model does not specify a unique friction force for zero velocity, as it can take on any value in the interval between $-T_{stat}$ and T_{stat}. Viscous friction is a force component caused by the viscosity of lubricants. Its behavior is typically approximated as linearly dependent on the speed:

$$\tau_{dyn} = T_{dyn} \cdot \omega_m, \tag{3}$$

where $T_{dyn} = T_{dyn_m} + n^2 T_{dyn_l}$ [Nms/rad] is the total damping of the system (T_{dyn_m}: motor damping coefficient, T_{dyn_l}: load damping coefficient). A nonlinear dependence on velocity can obtain a better fit for experimental data, but a linear model is generally sufficient for controller tuning.
The overall dynamic model which describes a rigid system is obtained as:

$$\tau_m = J_{tot} \cdot \frac{d\omega_m}{dt} + T_{dyn} \cdot \omega_m + T_{stat} \cdot \text{sgn}(\omega_m). \tag{4}$$

Dynamic model (4) can be integrated over time as:

$$\begin{aligned} &\int_0^t \tau_m dt = J_{tot} \cdot (\omega_m(t) - \omega_0(0)) + \\ &T_{dyn} \cdot (y(t) - y(0)) + T_{stat} \cdot \int_0^t \text{sgn}(\omega_m) dt \end{aligned} \tag{5}$$

where $y(t) = y(0) + \int_0^t \omega_m dt$ is the angular position. It is worth noticing that (5) is less noise-sensitive due to the integration operation. A linear model can be obtained by neglecting the nonlinear part of friction:

$$\tau_m = J_{tot} \cdot \frac{d\omega_m}{dt} + T_{dyn} \cdot \omega_m \tag{6}$$

which results in the following transfer function between motor torque and motor speed:

$$\frac{\Omega_m(s)}{T_m(s)} = \frac{1}{J_{tot}s + T_{dyn}} = \frac{\frac{1}{T_{dyn}}}{\frac{J_{tot}}{T_{dyn}}s + 1} = \frac{K}{T_1 s + 1} \tag{7}$$

where $K = \frac{1}{T_{dyn}}$ and $T_1 = \frac{J_{tot}}{T_{dyn}}$.

From (4), (5), and the definition of K and T_1 it is possible to write:

$$\begin{cases} K \int_0^t (\tau_m - T_{stat}\text{sgn}(\omega_m))\, dt \\ \qquad = T_1 (\omega_m(t) - \omega_0(0)) + (y(t) - y(0)) \\ K (\tau_m(t) - T_{stat}\text{sgn}(\omega_m(t)) = T_1 \frac{d\omega_m}{dt} + \omega_m(t) \end{cases} \tag{8}$$

A more complex model is used as a simulated testbed, considering also the transmission elasticity (see Figure 1). In this case, the system can be modeled as a couple of masses (the motor and the load) linked together by a visco-elastic element described by the transmission stiffness k and damping coefficients h. The following equations describe the behavior of the system:

$$\begin{cases} J_m \cdot \dot{\omega}_m + T_{dyn} \cdot \omega_m + \tau_t + T_{stat} \cdot \text{sgn}(\omega_m) = \tau_m \\ J_l \cdot \dot{\omega}_l = \tau_t \\ \tau_t = k \cdot (\alpha_m - \alpha_l) + h \cdot (\omega_m - \omega_l) \end{cases} \tag{9}$$

where α_l and α_m are the link and motor angular positions, respectively; T_{dyn} is the viscous friction coefficient; τ_t is the torque delivered by the flexible element. Notice that α_l and J_l are referred to the motor shaft, $J_l = J_{ll}/n^2$, $\alpha_l = n \cdot \alpha_{ll}$, where n is the transmission ratio, and J_{ll}, α_{ll} are the corresponding quantities defined at the load side.

B. Controller Architecture

The structure of the controller can be represented by the block diagram shown in Figure 2. It includes a Proportional-Integral (PI) controller, which is described by the following transfer function:

$$C(s) = K_p \frac{1 + T_i s}{T_i s} \tag{10}$$

where K_p is the proportional gain and T_i is the integral time constant. An additional feedforward action is added to compensate for the estimated static friction value, ideally canceling this nonlinear contribution. It is worth noting that only an accurate estimation of static friction allows for minimizing the system's nonlinearity. In an autotuning procedure, the estimation of the process model is fundamental for the recalibration of the controller in the face of significant changes in the process dynamics. The choice of the specific automatic tuning method depends on various factors, including the dynamics and characteristics of the system, the available data and information, the desired performance criteria, and the available computational resources. This work considers the mechatronic system presented in Section II-A, ignoring the transmission's elasticity and damping for simplicity. The aim of the tuning is to obtain a phase margin $\phi_m = 75°$ for robustness, and a gain crossover frequency, which is an user-defined, chosen equal to $\omega_c = 80$ [rad/sec]. Therefore, the controller parameters result in:

$$T_i = \frac{\tan\left(-90° + \phi_m + \arctan\left(\frac{J_{tot} \cdot \omega_c}{T_{dyn}}\right)\right)}{\omega_c} \tag{11}$$

and

$$K_p = \frac{T_i \cdot \omega_c \cdot \sqrt{(J_{tot} \cdot \omega_c)^2 + T_{dyn}^2}}{\sqrt{1 + (T_i \cdot \omega_c)^2}} \tag{12}$$

The goal is to estimate J_{tot} and T_{dyn}, to obtain the transfer function (7), used for controller tuning, while the value of T_{stat} is necessary for the estimate of the feedforward action. Two online identification methods are proposed and compared with a recursive least square (RLS) algorithm. The proposed algorithms consider the integral of (4) during a ramp response. The usage of the integral is motivated by the reduction of the measurement noise [9].

C. Identification Method I

The first strategy allows the user to estimate the static friction initially, then the gain K and the time constant T_1 of the first order model (7) and, consequently, the total inertia and dynamic friction values. Parameter estimation takes place following the application of a speed set-point y_{sp}, which is characterized by initial and final sections at a constant speed, respectively ω_0 [rad/s] and ω_f [rad/s] and a central section that grows linearly with constant acceleration a [rad/s^2]. The amplitude of the ramp is defined as $A = \omega_f - \omega_0$. The presence of static friction introduces a nonlinearity and affects the transfer functions and dynamic behavior. To accurately estimate the real system, a feedforward action with an estimate of static friction has been implemented, compensating for friction torque and making the system behave like a first-order linear system. The methodology can be summarized as follows:

1) Guess an initial value for the static friction estimate variable (T_{stat}^{est}) or assume it zero.
2) Considering positive velocities for simplicity, it is possible to estimate the gain K from (8) computed at the beginning and the end of the ramp.

$$K^{est} = \frac{A}{u_{ss} - T_{stat}^{est}} \tag{13}$$

and time constant (T_1) using (8):

$$T_1^{est} = \frac{\int K^{est} \cdot (u(t) - T_{stat}^{est}) - y(t)}{A} \tag{14}$$

where u_{ss} is the steady-state value of the torque.
3) Build a first-order system (P_{fo}) using the estimated values of K and T_1.

$$P_{fo}(s) = \frac{K^{est}}{T_1^{est} s + 1} \tag{15}$$

4) Perform an offline simulation on the estimated system P_{fo} using the real torque input $u(t)$ minus the current estimate of static friction T_{stat}^{est} as input.
5) Calculate the integral of the difference between the measured speed $y(t)$ and the simulated output as a performance index for the simulation.
6) Increment or decrement the value of T_{stat}^{est} by a fixed amount δ (chosen to be a small quantity w.r.t. the maximum torque) and repeat steps 2, 3, 4, and 5.
7) Compare the performance index associated with the current operating conditions (from step 6) and the one associated with the aforementioned operating conditions (from step 5).
8) If the current performance index is lower than the previous one, continue to increase (or decrease, depending on the initial choice of T_{stat}^{est}) the static friction estimate until the current index is greater than the previous one. If the current index exceeds the previous one, implement the previous friction value as the feedforward action.

D. Identification Method II

The second identification strategy involves estimating the system's gain (K) and time constant (T_1), as well as the total inertia, viscous friction coefficient, and static friction. This strategy applies when the motor is already in motion ($\omega_0 \neq 0$), and it compares the measured case and the case where data is translated to the origin of the Cartesian plane. The key concept behind this strategy is that the system has a non-zero initial torque when it is already in motion. The estimated static friction (T_{stat}^{est}) can be interpreted as a step disturbance that is added to the control variable. Without feedforward action, a PI controller deals with a first-order process, with the step disturbance compensated by the controller integrator. The output of the PI controller can be seen as the output that would occur without feedforward action but with a contribution added to compensate for the presence of static friction. To achieve this, the speed and torque trends are detrended to the Cartesian plane's origin, removing the effect of static friction, which is constant since there are no speed changes. Therefore the terms K, T_1, J_{tot}, and T_{dyn} can be determined neglecting static friction. Static friction can be obtained by considering the steady-state conditions before and after applying the ramp. The algorithm involves the following steps:

1) The recorded torque and speed values $u(t)$ and $y(t)$ are translated to the origin, obtaining $u_0(t) = u(t) - u(0)$ and $y_0(t) = y(t) - y(0)$.
2) The values of the gain K and time constant T_1 are estimated using (8) computed at the beginning and the end of the ramp:

$$K^{est} = \frac{A}{u_{0,ss}};\ T_1^{est} \quad = \frac{\int_0^{t_{ss}} K^{est} \cdot u_0(t) - y_0(t)}{A} \tag{16}$$

where $u_{0,ss}$, $A = y_{0,ss}$, and t_{ss} are the steady-state values of $u_0(t)$, $y_0(t)$ and t, respectively.
3) The static friction value is estimated using the non-detrended data:

$$T_{stat}^{est} = u_{ss} - \frac{y_{ss}}{K^{est}} \tag{17}$$

where u_{ss} and y_{ss} are the steady-state values of non-translated torque and velocity, respectively.

E. Baseline: RLS identification

The proposed methods are compared to the RLS method, a common technique for online model parameter estimation [11]. The model is represented as a linear combination of known (possibly nonlinear) functions, called regressor (φ_1), multiplied by a vector of unknown parameters θ, thus the observed variable $x = \varphi\theta$. The least-square (LS) methods determine the values of θ that minimize the least squares loss function. The LS problem can be solved analytically in batch (offline) form, where the observations are collected in matrices and vectors. This algorithm requires storing data and performing a matrix inversion, which is too complex for the limited computation capabilities of many motor controllers. RLS algorithms process data in a recursive fashion with limited storing and computation needs.

In this paper, we consider the Householder RLS algorithm [12]

$$\hat{\theta}(t) = \hat{\theta}(t-1) + S(t)^{-1}\varphi(t)\big[y(t) - \phi(t)^T\hat{\theta}(t-1)\big] \quad (18)$$

where

$$S(t) = S(t-1) + \varphi(t)\varphi(t)^T$$

Defining the terms $\gamma(t)$ and $\varepsilon(t)$ as follows:

$$\gamma(t) = S(t)^{-1}\varphi(t)$$
$$\varepsilon(t) = y(t) - \phi(t)^T\hat{\theta}(t-1)$$

the recursive expression (18) becomes:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\varepsilon(t) \quad (19)$$

To avoid the matrix inversion, it is possible to use the matrix inversion lemma (or Householder's lemma). This implies that once $S(t-1)^{-1}$ is given, calculating $S(t)^{-1}$ is only a matter of inverting a scalar and performing a few matrix multiplications. Finally, defining $Q(t) = S(t)^{-1}$, the form of recursive least squares can be summarized as follows:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\varepsilon(t) \qquad basic\ recursion$$
$$\gamma(t) = Q(t-1)\varphi(t)\big(1 + \varphi(t)^T Q(t-1)\varphi(t)\big)^{-1} \qquad definition$$
$$\varepsilon(t) = y(t) - \varphi(t)^T\hat{\theta}(t-1) \qquad definition$$
$$Q(t) = \big(I - \gamma(t)\varphi(t)^T\big)Q(t-1) \qquad auxiliary\ recursion$$

$\varepsilon(t)$ is the difference between the measured output y(t) and the one-step-ahead prediction $\hat{x}\big(t|t-1;\hat{\theta}(t-1)\big)$ made at time $t-1$.

Values of $x(t)$, $\varphi(t)^T$ and θ can be constructed from (5):

$$y(t) = \omega_m(t) - \omega_m(0) \quad (20)$$
$$\varphi(t)^T = \Big[\int \tau_m(t) \quad \int \omega_m(t) \quad \int \mathrm{sgn}(\omega_m(t))\Big] \quad (21)$$
$$\theta = \begin{bmatrix} \dfrac{1}{J_{tot}} & \dfrac{T_{dyn}}{J_{tot}} & \dfrac{T_{stat}}{J_{tot}} \end{bmatrix} \quad (22)$$

Quantity	Value	Unit of measurement
Total inertia, J_{tot}	0.008	[kgm^2]
Viscous friction, T_{dyn}	0.0025	[Nms/rad]
Static friction, T_{stat}	0.15	[Nm]

TABLE I: Values of inertia, static friction coefficient and viscous friction coefficient.

Quantity	Value 1	Value 2	Unit of measurement
Stiffness, k	62.8812	2.5152	[Nm/rad]
Damping, h	0.08	0.02	[Nms/rad]
Inertia load, J_l	0.00267	0.00267	[kgm^2]

TABLE II: Additional parameters for the elastic model.

III. Implementations

The proposed identification algorithms have been initially tested in simulation to verify their functioning. MATLAB 2021b and Simulink software have been used. Subsequently, each algorithm has been revised and implemented in a new form to be suitable for use in an industrial context (following the IEC61131 standard). The simulation tests are performed by adopting two different systems to represent the physical mechatronic system: one with rigid transmission (4) and one with elastic transmission (9). The values of total inertia J_{tot}, viscous friction T_{dyn}, and static friction T_{stat} coefficients are the same for both the rigid and the elastic systems. In particular, a mechatronic system with the values reported in Table I is considered for the simulation test. The identification algorithms aim to obtain estimates that ideally are equal to these fixed values that are unknown in reality.

The simulations of the system with elastic transmission are characterized by two pairs of elasticity and damping values, with the aim to simulate an "almost rigid" system and a "very elastic" system. Their values are shown in Table II. It is also chosen that $J_l = 0.5 \cdot J_m$ ($J_{tot} = J_m + J_l$).

Figure 3 reports a flow chart that summarizes the implementation of identification method 1. For simulation tests, the value of δ is set equal to 0.01 [Nm] and the first estimation of static friction T_{stat}^{est} is set equal to 0 [Nm]. The set-point goes from speed $\omega_0 = 0$ [rad/s] to $\omega_f = 30$ [rad/s], so $A_{sp} = 30$ [rad/s]. Figure 4 reports the flow chart of the function which implements identification method 2. The set-point ranges $\omega_0 = 30$ [rad/s] to $\omega_f = 60$ [rad/s], so $A_{sp} = 30$ [rad/s]. The main steps of the function that implements the identification algorithm based on the recursive least squares method are shown in the flow chart in figure 5. The algorithm is tested both with a set-point starting from speed $\omega_0 = 0$ [rad/s] and reaching $\omega_f = 30$ [rad/s], and with a set-point which ranges $\omega_0 = 30$ [rad/s] to $\omega_f = 60$ [rad/s].
As can be seen from the flow chart, θ and Q must be initialized. For θ an arbitrary parameter estimate is used, while for Q, the identity matrix multiplied by a number in the range between 100 and 10000 is chosen. The forgetting factor λ is set equal to 0.995, which has been added to the least square criterion to discount the importance of long-past data so that

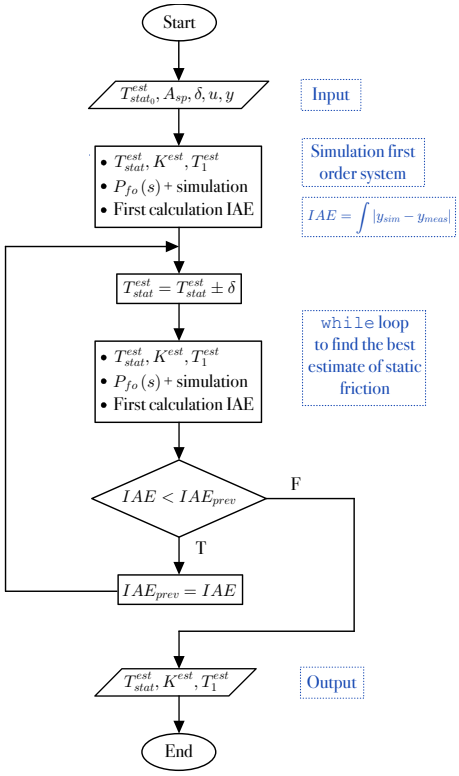


Fig. 3: Identification Algorithm 1 flow chart.

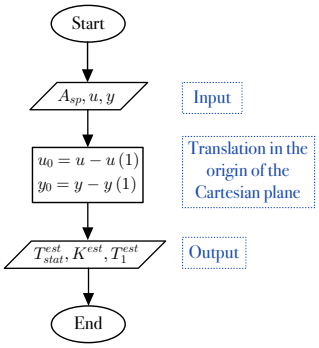


Fig. 4: Identification Algorithm 2 flow chart.

recent data mostly influence the estimation.

Three factors influence the estimation: static friction, measurement noise, and acceleration time. In the case of a variable acceleration time (*i.e.*, ramp slope), different simulations are performed for each system, changing the characteristics of the reference signal each time; otherwise, a constant acceleration is maintained.

After testing the identification algorithms in simulation, (see Section IV) they were adapted to accommodate the specific features of industrial controllers. When working with industrial drives or PLCs, implementation aspects are often disregarded when using simulation and calculation software such as MATLAB. These include limited computational and memory capabilities, the use of Single Precision format (32 bits) instead

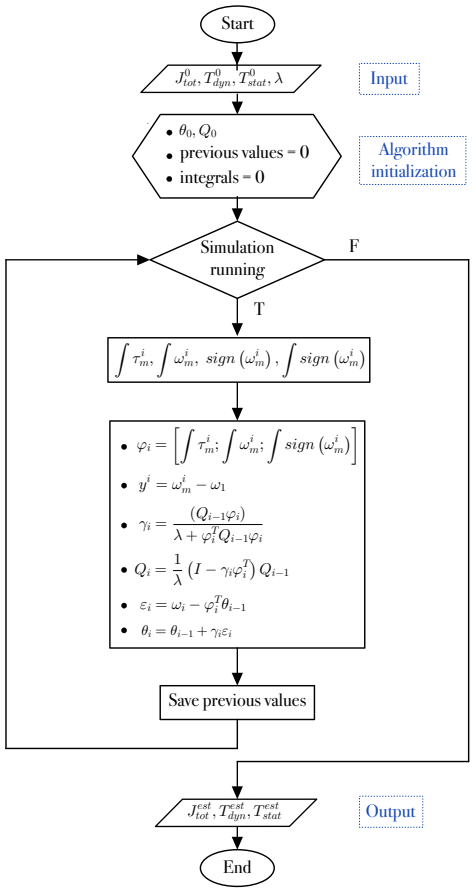


Fig. 5: Recursive least squares identification algorithm flow chart.

of Double Precision format (64 bits) to represent floating-point numbers, the presence of a real-time operating system that manages real-time and non-real-time tasks executed cyclically every few milliseconds, and high-frequency noise affecting all measured data. The proposed algorithms must be written in Structured Text language, one of the languages supported by the IEC 61131-3 standard. Before proceeding, the algorithms are modified and tested in simulation using Simulink software on a standard PC. This step has two main benefits: it facilitates the debugging process in a more user-friendly environment, allows obtaining information about the algorithms' operation, and selects the most promising ones to proceed with on the experimental system. Identification Algorithm 1 undergoes significant changes due to two main issues. Firstly, the offline simulation of the first-order system performed using the MATLAB command *lsim* must be replaced since there is no equivalent in Structured Text language. Secondly, only a few samples of the previous steps can be stored, which limits the processing of data in batch form. The ZOH method solves the first problem by discretizing the first-order system response and calculating it progressively at each step. However, the second problem is more challenging to handle while allowing the estimation of model parameters through the response to a

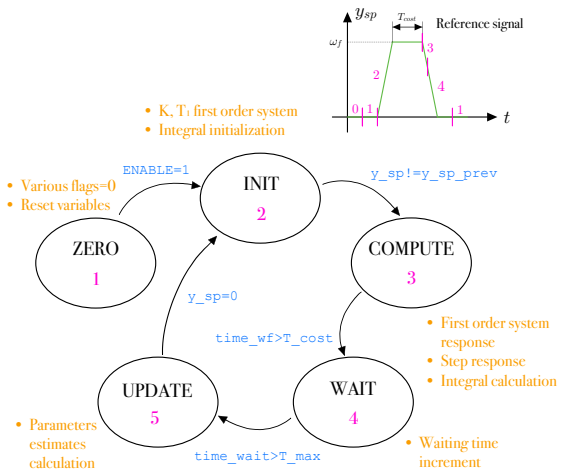


Fig. 6: State diagram of modified identification Algorithm 1.

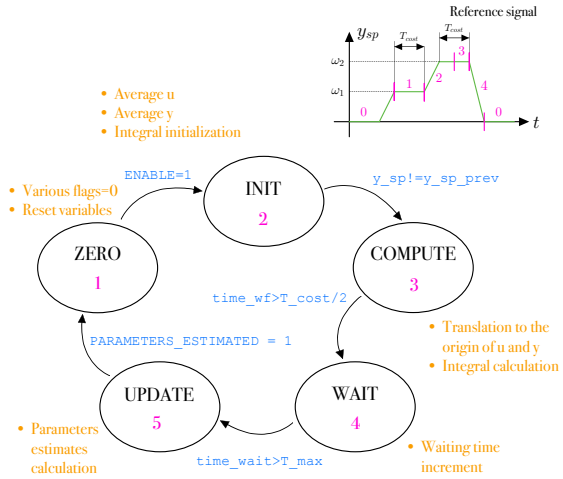


Fig. 7: State diagram of modified identification Algorithm 2.

single-speed set-point ramp. The modified algorithm includes the following steps: (1) initial estimates of total inertia and dynamic friction are given, (2) for the ascending section of the ramp, the discretized response and unit step response of the first order system are calculated at each step using the ZOH method, and integrals are computed through the trapezoidal rule, (3) static friction estimation is calculated, and new gain and time constant values are computed to obtain the estimates of total inertia and dynamic friction, and (4) steps 2 and 3 are repeated for a new ramp using the previously calculated gain and time constant values. This modification implies a recursive parameters estimation over multiple ramp responses. Figure 6 shows the diagram of the state machine implemented for modified identification Algorithm 1.

Identification Algorithm 2 requires less changes. The system receives a reference signal consisting of two ramps: the first, to make the motor rotate from zero speed to ω_1; the second, to bring the motor from speed ω_1 to speed ω_2. The algorithm is applied in the section starting from ω_1 and ending at ω_2. Figure 7 shows the diagram of the state machine implemented for modified identification Algorithm 2.

The Recursive Least Squares algorithm is suitable for PLC implementation as it processes data without batching. The algorithm only requires element-by-element calculations of matrices and vectors. However, the algorithm has three main problems: 1) symmetry and positive definiteness of $Q(t)$ can be lost for numerical reasons, which can be overcome by factorizing $Q(t)$ using Cholesky decomposition, but this may complicate the PLC code; 2) the Q matrix is poorly-conditioned, which is a general property of any least squares matrix; 3) if the elements of Q or γ are too large, overflow can occur due to the 32-bit representation of floating point numbers. Thus, a threshold for the number of points considered must be established. Figure 8 shows the state diagram of the modified recursive least squares identification algorithm.

IV. Results

A. Simulation Results

The simulation tests yielded important results, which are presented to highlight the key features, benefits, and drawbacks of each identification algorithm. The key features of identification Algorithm 1, as obtained from the simulations, are summarized as follows:

- Despite the presence of measurement noise, the algorithm performs well and improves the control system's performance, even when the parameters are not estimated accurately.
- The algorithm yields good parameter estimates for different accelerations, stimulating the system at various frequencies.
- The algorithm is effective across a wide range of speeds, with optimal performance at medium and high speeds.

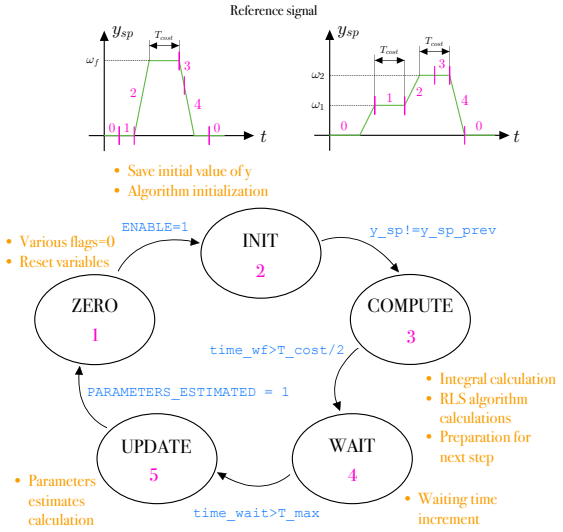


Fig. 8: State diagram of modified RLS identification algorithm.

- The algorithm requires reference signals starting from zero speed and demands careful attention when removing signals during steady-state conditions in the presence of noise.
- Due to its iterative simulation-based approach, the algorithm is computationally intensive and can only function offline.

Based on the simulation results, the main characteristics of Algorithm 2 can be summarized as follows:

- The algorithm performs well in the presence of measurement noise.
- It is advisable to use data for estimation only until shortly after the system reaches steady-state conditions.
- The algorithm performs well across the motor speed range (except very low speeds) and for any acceleration time.
- Compared to Algorithm 1, Algorithm 2 is less computationally intensive.

Regarding the recursive least squares algorithm, it exhibits the following characteristics:

- The algorithm is effective for reference signals starting at zero speed or with the system already in motion, making it more versatile than the other two algorithms.
- The algorithm produces reliable results across various speeds and accelerations.
- The algorithm already has a recursive form, simplifying its implementation on a PLC or industrial drive. However, a significant issue is the poor conditioning of the Q matrix, which tends to be very high in multiple iterations, resulting in unacceptable parameter estimates in the presence of noise. This aspect must be carefully monitored and could present problems implementing the algorithm on an industrial controller.

B. Implementation Results

Important factors are revealed by simulation experiments on identification algorithms modified for PLC implementation. Algorithm 1 cannot complete the estimation operation using a single set-point speed ramp. For this reason, it is necessary to use a recursive approach which could lead to instability, as shown in Figure 9. The recursive least squares algorithm is complex and sensible to inadequate conditioning. On the other hand, Algorithm 2 uses a single application of a double ramp set-point to get accurate estimations of model parameters. It works well with single-precision encoding of floating point numbers and noise. Table III shows the results of some tests with different final velocities w_f and accelerations a, where J_{tot}, T_{dyn}, and T_{stat} are the actual values of inertia, static friction coefficient, and viscous friction coefficient used for the simulations (see Table I), while $J_{tot}^{est}, T_{dyn}^{est}, T_{stat}^{est}$ are the estimated ones. Given the promising results of Algorithm 2 in simulations, it was selected for testing in a physical testbed.

C. Experimental Results

The experimental setup comprises several components: A brushless servomotor, an absolute encoder (Heidenhain EQN 1325) for position and speed measurement, an ADV200S drive by Gefran S.p.A. for motor control, a real-time controller, a standard host PC for software development, and a control panel to initiate servomotor start-up. EtherCAT communication transmits speed, torque data, and process-related signals with a 1 [ms] sampling time. The host PC utilizes MDPlc software to manage signal processing, model estimation, and parameter control. Two types of tests are performed: the first one to verify if the algorithm is consistent and robust, and the second one in which the identified parameters are used for the retuning of the control system. For the first tests, a reference signal consisting of five repetitions of a double ramp was used to evaluate how the estimates obtained from the first ramp differed from those obtained from subsequent ramps. The results indicate that the estimation of the parameters is consistent, and one ramp is enough to estimate them, particularly at medium and high speeds. The second set of tests assesses if the identified parameters improve the control system performance. The controller is initially detuned and tested without feedforward action. The identified parameters are then used to tune the PI controller according to a specific tuning rule, and the feedforward action is set equal to the static friction estimate. The tests were conducted under different conditions. However, for brevity, only those with a ramp from $\omega_1 = 10$ [rad/s] to $\omega_2 = 40$ [rad/s], and an acceleration time

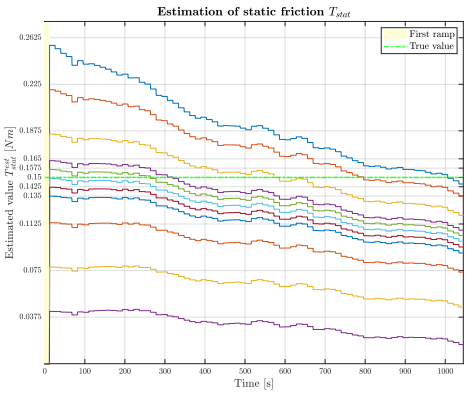


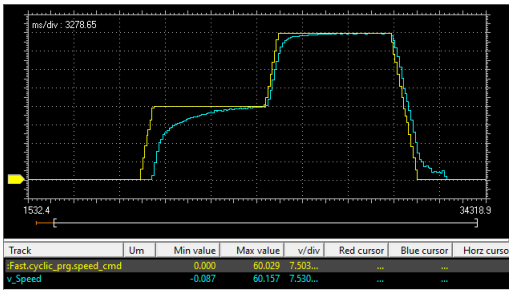
Fig. 9: Estimation of static friction for different initial values of T_{stat}^0, modified identification algorithm 1.

ω_f [rad/s]	a[rad/s^2]	J_{tot}^{est} [kgm^2]	T_{dyn}^{est}[Nms/rad]	T_{stat}^{est}[Nm]
20	1	0.0064	0.0039	0.1352
	10	0.0057	0.0032	0.1385
	100	0.0052	0.0023	0.1564
40	3	0.0054	0.0033	0.1473
	30	0.0071	0.0027	0.1461
	300	0.0070	0.0018	0.182
80	7	0.0052	0.0032	0.1617
	70	0.0076	0.0025	0.1517
	700	0.0075	0.0016	0.2236

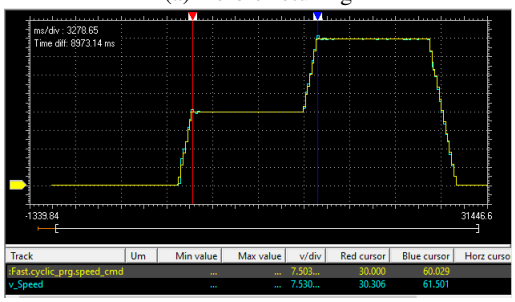
TABLE III: Rigid model parameters estimated by modified identification algorithm 2 with single precision.

Parameter	Value
J_{tot}^{est}[kgm^2]	0.00077175
T_{dyn}^{est}[Nms/rad]	0.00104824
T_{stat}^{est}[Nm]	0.10488
K_p^{old}[N/rpm]	0.0005
T_i^{old}[ms]	2000
FF^{old}[Nm]	0
K_p^{new}[N/rpm]	0.0012
T_i^{new}[ms]	34
FF^{new}[Nm]	0.10488

TABLE IV: Old and new identified rigid model parameters, old and new controller parameters, $a = 30$ [rad/s^2], experimental tests of identification algorithm 2.



(a) Before retuning



(b) After retuning

Fig. 10: Reference signal and measured motor speed before and after retuning, $a = 30$ [rad/s^2], experimental tests of identification algorithm 2. The reference signal is marked in yellow, the measured motor speed is marked in blue.

of $1s$ are presented. The numerical results are shown in table IV. Figure 10 shows the comparison between the response with the detuned controller (on the top) and with retuned controller (at the bottom).
It is worth noticing the performance improvements and the effectiveness of the proposed method to estimate the primary system dynamics starting from a poor-tuned situation.

V. CONCLUSION

In this study, two algorithms were developed and tested to estimate the dynamics of a mechatronic system by using the integral figures of merit. Their goal is to automatically tune the parameters of a PI controller and set the feedforward action for a speed control system. Most efforts were focused on estimating the parameters of an approximate model of a mechatronic system with rigid transmission. After simulation tests, the identification algorithms were modified considering the computational limit of an industrial controller and tested in Simulink using an elastic system as a testbed. The algorithm with the most robust and effective results, algorithm 2, was further tested on a Hardware-In-the-Loop system using PLC Structured Text language. Algorithm 1 was found to give accurate estimates of the model parameters in an offline form but can be computationally heavy. Its modified online form has convergence problems. It is worth noticing that the continuous increment of the computational capability of industrial controllers will attenuate this drawback in the future. Algorithm 2 shows the most robust and effective performance in identifying model parameters. A recursive least squares algorithm is used for comparison, and it shows potential issues with the conditioning. The results of this study have demonstrated the feasibility and effectiveness of using these algorithms for mechatronic system dynamics estimation and automatic tuning of controllers.

ACKNOWLEDGMENT

This work was supported by ECSEL Joint Undertaking in H2020 project IMOCO4.E, grant agreement No.101007311.

REFERENCES

[1] V. Lesi, Z. Jakovljevic, and M. Pajic, "Iot-enabled motion control: Architectural design challenges and solutions," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2284–2294, 2023.
[2] Y. Koyasako, T. Suzuki, S.-Y. Kim, J.-I. Kani, and J. Terada, "Real-time motion control method using measured delay information on access edge computing," in *2020 IEEE 17th Annual Consumer Communications and Networking Conference (CCNC)*, 2020, pp. 1–4.
[3] Y. Koyasako, T. Suzuki, T. Yamada, T. Shimada, and T. Yoshida, "Demonstration of real-time motion control method for access edge computing in pons," *IEEE Access*, vol. 10, pp. 168–175, 2022.
[4] M. Čech, A.-J. Beltman, and K. Ozols, "Digital twins and ai in smart motion control applications," in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–7.
[5] L. Yanan, L. Jiangang, Y. Xuehui, and L. Zexiang, "Self-tuning controller design for motion control systems," in *2009 International Conference on Information and Automation*, 2009, pp. 472–477.
[6] P. Niranjan, S. C. Shetty, C. D. Byndoor, K. V. S. S. S. S. Sairam, and S. Karinka, "Friction identification and control of ball screw driven system using plc," in *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, 2016, pp. 803–808.
[7] F. Liu, H. Peng, D. Zhang, X. Huang, L. Tong, R. Wu, and X. Peng, "Adaptive positioning control of servomotor based on model identification," *IEEE Transactions on Power Electronics*, vol. 37, no. 5, pp. 5272–5283, 2022.
[8] M. Giacomelli, D. Colombo, G. Finzi, V. Šetka, L. Simoni, and A. Visioli, "An autotuning procedure for motion control of oscillatory mechatronic systems," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 829–835.
[9] M. Veronesi and A. Visioli, "Performance assessment and retuning of pid controllers for integral processes," *Journal of Process Control*, vol. 20, no. 3, pp. 261–269, 2010.
[10] H. Olsson, K. J. Åström, C. C. De Wit, M. Gäfvert, and P. Lischinsky, "Friction models and friction compensation," *Eur. J. Control*, vol. 4, no. 3, pp. 176–195, 1998.
[11] K. J. Åström and B. Wittenmark, *Adaptive control*. Courier Corporation, 2013.
[12] A. A. Rontogiannis and S. Theodoridis, *Householder-Based RLS Algorithms*. Boston, MA: Springer US, 2009, pp. 1–23. [Online]. Available: https://doi.org/10.1007/978-0-387-09734-3_7