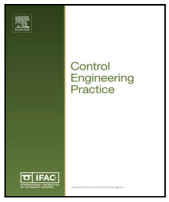
Control Engineering Practice 139 (2023) 105634

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac



A data-driven control methodology applied to throttle valves

Emmanuel Witrant [a,*], Ioan Doré Landau [a], Marie-Pierre Vaillant [b]

[a] Univ. Grenoble Alpes, CNRS, GIPSA-lab, 38000 Grenoble, France
[b] Univ. Grenoble Alpes, CNRS, G2Elab, 38000, Grenoble, France

ARTICLE INFO

Keywords:
Data-driven control
System identification
PI design
Adaptive control
Electronic throttle control (ETC)

ABSTRACT

Electric throttle valves represent a challenge for control design, as their dynamics involves strong nonlinearities, characterized by an asymmetric hysteresis. Large variability in the characteristics of each valve and erratic steady-state behaviors can also be noticed by carrying out experiments on multiple valves, impairing classical model-based control strategies. Nevertheless, local data-driven linear models can be obtained by system identification, and simple proportional–integral (PI) digital controllers can be tuned individually for each valve, providing good tracking performance. As these controllers cannot be transposed from one valve to another, a robust control design is considered. Taking into account the variability of electric throttle valves, a real-time data-driven strategy is then proposed, using identification in closed-loop and controller re-design. This methodology is necessary if control performance is a key issue, and can be embedded on a low-cost controller board (Arduino® Mega 2560). Experimental results going from frequency analysis and linear design to real-time data-driven control illustrate the methodology presented in the paper.

1. Introduction

Electric throttle valves are the most frequent devices used in industry for flow control. The valve considered in this paper is a butterfly valve, which regulates the downstream pressure by adjusting the rotation of a disk. Such valves, with a relatively low cost and a fast response time, are classically used in the automotive, chemical, pharmaceutical, and food industries. Nevertheless, the electromechanical apparatus of throttle valves induces complex dynamics that needs to be handled with care to satisfy precise flow regulation objectives.

Throttle valve control is a particularly challenging topic, which has motivated dedicated research since the 1960s when an output feedback design was proposed to obtain high-accuracy steam valve control (Johnson & Thompson, 1965). The process is modeled as a second-order system, and a frequency analysis leads to the design of a proportional feedback with reasonable gain. Considering the valve regulation in a large steam turbine-generator unit, the authors in Callan and Eggenberger (1965) conclude that a lead–lag compensator is more suitable and that PI control may result in limit cycles when combined with the valve dead zone for inadequate gain values. The complexity of the environment is considered one step further by Kwatny and Fink (1975), where the impact of acoustic waves in the piping of a boiling unit is considered. The concepts of state variable feedback and dynamic observer are used in this case to control a simplified model.

More recently, electric throttle valves triggered the interest of many researchers in automatic control, as the static friction effects and the nonlinearities of the gearbox and of the return spring induce particularly complex dynamics (usually modeled with an asymmetric hysteresis). A brief review of some works supported by experimental evaluation is given here. Linear approaches are proposed, for example, with linear quadratic control (Cassidy, Athans, & Wing-Hong, 1980), robust $\mathcal{H}_\infty$ design (Vargas et al., 2014), and linear parameter-varying (LPV) modeling and mixed constrained H_2/H_∞ control (Zhang, Yang, & Zhu, 2015). The valve friction has motivated specific control strategies, such as the one proposed by the authors of Canudas de Witt, Kolmanovsky, and Sun (2001), who derived a dynamic model including friction and aerodynamic torque, and proposed an adaptive pulse control strategy. A hybrid feedforward–feedback friction compensator with friction parameter adaptation is proposed in Panzani, Corno, and Savaresi (2013) and an adaptive PID feedback controller with adaptive feedforward compensators for friction, limp-home, and backlash is proposed in Jiao, Zhang, and Shen (2014). Some other methods focus on handling the asymmetry of the return-spring: it is taken into account with a nonlinear asymmetric PI controller in Pujol, Vidal, Acho, and Vargas (2016) and, combined with the friction effect, with a hybrid LPV method in Hamze (2019). Sliding mode control methods also brought some interesting contributions, with a robust adaptive chatter-free strategy using a genetic algorithm in Ye and Wang (2020) and an adaptive scheme based on the recursive terminal sliding mode in Hu et al. (2021).

Since electrical throttle valves show a quite large dispersion of their characteristics, a data-driven control design should be considered if

* Corresponding author.
E-mail addresses: emmanuel.witrant@univ-grenoble-alpes.fr (E. Witrant), ioan-dore.landau@gipsa-lab.fr (I.D. Landau), marie-pierre.vaillant@univ-grenoble-alpes.fr (M.-P. Vaillant).

https://doi.org/10.1016/j.conengprac.2023.105634
Received 20 February 2023; Received in revised form 13 July 2023; Accepted 16 July 2023
Available online xxxx
0967-0661/

one wants a high-performance control system. There are many data-driven approaches to control design. These methods can be roughly classified as direct and indirect data-driven approaches (see Karimi, Bahrani, Zheng, and Madani (2022) for a review). Indirect data-driven approaches rely on the fundamental idea that a "control-oriented model" can be identified (i.e., a model for which a design method providing the expected performance is available, also called a "design model", see Gevers (1993), Landau (1998), Van den Hof and Schrama (1995)). It is shown in this paper, based on extensive experiments with numerous electric throttle valves, that indeed a simple "control-oriented" linear model can be identified, leading to the design of a robust digital controller. This controller will be termed an "open-loop based controller"(OLBC) since it is based on plant model identification in an open loop. However, despite robustness issues addressed in the design of the OLBC controller, the large variability of the plant model from one valve to another requires online tuning of the controller to obtain the best performance. Previous works (Gevers, 1993; Landau, 1998) have shown that if an appropriate algorithm for system identification in a closed loop is available[1] then a better control model is identified, and the controller designed based on this model, called a "closed-loop based controller", provides better performance than the OLBC. The explanation is that such an algorithm weights the frequency bias distribution (identification error) by the square of the modulus of the output sensitivity function. This means that the quality of the identified model is enhanced in the critical frequency region for design, close to the Nyquist instability point.[2] An identification algorithm that satisfies this requirement is, for example, CLOE (closed loop output error identification algorithm) (Landau & Zito, 2006). This operation (identification in closed loop and re-design of the controller) can be repeated, leading to what is called "iterative identification in closed loop and controller redesign" (Gevers, 1993; Landau, Lozano, M'Saad, & Karimi, 2011; Van den Hof & Schrama, 1995). This can be interpreted as "two-time scale" indirect adaptive control. The system operates with a constant controller during the identification stage over a time horizon, then the controller is re-tuned once the identification in closed loop is done and the identification procedure is re-started with the updated controller. The applicability of this methodology is also related to the possibility of embedding on a microprocessor used for electro-valve control. In this paper, an Arduino® Mega 2560 board is used. Note that evaluation of P, PI, and PID controllers in the context of electro-valves using an Arduino® board has been reported in Jadhav, Tahmnakar, and Kamble (2016), Supriyo, Tawi, Kob, and Mazali (2015).

The paper is organized as follows. Section 2 describes the test bench used for experiments. Section 3 provides an experimental analysis of the steady state and dynamic behaviors of the electric throttle valves, as well as the design of a basic model-based PI digital controller based on an experimentally identified linear model. The robust control design is discussed in Section 4. The real-time data-driven procedure based on identification in closed loop and controller redesign is presented in Section 5 along with experimental results.

2. Experimental test bench of the throttle valve

The picture and block diagram description of the test bench of the throttle valve are depicted in Figs. 1 and 2, respectively. The valve has the commercial reference 03L128063, a valve that equipped some Audi cars before 2010. This valve has only one spring (some other car engine throttle valves often considered in the literature are more complex, with a non-zero limp home position that results in a system with two hysteresis). This rotational spring is set on the shaft of the valve plate, exerting a torque that counteracts the motor's torque, thus resulting in a control of the angular position of the plate (torsional energy storage by the spring, which integrates the velocity). The opening angle is thus regulated by modulating the input voltage (driving the motor torque). This is done by pulse width modulation (PWM, ranging from 0% to 100%) of the voltage with an Arduino® Mega 2560 board, programmed with the *Arduino Integrated Development Environment*® (IDE) software. The valve is equipped with a programmable magnetic angle sensor KMA221 that provides an analog output ratio-metric to the supply voltage (set to 5 V to match the specification of Arduino®'s analog inputs). The analog-to-digital converter of the Arduino® board then converts this signal into a 10-bit one when using the command `analogRead()`.

The 12 V DC motor of the valve is controlled using the SHIELD-MD10 board, a Cytron® 10 A motor driver shield for Arduino®. This shield uses an NMOS H-Bridge to achieve a speed control PWM frequency of up to 10 kHz. The default PWM frequency is increased to avoid an annoying high-frequency noise generated by the valve (the free-running timer 3 of ATmega32 is modified using `TCCR3B`). The motor is controlled through two digital pins of the Arduino® board (selected by the mini jumpers on the shield): one for direction (`pinDIR`, set to `HIGH` or `LOW`) and one for the velocity (`pinPWM`, between 0 and 255).

A similar experimental test bench was initially developed as a research topic for the car industry (Hamze, 2019) and then redesigned for teaching purposes. The initial throttle valve, used for testing the adequacy of this device to teach linear control, is referred to as *Experiment 0* in the sequel. Once satisfactory results have been obtained, 10 other valves have been made available for building a control lab teaching equipment (7 of them are considered in this paper for comparison purposes). While these 10 extra valves have the same commercial reference as *Experiment 0*, it was neither the same manufacturer nor the same series. This has a serious impact on the experiments, because of the important dispersion of the valves' static and dynamic characteristics. The proposed strategy is thus to develop the algorithms for *Experiment 0* (which has a more predictable behavior) and test their efficiency on the other valves.

3. From non-linear dynamics to linear control

The experimental test bench for the throttle valve described in the previous section provides rich data sets to investigate nonlinear dynamics and varying time constants. Furthermore, major differences can be noticed when comparing the responses of different valves. These complexities are analyzed in this section, and frequency analysis shows that a linear behavior can still be captured and used to design a simple PI feedback controller.

3.1. Steady-state behavior and time constants

The steady-state behavior of the valves is investigated by applying a sequence of steps to the input voltages (increases and decreases of the PWM signals by 5). Each step is maintained during 2.5 s, as presented in Fig. 3(a), to ensure that each experimental test bench has enough time to reach the steady-state value. Fig. 3(b) depicts the superposed responses of eight different valves. The starting times and magnitudes of the different hysteresis vary significantly. The measured angle corresponding to the fully open position (when the PWM is set to 0) is 90° for *Experiment 0* (used for calibration) and 80° for the others, showing some calibration discrepancies. The angle reached at the maximum PWM input (40% here) also differs from valve to valve, probably due to different friction coefficients.

Repeating the same sequence of input steps multiple times on a given valve, as shown in Fig. 3(c), also results in a large variability

[1] i.e. an algorithm that on the one hand takes into account the effect of the output measurement noise through the input–output coupling introduced by the controller, and on the other hand introduces an appropriate weighting on the frequency distribution of the bias (identification error).

[2] The inverse of the maximum of the modulus of the sensitivity function defines the minimum distance between the hodograph of the open loop transfer function and the Nyquist instability point.

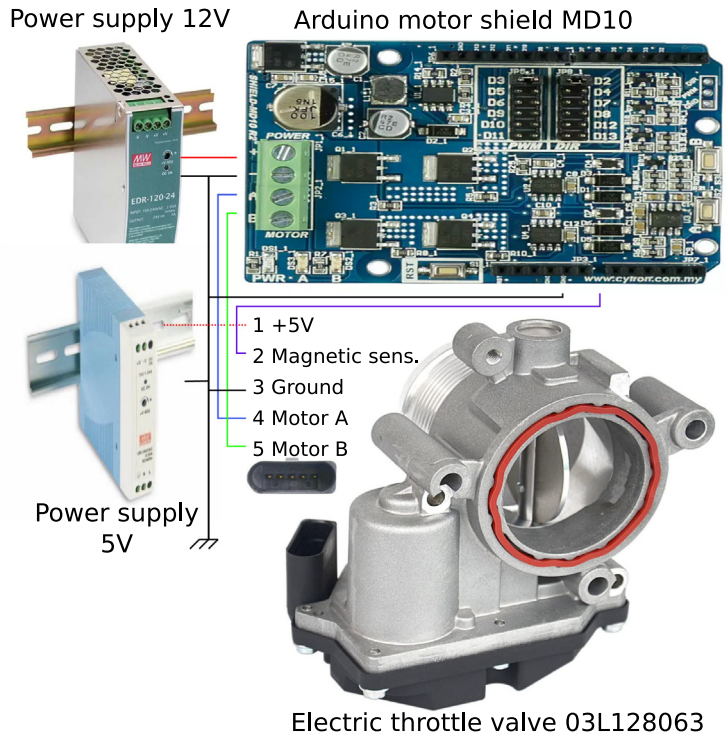


Fig. 1. Experimental test bench of the throttle valve: power supply and wiring.

of the hysteresis shape and the angles corresponding to the open or close positions. The use of a nonlinear model based on a hysteresis, as it is classically done in the modern automatic control literature, should thus be done carefully, as the hysteresis parameters vary largely from valve to valve and even during time for a given valve.

The response time of the valves is evaluated by zooming in on the first second following an input increase or decrease, and also has a large variability from valve to valve and for a given valve. The rise time t_R (to go from 10% to 90% of the final values) ranges from 0.2 to 0.8 s, approximately. The sampling time is chosen following the guidelines proposed in Landau and Zito (2006) (two to nine samples per rise time) as $T_s = 50$ ms ($f_S = 20$ Hz).

3.2. Frequency-domain analysis

The first step in modeling the process from a data set is to generate a signal that is sufficiently rich in terms of frequency content. This can be done efficiently, for example, by using a Pseudorandom Binary Sequence (PRBS) (Landau & Zito, 2006; Ljung, 1999) obtained from N_r shift registers with feedback. The PRBS design is performed here according to the method proposed in Landau and Zito (2006). As the longest step of the PRBS has to encompass the longest rise time, and as the PRBS length is limited by implementation issues, the following constraint is used:

$$pN_rT_s > t_R \tag{1}$$

where p is the ratio between the PRBS frequency and the sampling frequency. Choosing $p = 2$ and $N_r = 9$ one gets a PRBS of length $2^{N_r} - 1 = 511$, which is reasonable to implement as a pre-computed table in the Arduino® board (computed during the call of the `setup` function, before the `loop` function). Note that using a frequency divider $p = 2$, one gets a flat power spectrum density up to $0.35 f_S$ (see Landau and Zito (2006) pg. 233) well beyond the bandwidth of the valves.[3]

[3] This testing signal will also be used for system identification in a later stage.

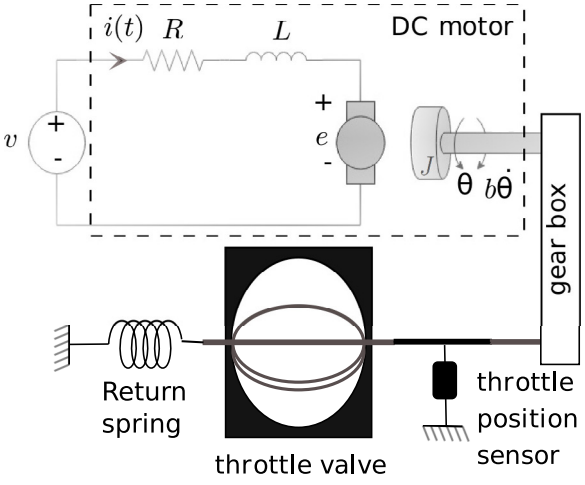
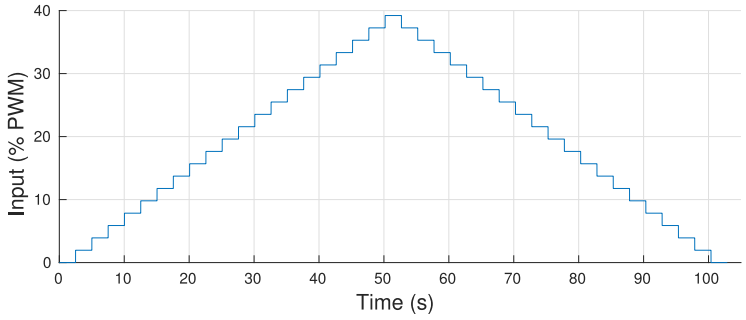


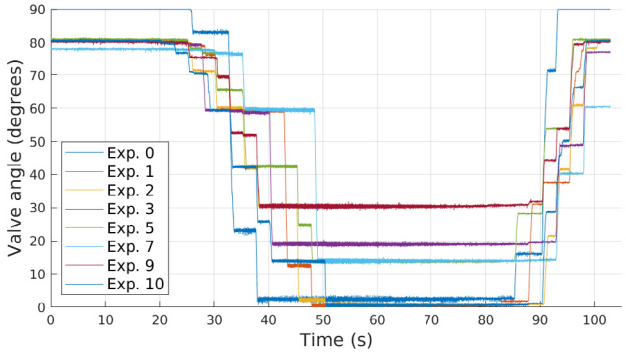
Fig. 2. Block diagram of the valve.

The PRBS is used to generate a PWM input signal centered around an operation point fixed at 16% of the full aperture, modulated by an amplitude between 10% and 14% that depends on the valve (the larger magnitude overcomes the Coulomb friction for all the valves). The frequency response of each valve is obtained by calculating the Empirical Transfer Function Estimate (ETFE, defined as the ratio between the discrete Fourier transforms of the output and the input sequences (Ljung, 1985)) for each experiment using Matlab®. The raw ETFE of *Experiment 0*, the smoothed ETFEs (by a Hann window of size 25, as suggested by Ljung (1999)) of all the experimental test benches, and the slope of −20 dB/dec are depicted in Fig. 4. One can note the following:

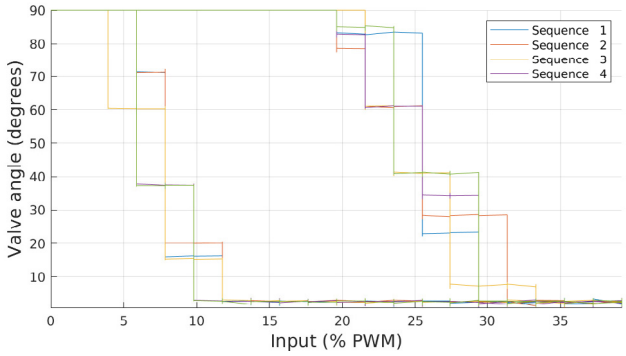
- the smoothed ETFE provides a reasonable approximation of the ETFE below 45 rad/s;



(a) Sequence of steps generated as an input PWM voltage.



(b) Responses of the different experiments to the same sequence of steps.



(c) Response of Experiment 0 to the same sequence of steps repeated multiple times: input-output map.

Fig. 3. Steady-state behavior analysis: time response of the valves to a sequence of steps.

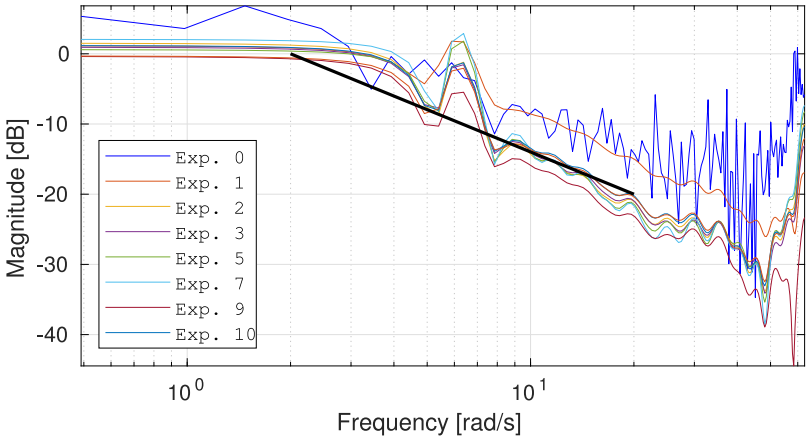


Fig. 4. Empirical Transfer Function Estimate (ETFE) of the valves' responses to a PRBS input. The raw ETFE of *Experiment 0* is depicted in dark blue while filtered values are shown in the other colors for all the valves. The black line follows a slope of −20 dB/dec, which captures the main transient dynamics of the valves.

- the ETFEs of the different experiments have similar behavior, except for *Experiment 0* which has a larger gain;
- a slope of −20 dB/dec reasonably approximates the ETFE slope, motivating the use of models with a single pole;
- the change of slope at 3 rad/s suggests a time constant of 0.5 s (consistent with the observed rise time).

3.3. Linear models

Consider a class of autoregressive models with exogenous inputs that write as

$$\begin{aligned} y(t) &= -a_1 y(t-1) - \cdots - a_{n_a} y(t-n_a) \\ &\quad + b_1 u(t-1) + \cdots + b_{n_b} u(t-n_b) \qquad (2) \\ &= \phi^T(t)\theta, \qquad (3) \end{aligned}$$

where $y(t)$ is the valve angle, $u(t)$ is the PWM input, n_a and n_b define the number of past samples (outputs and inputs, respectively) used to compute the actual output, and $\{a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}\}$ are constant parameters that form the *parameter vector*

$$\theta = [a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}]. \qquad (4)$$

The past data necessary to compute the value at t is stored in the regressor (measurement vector)

$$\begin{aligned} \phi(t) &= [-y(t-1), -y(t-2), \ldots, -y(t-n_a), \\ &\qquad u(t-1), \ldots, u(t-n_b)]^T. \qquad (5) \end{aligned}$$

The dimensions of the variables are $u(t), y(t) \in \mathbb{R}^1$, $\theta, \phi \in \mathbb{R}^n$, $n = n_a + n_b$, where $\mathbb{R}^n$ is the real n-dimensional Euclidean space.

The data sets used to estimate the model parameters are generated by the PRBS defined previously in Section 3.2. The predicted output at time t using the data set available at time $t-1$ is

$$\hat{y}(t|\hat{\theta}) = \phi^T(t)\hat{\theta},$$

where the vector containing the estimated parameters is

$$\hat{\theta} = [\hat{a}_1, \ldots, \hat{a}_{n_a}, \hat{b}_1, \ldots, \hat{b}_{n_b}]. \qquad (6)$$

The optimal set of parameters is obtained by minimizing the least-squares criterion

$$V_N(\hat{\theta}, Z^N) = \frac{1}{\bar{N}} \sum_{t=\bar{n}+1}^{N} \frac{1}{2} \left(y(t) - \phi^T(t)\hat{\theta}\right)^2, \qquad (7)$$

where $\bar{n} = \max\{n_a, n_b\}$, $\bar{N} = N - \bar{n}$, and Z^N represents the data set that contains the inputs and outputs for $t = 1, \ldots, N$ used for estimation.

Computing the criterion (7) for n_a and n_b varying between 1 and 3 provides the results presented in Fig. 5. Note that similar results (qualitatively) were obtained using Akaike's information criterion, a classical criterion to assess the suitable model order. The results on Fig. 5 show that, for most valves, having three free parameters ($n_a = 1$ and $n_b = 2$, or $n_a = 2$ and $n_b = 1$) decreases V in comparison with the $n_a = n_b = 1$ case. Nevertheless, the model with $n_a = n_b = 1$ still gives reasonable results except for *Experiment 0* (for which the least-squares criterion is doubled). Increasing the order of the denominator n_a (globally for every n_b) does not significantly decrease V, as expected from the ETFE analysis. Some valves (such as *Experiment 10*) can get more benefits from a more complex model than others (such as *Experiment 5* or *7*, where only a slight improvement can be noticed).

To be consistent with the frequency analysis and to simplify the control design, only linear models characterized by $n_a = 1$ and $n_b = 1$ are considered for the controller design. This result can be related to classical physical models of the valve. As discussed by Al-Samarraie and Abbas (2012) the DC motor impedance can be neglected, giving a linear relationship between the motor shaft velocity and the input voltage. The fact that a 1st order model gives good results implies that the impact of the rotary inertia of the motor can be neglected in comparison with the effect of the spring and the viscous damping. This could be expected, since the spring transforms the angular velocity of the motor shaft into the angular position of the throttle.

3.4. PI feedback control

As a reference feedback control, consider the digital implementation of a PI control that fulfills a pole placement objective on the closed-loop system (Åström & Wittenmark, 1984; Landau & Zito, 2006). The digital control proposed by Landau and Zito (2006) can then be used directly as

$$u(t) = u(t-1) - r_0 y(t) - r_1 y(t-1) + (r_0 + r_1) r(t), \qquad (8)$$

where $r(t)$ is the desired reference and r_0 and r_1 are the controller's gains. Introducing the unit delay operator q^{-1} such that $y(t-1) = q^{-1}y(t)$, the gains r_0 and r_1 are computed, using the identified model parameters and the desired denominator of the discrete closed-loop transfer function expressed as $1 + p_1 q^{-1} + p_2 q^{-2}$, with

$$r_0 = \frac{p_1 - \hat{a}_1 + 1}{\hat{b}_1} \quad \text{and} \quad r_1 = \frac{p_2 + \hat{a}_1}{\hat{b}_1}. \qquad (9)$$

A pole placement design specifying p_1 and p_2 can thus be directly implemented in the controller using (8)–(9). The closed-loop poles associated with the parameters p_1 and p_2 result from the choice of second-order dynamics having specified damping and time response (obtained for example using the diagrams in Landau et al. (2011)).

This controller is implemented on the experimental test benches as follows. As mentioned in Åström and Wittenmark (1984), Landau and Zito (2006) the speed of the response is mainly defined by the frequency of the complex poles when using pole placement design techniques. A damping ratio that gives a reasonably robust response is chosen ($\zeta = 1$ for the valve test bench), then the frequency corresponding to a desired response time is set. Two different response times are considered: one with a target $t_R = 0.8$ s and a more demanding control with $t_R = 0.4$ s. For each valve, r_0 and r_1 are computed using the specific values of a_1 and b_1 identified from the PRBS response of the corresponding valve. The controller's efficiency to track a reference is investigated for $r(t) = 40 \pm 25\,°$, thus over a large operating range. The results are depicted in Figs. 6–7, where the outputs are also compared with the dynamics expected from the closed-loop denominator $1 + p_1 q^{-1} + p_2 q^{-2}$. The closed-loop responses are particularly consistent between the valves, despite the previously discussed differences and nonlinearities. The responses also follow the expected closed-loop responses closely, especially for the higher-gain design ($t_R = 0.4$ s).

4. Robust control design

The objective of the PI feedback control (8)–(9) is to set the simplest digital PI controller for the throttle valve. It starts with the hypothesis that a first-order discrete-time model with $n_b = n_a = 1$ can be identified from data. It is, however, very important to discuss the robustness of the design concerning neglected dynamics and variations of the plant parameters. In fact, the first-order model is a rough approximation of reality (even in continuous time). In addition, some high-frequency dynamics (also called "parasitic" dynamics) are always present and should be taken into account in the identification and design stages (if this aspect is neglected in the identification stage, then hypotheses should be made upon the existence of neglected dynamics to take it into account in the design stage). This high-frequency dynamics can often be modeled as an additional fractional delay, which will lead to a model with $n_b = 2$ (see Sections 7.5.1–7.5.2 in Landau and Zito (2006) for details). Indeed, the previous identification results have shown that better model validation is obtained using $n_b = 2$ (with $b_2 < b_1$). This effectively indicates the presence of a fractional delay.

Independently considering a more accurate model, the robustness of the design concerning plant model uncertainties (neglected high-frequency dynamics, variations of the plant parameters) should be evaluated. The robustness of the design is assessed by examining the sensitivity functions in the frequency domain. For the specific problem considered, one can focus on two sensitivity functions: (1) the output

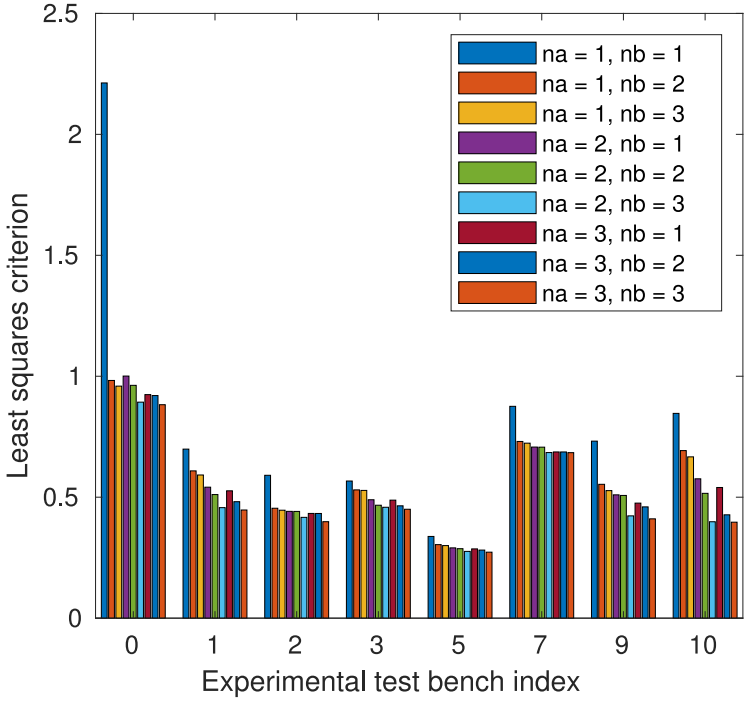
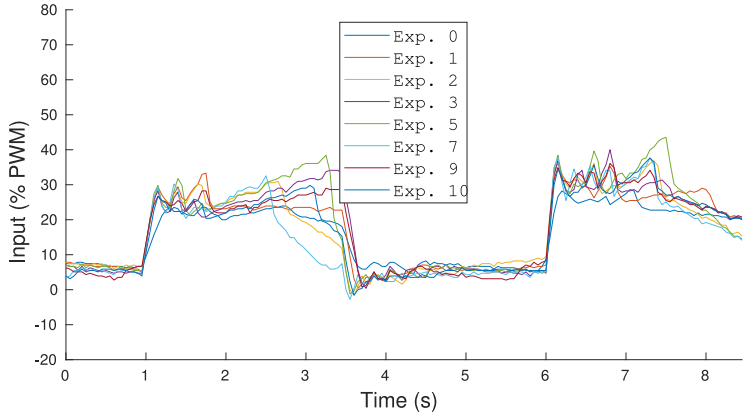
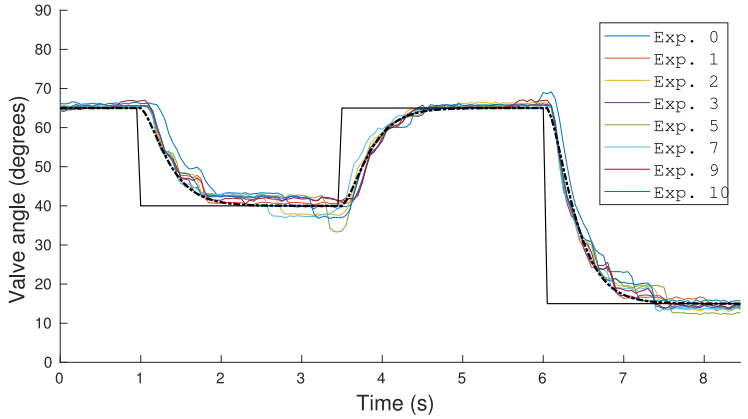


Fig. 5. Normalized least-squares criterion (7) for different combinations of the number of free parameters of the model (varying both the number of past outputs n_a and past inputs n_b used for the predicted output). The same computations are carried out for each valve and presented in parallel. While increasing the number of parameters improves the accuracy, the model obtained with $n_a = n_b = 1$ gives reasonable results for most of the valves.

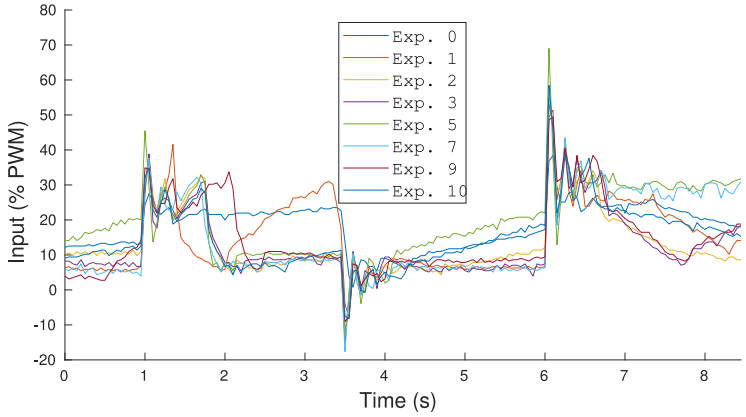


(a) Inputs for a target damping ratio of 1.0.

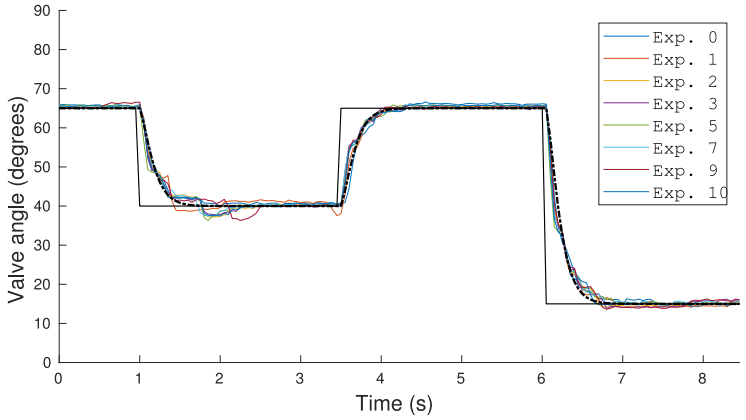


(b) Outputs for a target damping ratio of 1.0.

Fig. 6. Closed-loop responses of the 8 experimental test benches with PI feedback control for a desired rise time of 0.8 s. Tracked reference: continuous black line, expected output: dash-dot black line. Both control designs follow the expected closed-loop trajectories closely.



(a) Inputs for a target damping ratio of 1.0.



(b) Outputs for a target damping ratio of 1.0.

Fig. 7. Closed-loop responses of the 8 experimental test benches with PI feedback control for a desired rise time of 0.4 s. Tracked reference: continuous black line, expected output: dash-dot black line. Both control designs follow the expected closed-loop trajectories closely.

sensitivity function, and (2) the input sensitivity function. First, note that the linear dynamics described by (2) writes in the standard transfer operator form (Landau & Zito, 2006)

$$\frac{y(t)}{u(t)} = \frac{b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}} = \frac{B(q^{-1})}{A(q^{-1})}. \tag{10}$$

The digital controller canonical structure is considered with the RST formulation as

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})r(t). \tag{11}$$

Note that the discrete-time transfer functions are expressed in a complex frequency-domain ($z - domain$) by replacing q^{-1} with z^{-1}. With these notations, the *output sensitivity* function is defined as:

$$\begin{aligned} S_{yp}(z^{-1}) &= \frac{A(z^{-1})S(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})}, \\ &= \frac{A(z^{-1})S(z^{-1})}{P(z^{-1})}, \end{aligned}$$

where P defines the computed poles of the closed loop. The *input sensitivity* function is

$$\begin{aligned} S_{up}(z^{-1}) &= \frac{-A(z^{-1})R(z^{-1})}{A(z^{-1})S(z^{-1}) + B(z^{-1})R(z^{-1})}, \\ &= \frac{-A(z^{-1})R(z^{-1})}{P(z^{-1})}. \end{aligned}$$

The first robustness indicator is the "modulus margin" ΔM. It is the minimum distance between the instability point and the Nyquist plot of the open loop transfer function and is given by Landau and Zito (2006):

$$\begin{aligned} \Delta M &= |1 + H_{OL}(j\omega)|_{min} = |S_{yp}^{-1}(j\omega)|_{\min} \\ &= (|S_{yp}(j\omega)|_{\max})^{-1} \end{aligned}$$

Since one looks for a modulus margin:

$$\Delta M \geq 0.5 = -6\ dB \tag{12}$$

To ensure this condition, the controller should be designed such that:

$$|S_{yp}(j\omega)|_{max} \leq 6\ dB \tag{13}$$

To satisfy this condition for the simple PI control design, the dominant dynamics (dominant poles) have to be chosen by selecting a second-order system with a certain damping and resonance frequency and/or increasing the value of an auxiliary pole.

The *input sensitivity* function characterizes the tolerance to neglected dynamics and parameter variations (particularly in the high-frequency range). It can be shown (Landau & Zito, 2006) that for guaranteeing the closed loop stability, the tolerated additive uncertainty should satisfy the condition

$$|\frac{B'}{A'} - \frac{B}{A}| < |S_{up}^{-1}(j\omega)|, \tag{14}$$

where $\frac{B'}{A'}$ defines the perturbed plant model and $\frac{B}{A}$ is the plant model used for control design. This equation can be interpreted as follows: a good tolerance to additive plant uncertainties is obtained at the frequencies where $|S_{up}(j\omega)|$ is small, and conversely, a low tolerance to additive plant model uncertainties occurs at the frequencies where $|S_{up}(j\omega)|$ has a large value. Since the uncertainties are mainly located

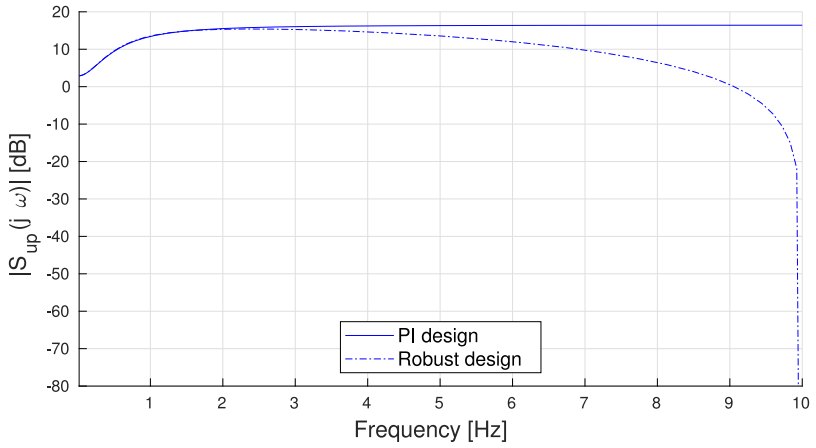


Fig. 8. Input sensitivity function for the robustness analysis ($a_1 = 0.9152; b_1 = -0.0609$). Plain lines: PI design setting the closed-loop behavior as a damped 2nd order dynamics; dashed lines: robust design when opening the loop at $0.5\,f_s$. High-frequency disturbances are removed from the control signal by the robust design.

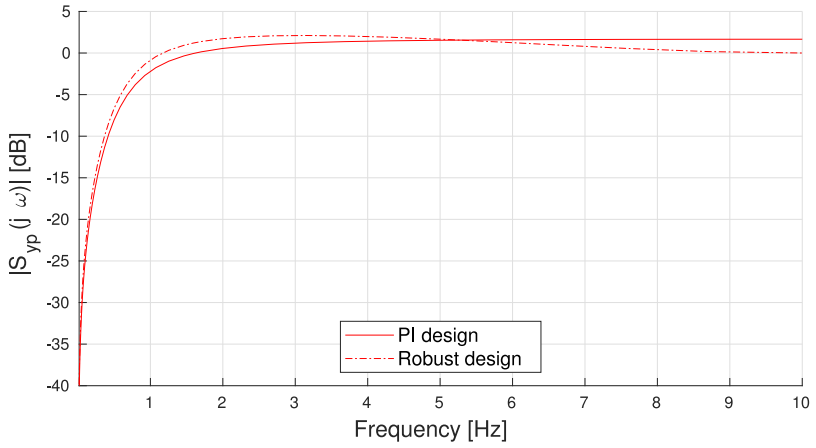


Fig. 9. Output sensitivity function. The performance of the robust design is close to the one of the PI design.

in the high frequency range, it is desirable to get the lowest possible value for $|S_{up}(j\omega)|$ in this frequency range.

Consider the design using the plant model with $a_1 = -0.9152$ and $b_1 = -0.0609$ (parameters identified for *Experiment 0*). The desired dominant poles are defined by a second-order system with $\omega_0 = 5$ and damping $\zeta = 1$ (resulting from a desired time response of 0.8 s). The corresponding controller parameters are $r_0 = -5.8719$ and $r_1 = 5.0685$. Fig. 9 shows the magnitude Bode plot of the output sensitivity function. The maximum of $|S_{yp}|$ is less than 6 dB: the modulus margin is thus larger than 0.5. Therefore the design is satisfactory from the point of view of the minimum distance to the Nyquist point.

Fig. 8 shows the magnitude Bode plot of the input sensitivity function. The value of $|S_{up}(j\omega)|$ is large in the high-frequency regions, which implies a low tolerance of the design concerning the neglected high-frequency dynamics. An improved design should lower as much as possible $|S_{up}(j\omega)|$ in the high frequency's region (equivalently, the controller should have a very low gain at high frequency). Unfortunately, with the actual complexity of the controller, there are not enough degrees of freedom to achieve both performance and a low value of the sensitivity $|S_{up}(j\omega)|$ at high frequency. To achieve this, a fixed part is added to the R polynomial of the controller

$$R(q^{-1}) = R'(q^{-1})(1 + q^{-1}),$$

where $R'(q^{-1})$ is the new polynomial that needs to be designed (to achieve the pole placement objective). The term $1+q^{-1}$ has a zero gain at $0.5\,f_s$ (f_s being the sampling frequency), and the input sensitivity is thus zero at this frequency. Similarly, the integral effect is included with the constraint

$$S(q^{-1}) = S'(q^{-1})(1 - q^{-1}),$$

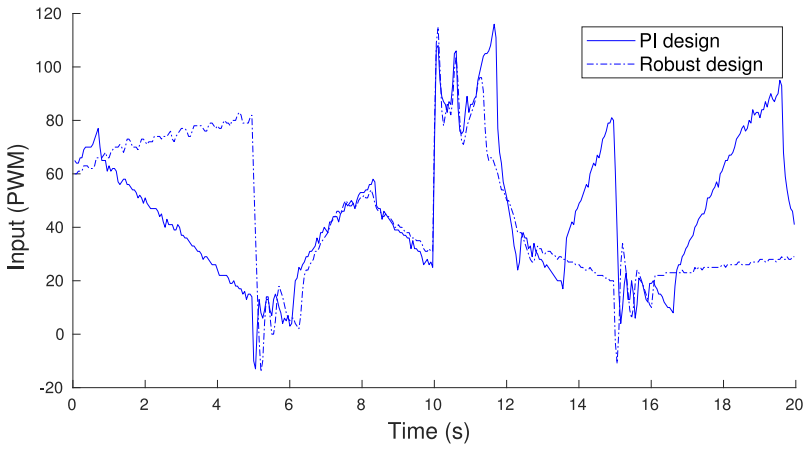
where $1 - q^{-1}$ sets the integral action and $S'(q^{-1})$ has to be designed.

To design this controller, the following Bezout equation needs to be solved

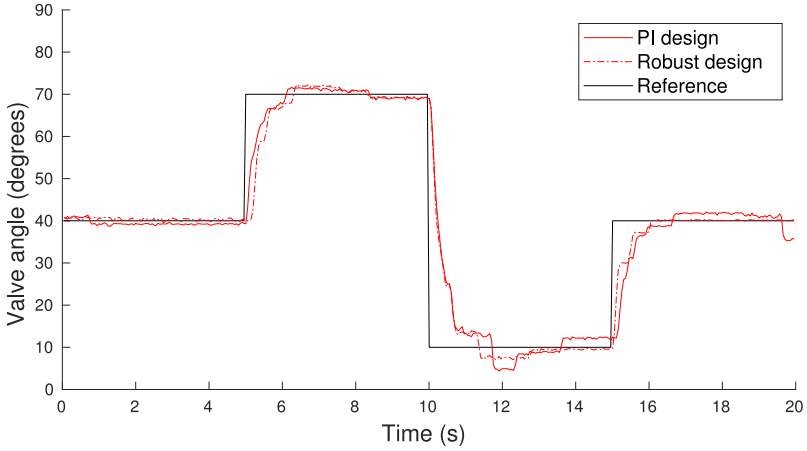
$$AS'(1 - q^{-1}) + BR'(1 + q^{-1}) = P = P_D P_F,$$

for given A, B, and P (decomposed into dominant poles P_D and auxiliary poles P_F that can be added to improve robustness), and the unknowns R' and S'. The resulting controller has the RST form (11).

Specifically for the previous example, if the loop is opened at $0.5 f_s$ and if the assigned dominant poles P_D ($P_F = 1$) do not change, the coefficients of the resulting controller are $r_0 = -3.0157$, $r_1 = -0.4017$, $r_2 = 2.6140$, and $s_0 = 1.0000$, $s_1 = -0.8261$, $s_2 = -0.1739$. The corresponding frequency characteristics of the output and input sensitivity functions are shown in Figs. 8–9 (dashed line). In comparison with the previous PI design (continuous line), the modulus of the new input sensitivity function goes towards 0 ($-\infty$ dB) at high frequencies close to $0.5\,f_s$, which is not the case when the loop is not opened at $0.5\,f_s$ (10 Hz). The influence on the output sensitivity function is minor (it



(a) Input signals.



(b) Output signals.

Fig. 10. Evaluation of the robust design performance on *Experiment 6*. Plain lines: PI design setting the closed-loop behavior as a damped 2nd order dynamics; dashed lines: robust design when opening the loop at $0.5\,f_s$. Robustness does not impair the controller's performance.

reaches 0 dB at 10 Hz, which indicates that the system will be in open loop at this frequency).

It is interesting to compare the two controllers in terms of performance. The simulated step responses of the two closed-loop systems (not shown) are indistinguishable; ideally, both controllers achieve the same desired pole placement. The controllers are evaluated in another experiment (*Experiment 6*) with the idea that, from an industrial perspective, one would want the controller to perform equally well on all the devices of the same brand. The experimental results of a tracking scenario are shown in Fig. 10. The robust controller mostly achieves better tracking, whether close to the linearization setpoint (40°) or far from it (references at 70° and 10°).

5. Real-time data driven control

The final objective of data-driven control is to use data acquired in closed-loop operations to improve the performance of the closed loop. This approach should also be able to take into account the possible variations of the plant parameters during operation, which may cause serious performance degradation.

The basic idea used in this paper is to estimate in real-time the parameters of the plant model and re-tune in real-time the parameters of the controller based on the current estimate of the plant parameters (without the designer in the loop!). If these two operations, parameter estimation and controller re-design, are done at each sampling time, one has a genuine indirect adaptive control (self-tuning) system (Åström & Wittenmark, 1995; Landau et al., 2011). If a time separation is introduced between these two operations (the adaptation system identifies the plant during a certain time horizon in the presence of a fixed controller and then, based on these estimated parameters, re-designs the controller and applies it), one has the technique called *iterative identification in closed loop and controller re-design* (Landau et al., 2011) (implemented at the end of this section on the experimental test bench). This technique can also be interpreted as a two-time scale indirect adaptive (self-tuning) control (Landau, 1999; Landau et al., 2011). However, to initialize the procedure in the absence of an initial controller, an identification in open loop operation is necessary, followed by the design of the controller based on the identified model (OLBC controller).

Two aspects have to be taken into account. (1) Since the system is operated in real-time, and to have an estimation of the plant model in real-time (as the plant evolves in time), one has to move from off-line open loop identification to online identification, which updates the parameters of the plant model at each sampling instant. To implement an online identification procedure, a recursive algorithm for plant model estimation is necessary.

(2) When doing *iterative identification in closed loop and controller re-design*, the identification is done in closed loop. In this context, there is feedback from the plant output to the plant input via the controller, and this alters the performance of open-loop identification algorithms.

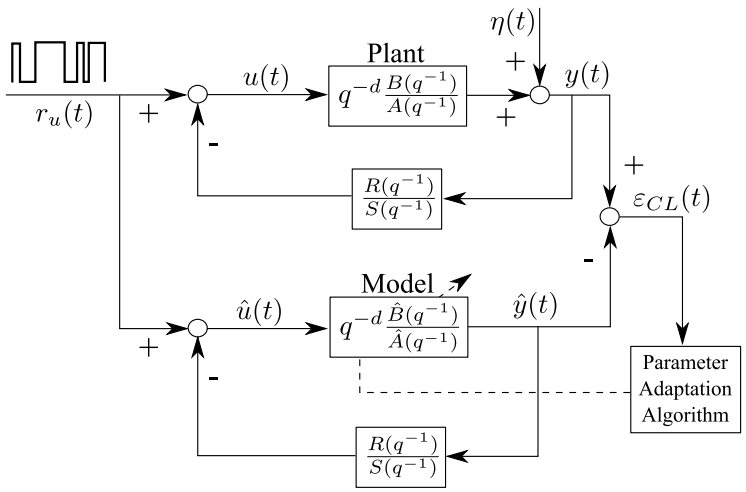


Fig. 11. Identification in closed loop. Excitation is superposed to control output.

In closed-loop operations, the identification paradigm (objective) is different from the open-loop case. The objective is to find a plant model that, when connected in feedback with the existing controller, provides the best model of the closed loop.

5.1. Recursive identification in closed loop operation

Using an open loop recursive identification algorithm in closed loop operation, in the presence of a fixed controller (a situation that is encountered in the *iterative identification in closed loop and controller redesign* method), does not provide, in general, a reliable model of the plant because one identifies the plant in closed loop with the controller. The identification paradigm calls for the identification of the plant model that gives the best prediction of the closed-loop output for a given controller. The principle of closed-loop output error identification algorithms is illustrated in Fig. 11. The upper part represents the true closed-loop system, and the lower part represents an adjustable predictor of the closed-loop. This closed-loop predictor uses the same controller as the one used on the real-time system.

The prediction error between the output of the real-time closed-loop system and the closed-loop predictor (closed-loop output error) is a measure of the difference between the true plant model and the estimated one. This error can be used to adapt the estimated plant model such that the closed-loop prediction error is minimized (in the sense of a certain criterion). In other words, the objective of the identification in closed-loop is to find the best plant model that minimizes the prediction error between the measured output of the true closed-loop system and the predicted closed-loop output. The use of these methods requires the knowledge of the controller. Considering the general case where the input can be delayed by d samples, the plant is described by

$$G(q^{-1}) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}, \tag{15}$$

where A and B are defined as in Eq. (10). The plant is operated in closed loop with an RST digital controller (without lack of generality). Introducing the polynomials $A^*(q^{-1})$ and $B^*(q^{-1})$ such that $A(q^{-1}) = 1 + q^{-1}A^*(q^{-1})$ and $B(q^{-1}) = q^{-1}B^*(q^{-1})$, the output of the plant operating in closed-loop is given by (see Fig. 11)

$$\begin{aligned} y(t+1) &= -A^*y(t) + B^*u(t-d) + A\eta(t+1) \\ &= \theta^T\varphi(t) + A\eta(t+1), \end{aligned} \tag{16}$$

where $u(t)$ is the plant input, $y(t)$ is the plant output, $\eta(t)$ is the output noise, θ is defined in (4) and

$$\begin{aligned} \varphi^T(t) &= [-y(t)\dots,-y(t-n_a+1), \\ &\quad u(t-d)\dots,u(t-n_b+1-d)], \end{aligned} \tag{17}$$

$$u(t) = -\frac{R}{S}y(t) + r_u, \tag{18}$$

where r_u is the external excitation added to the output of the controller (input of the plant) for identification purposes.

The *a priori* predictor of the closed-loop can be expressed as

$$\hat{y}^\circ(t+1) = -\hat{A}^*(t)\hat{y}(t) + \hat{B}^*(t)\hat{u}(t-d) = \hat{\theta}^T(t)\phi_d(t), \tag{19}$$

where $\hat{\theta}$ is defined in (6) and

$$\begin{aligned} \phi_d^T(t) &= [-\hat{y}(t)\dots,-\hat{y}(t-n_a+1), \\ &\quad \hat{u}(t-d)\dots,\hat{u}(t-n_b+1-d)] \end{aligned} \tag{20}$$

$$\hat{u}(t) = -\frac{R}{S}\hat{y}(t) + r_u, \tag{21}$$

The a posteriori predictor of the closed loop can be expressed as

$$\hat{y}(t+1) = \hat{\theta}^T(t+1)\phi_d(t). \tag{22}$$

The *a priori* closed-loop prediction (output) error is defined as

$$\varepsilon^\circ_{CL}(t+1) = y(t+1) - \hat{y}^\circ(t+1), \tag{23}$$

and the *a posteriori* closed-loop prediction error is defined as

$$\varepsilon_{CL}(t+1) = y(t+1) - \hat{y}(t+1). \tag{24}$$

The parameter adaptation algorithm has the form

$$\hat{\theta}(t+1) = \hat{\theta}(t) + F(t)\phi(t)\epsilon_{CL}(t+1) \tag{25}$$

$$\begin{aligned} F(t+1)^{-1} &= \lambda_1(t)F(t)^{-1} \\ &+ \lambda_2(t)\phi(t)\phi^T(t) \end{aligned} \tag{26}$$

$$0 < \lambda_1(t) \le 1\ ;\ 0 \le \lambda_2(t) < 2\ ;\ F(0) > 0 \tag{27}$$

$$F(t+1) = \frac{1}{\lambda_1(t)}\left[F(t) - \frac{F(t)\phi(t)\phi^T(t)F(t)}{\frac{\lambda_1(t)}{\lambda_2(t)} + \phi^T(t)F(t)\phi(t)}\right] \tag{28}$$

$$\epsilon_{CL}(t+1) = \frac{y(t+1) - \hat{\theta}^T(t)\phi(t)}{1 + \phi^T(t)F(t)\phi(t)} \tag{29}$$

Note that $\lambda_1(t)$ and $\lambda_2(t)$ in (27) have opposite effects on the adaptation gain. $\lambda_1(t) < 1$ tends to increase the adaptation gain (the inverse of the gain decreases) while $\lambda_2(t) > 0$ tends to decrease the adaptation gain (the gain inverse increases). For each choice of sequences $\lambda_1(t)$ and $\lambda_2(t)$ corresponds a *variation profile* of the adaptation gain, and an

interpretation can be inferred in terms of the error criterion (which is minimized by the parameter adaptation algorithm). In particular, to estimate time-varying parameters, one should avoid having an adaptation gain that tends towards 0 (which corresponds to the case when $\lambda_1(t) = \lambda_2(t) = 1$). One of the most commonly used options, both for identification of systems with constant parameters and with slowly time-varying parameters, is the so-called "variable forgetting factor". In this case

$$\lambda_2(t) = \lambda_2 = 1, \tag{30}$$

and the forgetting factor $\lambda_1(t)$ is given by

$$\lambda_1(t) = \lambda_0\lambda_1(t-1) + 1 - \lambda_0 \; ; \; 0 < \lambda_0 < 1. \tag{31}$$

The typical ranges for the initial and constant terms are

$$\lambda_1(0) = 0.95 \text{ to } 0.99 \; ; \; \lambda_0 = 0.5 \text{ to } 0.99$$

(the frequently used values are $\lambda_1(0) = \lambda_0 = 0.97$). $\lambda_1(t)$ can be interpreted as the output of a first-order filter $\left(1-\lambda_0\right) / \left(1-\lambda_0 q^{-1}\right)$ with a unitary steady state gain and an initial condition $\lambda_1(0)$. Relation (31) leads to a forgetting factor that asymptotically converges towards 1 (decreasing adaptation gain). This type of profile, when used for the model identification of stationary systems, avoids a too rapid decrease of the adaptation gain, thus generally resulting in an acceleration of the convergence (by maintaining a high gain at the beginning when the estimates are far from the optimum values).

This algorithm is called the *closed-loop output error* (CLOE) (Landau et al., 2011). For this algorithm the frequency distribution of the bias error (identification error) is given by Landau et al. (2011)

$$\hat{\theta}^* = \arg\min_{\hat{\theta}\in D} \int_{-\pi}^{\pi} |S_{yp}|^2 [|G-\hat{G}|^2 |\hat{S}_{yp}|^2 \phi_{r_u}(\omega) + \phi_v(\omega)] d\omega \tag{32}$$

where G is the transfer function of the plant, $\hat{G}$ is the estimated transfer function, $\phi_{r_u}(\omega)$ is the spectrum of the excitation signal, $\phi_v(\omega)]$ is the spectrum of the measurement noise and $\hat{S}_{yp}$ is the estimated output sensitivity function. This expression shows that:

- the bias distribution is not only weighted by the sensitivity function but is further weighted by the estimated sensitivity function (i.e., the best approximation of the true plant transfer function is obtained in the critical frequency region closest to the Nyquist point);
- the estimation of the plant model parameters is unbiased when G is in the model set (i.e., the order of the plant model is equal to the order of the estimated plant model);
- the bias distribution is not affected by the spectrum of the measurement noise.

5.2. Iterative re-design of the controller

As indicated at the beginning of this section, in the technique of iterative closed-loop identification and controller re-design, after an identification in closed loop during a certain time horizon in the presence of an external excitation, the estimated plant parameters are used to re-design the controller. The control design and the desired performance are the same as those used in the case when a model identified in an open loop has been used. A criterion for performance evaluation has to be defined, and the procedure can be stopped if the performance has not improved after several iterations. Reported experimental results (see Landau et al. (2011), Landau and Zito (2006)) indicate that the first two iterations are those that lead to the most significant performance improvement.

Table 1
Relative performance of the control designs.

	Robust	PI	Adaptive iteration				
			1st	2nd	3rd	4th	5th
J_ϵ:	100	111	175	103	94	93	92

5.3. Experimental results for iterative identification in closed loop and controller re-design

The method for iterative identification in closed loop and controller re-design is implemented on the throttle valve as follows:

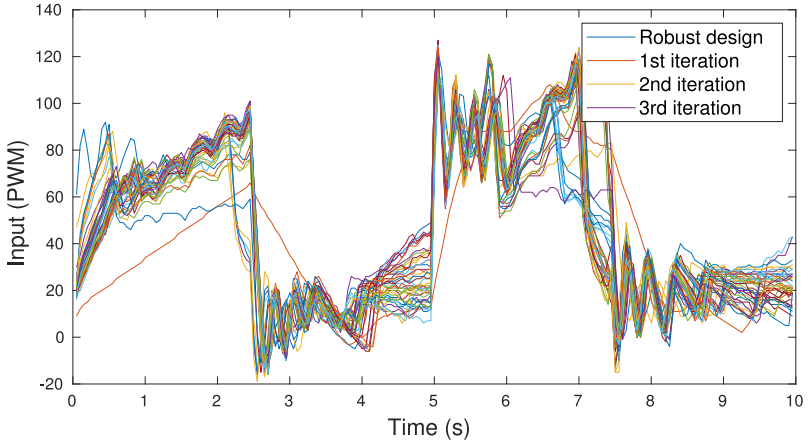
1. the initial RST controller design is done according to the robust method described in Section 4 using a generic (known *a priori*) set of parameters θ;
2. a persistently exciting signal (PRBS) is applied (added to the plant input) during a specific period and the vector of system parameters $\hat{\theta}$ is evaluated using the CLOE algorithm;
3. the tuning of the robust RST controller is updated using the estimated vector of parameters $\hat{\theta}$ and implemented in the closed loop;
4. the previous two steps are repeated in a loop.

The resulting algorithm is developed with Arduino IDE® and uploaded in the Arduino Mega 2560® board. This program uses only 5% of the program's storage space, and the global variables use 7% of the program's dynamic memory. The desired sampling time is mostly met despite the computation load. Comparing the different algorithm steps with the same calculations performed with Matlab®, minor numerical discrepancies are noticed. This could be expected from the matrix operations involved in the design and may be of particular interest to investigate further the importance of robust methods in a real-time embedded environment. More robust parameter estimation is obtained by increasing the power spectrum of the PRBS at lower frequencies (setting the duration of the longest pulse of the PRBS to 1.6 s instead of 0.8 s). The excitation signal is set to 300 samples, which does not include the full PRBS but gives a satisfactory parameter convergence. No assumption is made on the initial knowledge of the parameters and the algorithm is started with $\hat{\theta}(0) = 0$. Each sequence of parameter estimation and controller update is followed by an evaluation of the tracking efficiency, with the scenario depicted in Fig. 12. Comparing the tracking efficiency for different iterations on Fig. 12, consistent closed-loop responses and more noticeable improvement (compared to the robust PI design) can be noticed at the 4th iteration.

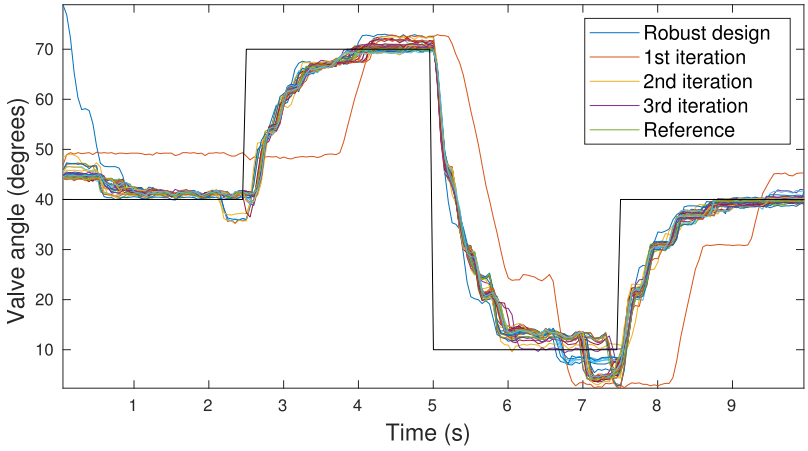
The different results can be compared with the criterion of the form $J_\epsilon = \sum_{t=0}^{N} \epsilon^2(t)$, where $\epsilon(t)$ is the difference between the reference and the plant output. The normalized results are summarized in Table 1. While the first controller redesign increases the error function, J_ϵ decreases by almost 10% after a few iterations. The evolution of the controller's gains is presented in Fig. 13. This result shows that the gains reach their final values after a few iterations. One can also notice that the gains are close to those of the robust method, even though a different (recursive) algorithm is implemented and much fewer computing resources are available. The slight increase in control authority may be due to the need to compensate for the nonlinear friction effects, which are implicitly detected by the adaptive method with online tuning.

6. Conclusions

Despite the observed complexity of the electric throttle valve dynamics, it was shown that a linear design in the discrete-time framework offers a valid solution to the control problem. A relatively low number of samples was used (less than 26 s of measurement). A more complex design that takes into account a robustness objective is also



(a) Input signals.



(b) Output signals.

Fig. 12. Performance evaluation of the adaptive method *iterative identification in closed loop and controller re-design* on *Experiment 6*. The response corresponding to the *Robust control design* is compared with successive iterations of the adaptive design. More accurate tracking is obtained with the adaptive method after a few iterations.

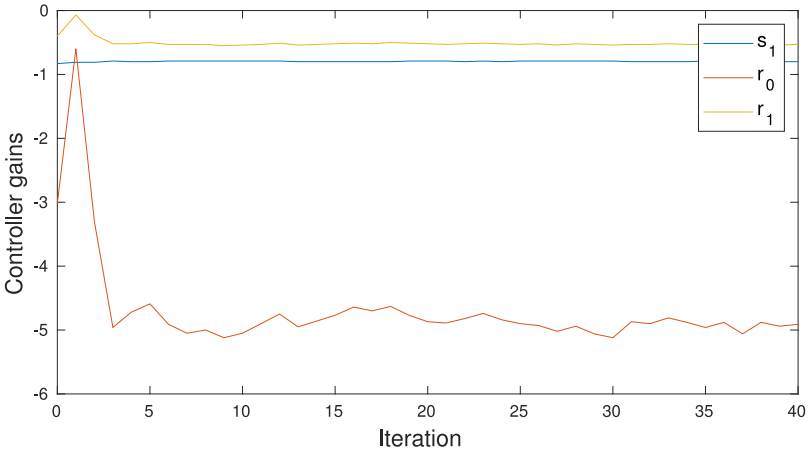


Fig. 13. Evolution of the controller's gains during the iterations of the adaptive method.

presented. The last control design involves a real-time data-driven method in which the controller learns the process parameters online and updates the feedback gains when new measurements are received. Despite the number of real-time calculations associated with this procedure, the algorithm has been successfully embedded on a low-cost Arduino® microprocessor board.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank Dr. Vincent Talon for his numerous insights and valuable discussions on throttle valve control for car engines, and the numerous UGA students who contributed to this experimental test bench during their project class for the Master in Systems, Control and Information Technologies. They also thank the reviewers and editor for their fruitful and constructive comments, which significantly improved the final version of this work.

References

Al-Samarraie, S. A., & Abbas, Y. K. (2012). Design of electronic throttle valve position control system using nonlinear PID controller. *International Journal of Computer Applications*, *59*(15), 27–34.

Åström, K. J., & Wittenmark, B. (1984). *Computer controlled systems, theory and design*. Englewood Cliffs, N.J.: Prentice-Hall.

Åström, K. J., & Wittenmark, B. (1995). *Adaptive control* (2nd Ed.). Addison-Wesley.

Callan, P. C., & Eggenberger, M. A. (1965). Basic analysis of pressure-control systems used on large steam turbine-generator units. *Journal of Engineering for Power*, *87*(4), 389–400.

Canudas de Witt, C., Kolmanovsky, I., & Sun, J. (2001). Adaptive pulse control of electronic throttle. In *Proceedings of the 2001 american control conference, Vol. 4* (pp. 2872–2877).

Cassidy, J., Athans, M., & Wing-Hong, L. (1980). On the design of electronic automotive engine controls using linear quadratic control theory. *IEEE Transactions on Automatic Control*, *25*(5), 901–912.

Gevers, M. (1993). Towards a joint design of identification and control. In H. L. Trentelman, & J. C. Willems (Eds.), *Essays on control: perspectives in the theory and its applications* (pp. 111–152). Boston, U.S.A.: Birkhäuser.

Hamze, S. (2019). *Model-based system engineering for powertrain systems optimization, no. 2019GREAT055* (Ph.D. thesis), Université Grenoble Alpes.

Hu, Y., Wang, H., He, S., Zheng, J., Ping, Z., Shao, K., et al. (2021). Adaptive tracking control of an electronic throttle valve based on recursive terminal sliding mode. *IEEE Transactions on Vehicular Technology*, *70*(1), 251–262.

Jadhav, R. B., Tahmnakar, S. G., & Kamble, P. (2016). Throttle by wire using embedded environment. *International Journal of Innovative Research in Science, Engineering and Technology*, *5*(9).

Jiao, X., Zhang, J., & Shen, T. (2014). An adaptive servo control strategy for automotive electronic throttle and experimental validation. *IEEE Transactions on Industrial Electronics*, *61*(11), 6275–6284.

Johnson, R. A., & Thompson, F. T. (1965). Throttle valve position control system. *IEEE Transactions on Industry and General Applications*, *IGA-1*(3), 199–205.

Karimi, A., Bahrani, B., Zheng, D., & Madani, S. S. (2022). Data deiven conrol and its application to microgrids. In *Proc. IEEE CCTA 2022, conference on control technology and application*. Trieste, Italy.

Kwatny, H., & Fink, L. (1975). Acoustics, stability, and compensation in boiling water reactor pressure control systems. *IEEE Transactions on Automatic Control*, *20*(6), 727–739.

Landau, I. D. (1998). Competition, interaction and control. In D. Normand-Cyrot (Ed.), *Perspectives in control* (pp. 309–340). Berlin, Germany.: Springer-Verlag.

Landau, I. D. (1999). From robust control to adaptive control. *Control Engineering Practice*, *7*(9), 1113–1124.

Landau, I. D., Lozano, R., M'Saad, M., & Karimi, A. (2011). *Communications and control engineering, Adaptive control: algorithms, analysis and applications*. Springer.

Landau, I., & Zito, G. (2006). *Communications and control engineering, Digital control systems: design, identification and implementation*. Berlin, Heidelberg: Springer-Verlag.

Ljung, L. (1985). On the estimation of transfer functions. *IFAC Proceedings Volumes*, *18*(5), 1653–1657.

Ljung, L. (1999). System identification: Theory for the user, *Information and system sciences*, (2nd ed.). Upper Saddle River, NJ: PTR Prentice Hall.

Panzani, G., Corno, M., & Savaresi, S. M. (2013). On adaptive electronic throttle control for sport motorcycles. *Control Engineering Practice*, *21*(1), 42–53.

Pujol, G., Vidal, Y., Acho, L., & Vargas, A. N. (2016). Asymmetric modelling and control of an electronic throttle. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *29*(2), 192–204.

Supriyo, B., Tawi, K., Kob, M., & Mazali, I. (2015). Arduino based electro-mechanical throttle controller for automotive applications. *ARPN Journal of Engineering and Applied Sciences*, *10*, 7769–7772.

Van den Hof, P., & Schrama, R. (1995). Identification and control – closed-loop issues. *Automatica*, *31*(12), 1751–1770.

Vargas, A. N., Acho, L., Pujol, G., Oliveira, R. C. L. F., do Val, J. B. R., & Peres, P. L. D. (2014). Robust H_2 static output feedback to control an automotive throttle valve. In *2014 american control conference* (pp. 3141–3146).

Ye, M., & Wang, H. (2020). A robust adaptive chattering-free sliding mode control strategy for automotive electronic throttle system via genetic algorithm. *IEEE Access*, *8*, 68–80.

Zhang, S., Yang, J. J., & Zhu, G. G. (2015). LPV modeling and mixed constrained H_2/H_∞ control of an electronic throttle. *IEEE/ASME Transactions on Mechatronics*, *20*(5), 2120–2132.