13

Model-free Adaptive Control in Frequency Domain: Application to Mechanical Ventilation

Clara Ionescu and Robin De Keyser
Ghent University, Department of Electrical energy, Systems and Automation
Belgium

1. Introduction

Looking back at the history of control engineering, one finds that technology and ideas combine themselves until they reach a successful result, over the timeline of several decades (Bernstein, 2002). It is such that before the computational advances during the so-called Information Age, a manifold of mathematical tools remained abstract and limited to theory. A recent trend has been observed in combining feedback control theory and applications with well-known, but scarcely used in practice, mathematical tools. The reason for the failure of these mathematical tools in practice was solely due to the high computational cost. Nowadays, this problem is obsolete and researchers have grasped the opportunity to exploit new horizons.

During the development of modern control theory, it became clear that a fixed controller cannot provide acceptable closed-loop performance in all situations. Especially if the plant to be controlled has unknown or varying dynamics, the design of a fixed controller that always satisfies the desired specifications is not straightforward. In the late 1950s, this observation led to the development of the gain-scheduling technique, which can be applied if the process depends in a known or measurable way on some external, measurable condition (Ilchmann & Ryan, 2003). The drawback of this simple solution is that only static (steady state) variations can be tackled, so the need for dynamic methods of controller (re)tuning was justified.

One can speak of three distinct features of the standard PID controller tuning: auto-tuning, gain scheduling and adaptation. Although they use the same basic ingredients, controller auto-tuning and gain scheduling should not be confused with adaptive control, which continuously adjusts controller parameters to accommodate unpredicted changes in process dynamics. There are a manifold of auto-tuning methods available in the literature, based on input-output observations of the system to be controlled (Bueno *et al.*, 1991; Åström & Hagglund, 1995; Gorez, 1997).

The tuning methods can be classified twofold:

- direct methods, which do not use an explicit model of the process to be controlled; these can then be either based on tuning rules (Åström & Hagglund, 1995), either on iterative search methods (Åström & Wittemark, 1995; Gorez, 1997).
- indirect methods, which compute the controller parameters from a model of the process to be controlled, requiring the knowledge of the process model; these can be based on

either models: transient-response models (step response model), frequency response models, or transfer function models.

Although many adaptive control methods are available in the literature, their implementation in practice is challenging and prone to failures (Anderson, 2005). If the process parameters are not known and not necessary, direct adaptive methods can be derived based on specifications of the closed loop performance, using explicit reference models. A relatively large gap exists between theoretical and practical model-reference adaptive control, initiated from the unknown process (Butler, 1990). Nevertheless, direct adaptive control with model reference has proved successful for a variety of applications, some of which will be presented in this contribution.

This chapter will present a simple and straightforward adaptive controller strategy from the class of direct methods, based on reference models. The algorithm will offer an alternative solution to the burden of process identification, and will present possibilities to tune both integer- and fractional- order controllers. Three examples will illustrate the simplicity of the approach and its results. A discussion section will provide advantages and dis-advantages of the proposed algorithm and some implementation issues. A conclusion section will summarize the outcome of this investigation.

2. Methods

2.1 The DIRAC principle

The DIRAC (DIrect Adaptive Controller) algorithm belongs to the class of *model-free* tuning methods, since it does not require the knowledge of the process, nor it needs to identify it during the tuning procedure (De Keyser, 1989; De Keyser, 2000). The most important feature in this model reference adaptive control strategy is the design of the adaptive laws, which take place directly, without an explicit process identification procedure. The aim of making the closed loop response approximately equal to a specified response is the key ingredient of DIRAC, and the design of this reference model plays a decisive role. The perfect model matching condition places some requirements on the reference model, which results in the following rule-of-thumb: the relative degree (pole excess) must be equal to the relative degree of the process; however, as shown here, this condition can be avoided. In addition to the stability and minimum-phase demands on the reference model, these are just theoretical aspects. It should be noted that in practice, where some theoretical requirements may not be satisfied, the reference model should be chosen reasonably, in the sense that the process output can actually follow the reference model output. For example, if the reference model is chosen too fast (compared to the process dynamics), the control signal needs to be extremely high, causing input saturation effects or nonlinear dynamics which may disturb the overall closed loop behaviour.

Because the actual process capabilities may be unknown or varying, the choice of the reference model is not always obvious. Choosing a conservative performance may be more robust, but it may also lead to slower closed loop behaviour than necessary.

In the standard control loop, where $C(s)$ denotes the controller, $P(s)$ denotes the (unknown) process, $w(t)$ is the reference signal $y(t)$ is the output, $e(t)=y(t)-w(t)$ is the error, the closed loop transfer function is given by:

$$y(t) = \frac{C(s)P(s)}{1 + C(s)P(s)} w(t) \tag{1}$$

The desired closed loop performance will be then specified by a reference model $R(s)$, a-priori user-defined, which can be used to specify the desired characteristics of the loop, for instance, the speed (bandwidth). It follows that the tuning task can be summarized as follows: find the corresponding controller's parameters (PID or any other transfer function) such that the closed-loop transfer function is more or less equal to the reference model:

$$\frac{C(s)P(s)}{1+C(s)P(s)} \cong R(s) \tag{2}$$

The trivial solution arising from solving (2) for the unknown controller will lead to un-desired results, such as: i) identification of the unknown process (which is not aimed), and ii) the result will lead to a transfer function for $C(s)$ and not to a 2nd order polynomial, which is required for obtaining a PID controller in the form:

$$C(s) = \frac{1}{s}\underbrace{\left(c_0 + c_1 s + c_2 s^2\right)}_{C^*(s)} \tag{3}$$

explicitly containing an integrator to ensure zero steady state error. In order to avoid this dead-end solution, one can extract the controller from (2) taking into account the measurable signals u(t) and y(t) and the relation $P \cdot u(t) = y(t)$:

$$C^* \cdot y_f(t) + \varepsilon(t) = u_f(t) \tag{4}$$

with $u_f(t)$ and $y_f(t)$ obtained as in schematically depicted in figure 1.

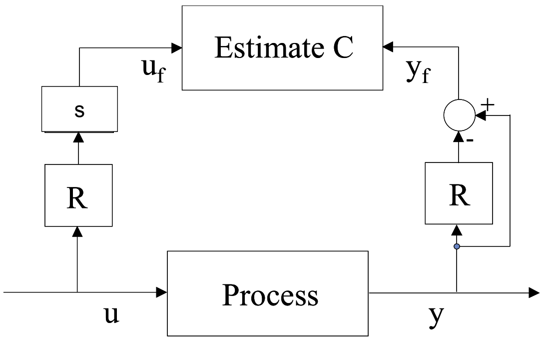

Figure 1. Block-scheme of the DIRAC strategy

Relation (4) becomes then a standard identification problem which can be solved either offline (tuning), either online (adaptation) using any parameter estimation method (Ljung, 1987). It should be noted that the least squares gives unbiased estimates even in the case of coloured noise, since (4) does not contain recursion of $u_f(t)$. Some guidelines to define the reference model for various classes of processes are given in (De Keyser, 2000), discussing the implementation aspects in discrete time, along with some typical examples.

2.2 Controller's structure

From the previous section we have concluded that the controller has to satisfy the equality:

$$R(s) \cong \frac{C(s)P(s)}{1+C(s)P(s)} \cong \frac{C^*(s)P(s)}{s+C^*(s)P(s)} \tag{5}$$

from which the *utopic controller* can be extracted as:

$$C^*(s) \cong \frac{sR(s)}{\left[1-R(s)\right]P(s)} \tag{6}$$

Obviously, the standard 'textbook' PID transfer function $K_p\left(1+\frac{1}{T_i s}+T_d s\right)$ consisting of the proportional, integral and derivative terms is the most used in practice, yielding satisfactory results. If the PID controller is written in the form required by (6), it results in estimating a 2nd order polynomial:

$$C^*(s) \cong c_0 s + c_1 + c_2 s^2 \tag{7}$$

with $c_0 = K_p$, $c_1 = K_p / T_i$ and $c_2 = K_p T_d$ the three unknown parameters to be identified.
Yet, the end of the 20th century has brought numerous advances in technology, with visible improvements in the computational aspects and pushing onward the limits of numerical complexity. As a result, mathematical tools which were abstract and numerically too complex for practical usefulness, were enabled as powerful tools for identification and control. As a result, the control engineering research community has oriented its attention to the possibility of using non-integer order controllers, namely $PI^\lambda D^\mu$ (Monje *et al.*, 2008). Such a controller is in fact a generalization of the standard PID, $K_p\left(1+\frac{1}{T_i s^\lambda}+T_d s^\mu\right)$, and (7) can be re-written as:

$$C^*(s) \cong c_0 s + c_1 s^{1-\lambda} + c_2 s^{1+\mu} \tag{8}$$

with $c_0, c_1, c_2, \lambda, \mu$ five unknown parameters to be identified. To estimate the parameters in (7), a linear identification algorithm suffices to obtain good results, such as the linear least squares method (Ljung, 1987). For (8), however, we are dealing with a polynomial which is nonlinear in the parameters and it is necessary to use nonlinear identification methods, such as nonlinear least squares (Ljung, 1987).
Simulation in time domain for fractional order controllers (FOC) such as the one described by (3) with the controller structure from (8), may be challenging. There are several definitions of the differ-integral in time domain, of which two commonly used are the Grünwald-Letnikov and Rieman-Liouville definitions (Podlubny, 1999):

$$\textbf{GL:}\ {}_{a}D_t^\alpha f(t) = \lim_{h \to 0} h^{-\alpha} \sum_{j=0}^{\left[(t-a)/h\right]} (-1)^j \binom{\alpha}{j} f(t-jh) \tag{9}$$

where $[\ ]$ denotes the integer part, respectively:

$$\textbf{RL:}\ {}_{a}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(1-\tau)^{\alpha-n+1}} d\tau \tag{10}$$

for $n-1<\alpha<n$ and Γ is the Euler's Gamma function; a denotes the initial conditions, t is the differ-integration time, α is the fractional order. The Laplace transform of the Rieman-Liouville fractional derivative/integral (10) under zero initial conditions can be written as:

$$\pounds\left\{D_t^{\pm\alpha} f(t)\right\} = s^{\pm\alpha} F(s) \tag{11}$$

With (11) at hand, the interpretation of fractional order derivative/integral can be simplified using the complex plane representation and Bode characteristics, the latter defined by its magnitude and phase. In this line-of-thought, $s^{\pm\alpha}$ becomes $(j\omega)^{\pm\alpha}$ in frequency domain, with $j=\sqrt{-1}$ and ω (rad/s) the angular frequency. The Bode plot can be then defined as:

$$\begin{aligned} \left|s^{\pm\alpha}\right| &= \pm\alpha \cdot 20dB/dec \\ \angle(s^{\pm\alpha}) &= \pm\alpha \cdot \frac{\pi}{2} rad \end{aligned} \tag{12}$$

It is now easy to understand why FOC is so interesting from identification/control standpoint: its intrinsic capability to capture variations in frequency domain which are not limited to integer-multiples of 20dB/dec, or π/2, respectively (such as for integer order systems).

If the controller is in the form of (7), it results directly in the transfer function of the PID with the integrator added explicitly to the 2nd order polynomial from (7). If the controller is in the form of (8), namely fractional order controller FOC, then an extra step is necessary to be implemented before being able to simulate the closed-loop behavior. The reason is that fractional order controllers cannot be yet implemented in practice since there are no direct analogue components available. However, it is possible to obtain the equivalent frequency response of a fractional order transfer function using high order integer-order approximations. Various methods for integer-order approximations of FOC have been proposed and successfully implemented in practice (Oustaloup *et al.*, 2000; Melchior *et al.*, 2002; Monje *et al.*, 2008). Nevertheless, the burden of this extra step remains necessary in the case of FOC, in which the proper implementation is not trivial.

2.3 Frequency domain approach

For simplicity in formulation of the FOC, the frequency domain will be used to illustrate the determination of the *utopic controller* frequency response $C^*(j\omega)$. Recalling (6), the equivalent frequency domain formulation can be written as:

$$C^*(j\omega) = \frac{(j\omega)\cdot R(j\omega)}{\left[1-R(j\omega)\right]P(j\omega)} \tag{13}$$

Since we do not want to identify the process transfer function $P(j\omega)$, we introduce the signals $u_f(t)$ and $y_f(t)$. Supposing the input signal $u(t)$ is a sine-sweep with *n* samples, in the form:

$$u(n) = \sin\left[K\left(e^{-n/Lf_s} - 1\right)\right] \tag{14}$$

which is in fact a sinusoid whose frequency is exponentially increased from the lower bound to the higher bound of frequency range (ω_1, ω_2) over T seconds, with *f*s the sampling frequency, $K = \frac{\omega_1 T}{\ln \frac{\omega_2}{\omega_1}}$ and $L = \frac{T}{\ln \frac{\omega_2}{\omega_1}}$. This formulation of the excitation signals allows us excite one frequency at a time, in a single trial and re-formulate (13) in function of the input-output signals:

$$A_{C^*} e^{j\varphi_{C^*}} = e^{j\varphi} \frac{A_y e^{j\varphi_y}}{A_u e^{j\varphi_u}} \frac{A_R e^{j\varphi_R}}{1 - A_R e^{j\varphi_R}} \tag{15}$$

where $A_u e^{j\varphi_u}$ and $A_y e^{j\varphi_y}$ are available through measurements and $A_R e^{j\varphi_R}$ through calculations at each excited frequency (14) of the desired $R(j\omega)$. Finally, we obtain the frequency response of the *utopic controller* $C^*(j\omega)$ in its Bode characteristic representation, namely magnitude and phase, at the desired frequency points of interest. The frequency range in which the controller is evaluated with (15) and then identified in the form given by (7) or (8), depends on the characteristics of the process to be controlled P and the desired closed-loop performance defined by R. Once (15) is available, a linear or nonlinear optimization problem must be solved, identifying the unknown parameters of the controller.

Global optimization is the task of finding the absolutely best set of admissible conditions to achieve an objective under given constraints, assuming that both are formulated in mathematical terms. Some large-scale global optimization problems have been solved by current methods, and a number of software packages are available that reliably solve most global optimization problems in small (and sometimes larger) dimensions. However, finding the global minimum, if one exists, can be a difficult problem (very dependant on the initial conditions). Superficially, global optimization is a stronger version of local optimization, whose great usefulness in practice is undisputed. Instead of searching for a locally feasible point one wants the globally best point in the feasible region. However, in many practical applications finding the globally best point, though desirable, is not essential, since any sufficiently good feasible point is useful and usually an improvement over what is available without optimization (this particular case). Besides, sometimes, depending on the optimization problem, there is no guarantee that the optimization functions will return a global minimum, unless the global minimum is the only minimum and the function to minimize is continuous (Pintér, 1996). Taking all these into account, and considering that the set of functions to minimize in this case is continuous and can only present one minimum in the feasible region, any of the optimization methods available could be effective, a priori. For this reason, and taking into account that Matlab is a very appropriate tool for the analysis and design of control systems, the optimization toolbox of Matlab has been used to reach out the best solution with the minimum error. The **lsqnonlin** nonlinear least-squares function has been used which returns the set of parameters from either (7), either (8), depending on the desired structure of the controller (Mathworks, 2000a).

An elegant solution to avoid this extra step is to fit directly the frequency response of the controller $C^*(j\omega)$ with a properly chosen order transfer function. The fact that $C^*(j\omega)$ is a polynomial, instead of a transfer function, does not present any particular difficulty. One may choose to use the Matlab function **fitfrd** which delivers a state space representation of a fitted transfer function to the given frequency response of the controller (Mathworks, 2000b). For example, in the case of the standard PID form (7), it is necessary to set the specifications to a 2nd order transfer function, with a relative degree equal to 2 (excess poles). This will result in a transfer function of the form $C_{fit}(s)=\dfrac{k}{c_0 s+c_1+c_2 s^2}$, from which the controller transfer function becomes:

$$C(s)=\frac{1}{s}\cdot\frac{1}{C_{fit}(s)} \tag{16}$$

2.4 Adaptation procedure

Once the controller's parameters have been found, these parameters can be adapted if the changes in the process require another tuning values for fulfilling the specified closed loop performance.

The adaptation procedure can be summarized in few steps as following:

- perform an input-output test measurement in the practical frequency range of interest;
- calculate the magnitude-phase frequency response using frequency domain analysis techniques;
- calculate the frequency response of the *utopic controller* with (13)-(15);
- fit the controller structure from (7) with linear least squares or (8) with nonlinear least squares identification procedure;
- apply the controller using (16).

3. Illustrative examples

In this section, two typical examples which are considered of academic interest, will be presented. Both integer and fractional order controllers will be developed based on closed loop specifications given by the reference model.

3.1 A typical position servo system

A typical position servo system contains a first order plant with an integrator, for example:

$$P(s)=\frac{0.25}{s(s+1)} \tag{17}$$

The difficulty in this case arises from the presence of a double integrator in the closed loop, namely the one from the plant and the one from the controller. The unit impulse response of the system from (17) and the corresponding frequency response is given in figure 2, along with the frequency responses of the two reference models, namely:

$$R(s)=\frac{1+4\tau s}{(1+\tau s)^4} \tag{18}$$

with τ=0.05 and 0.01, respectively. The frequency band of interest for tuning the *utopic controller* is $\omega \in \left(10^{-1}, 10^{1}\right)$. Figures 3-4 present the optimisation result in fitting the controller frequency response, and the closed loop unit step response in the two design cases.

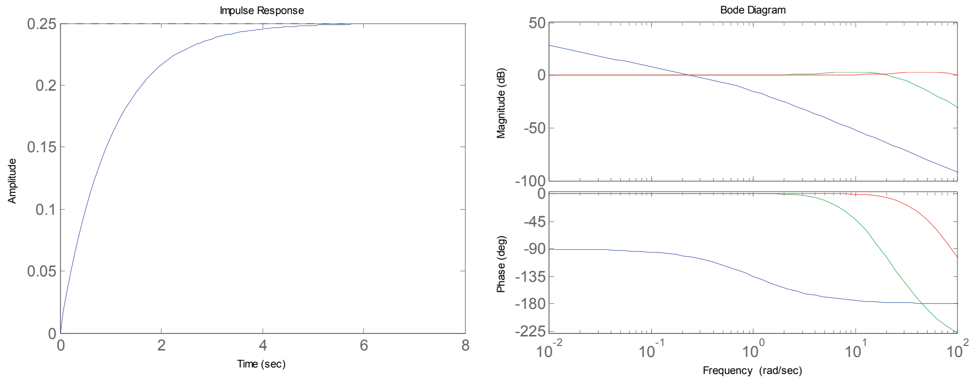

Figure 2. Left: open loop unit impulse response of the process; Right: Bode characteristics of the process (blue) and the reference models for τ=0.05 (green) and for τ=0.01 (red)

The corresponding controller parameters are given in Table 1.

	τ	c_0	c_1	c_2	λ	μ
IO_PID	0.05	266.635	313.144	42.608	1	1
FO_PID	0.05	273.332	298.938	73.356	0.969	0.759
IO_PID	0.01	6666.7	6891.6	220.5	1	1
FO_PID	0.01	6743.9	6678.7	267.2	1	0.9

Table 1. Controller parameters for the position servo system

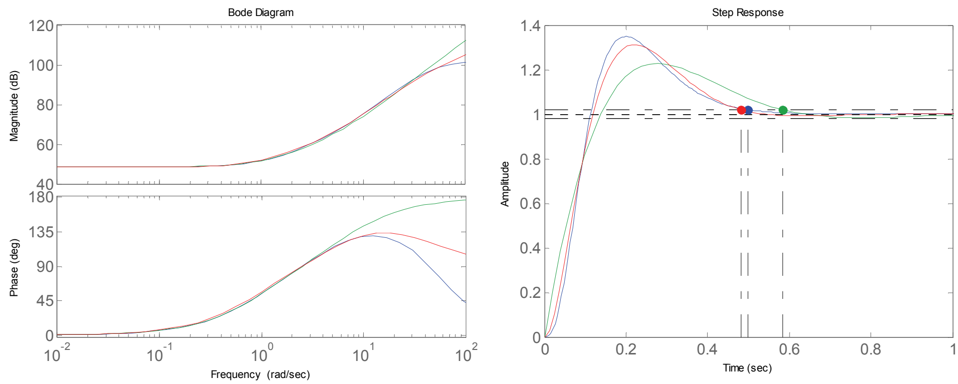

Figure 3. Left: frequency domain approximation and Right: unit step responses for τ=0.05; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

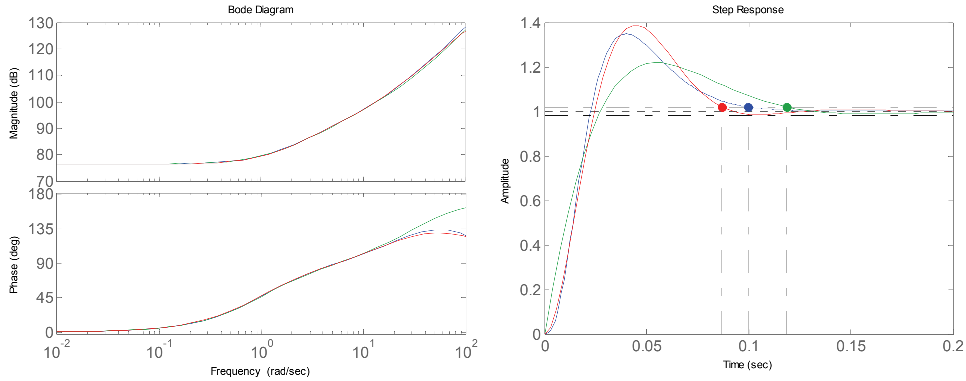

Figure 4. Left: frequency domain approximation and Right: unit step responses for τ=0.01; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

3.2 Highly oscillatory system

The process in this example is highly oscillatory due to the low damping factor ξ:

$$P(s) = K\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}\frac{1}{(1+as)^2} \tag{19}$$

with K=0.3, $\omega_n = 0.04\pi$, $\xi = 0.1$ and a=5. The reference model has been chosen as in:

$$R(s) = \frac{1}{(1+\tau s)^4} \tag{20}$$

with τ=15 and τ=10. The open loop unit step response is depicted in figure 5, clearly showing the oscillatory behaviour of the system, making it difficult to control. The frequency characteristics of the process and the two reference models are given in figure 5, right. From these, the useful frequency range of the controller is taken as $\omega \in \left(10^{-2.8}, 10^{-0.8}\right)$. Figures 6-7 present the optimisation result in fitting the controller frequency response, and the closed loop unit step response in the two design cases, while Table 2 summarizes the corresponding controller parameters.

	τ	c_0	c_1	c_2	λ	μ
IO_PID	15	0.056	0.011	2.809	1	1
FO_PID	15	0	0.051	2.891	1	1
IO_PID	10	0.083	0.071	4.426	1	1
FO_PID	10	0	0.082	3.623	1	0.921

Table 2. Controller parameters for the highly oscillatory system

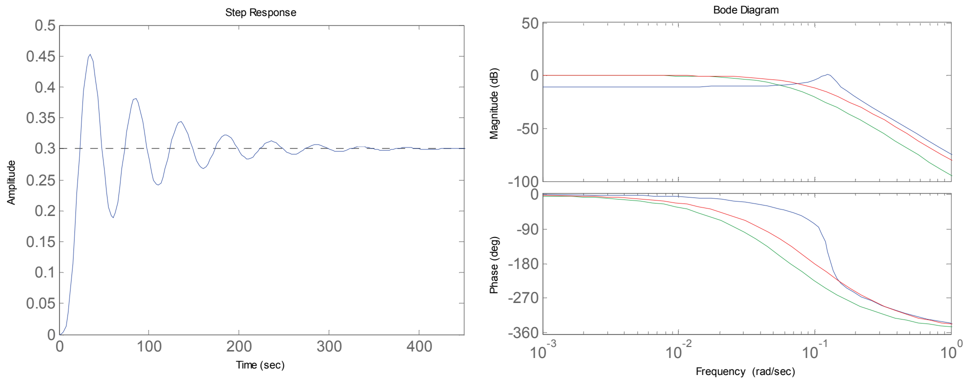

Figure 5. Left: open loop unit step response of the process; Right: Bode characteristics of the process (blue) and the reference models for τ=15 (green) and for τ=10 (red)

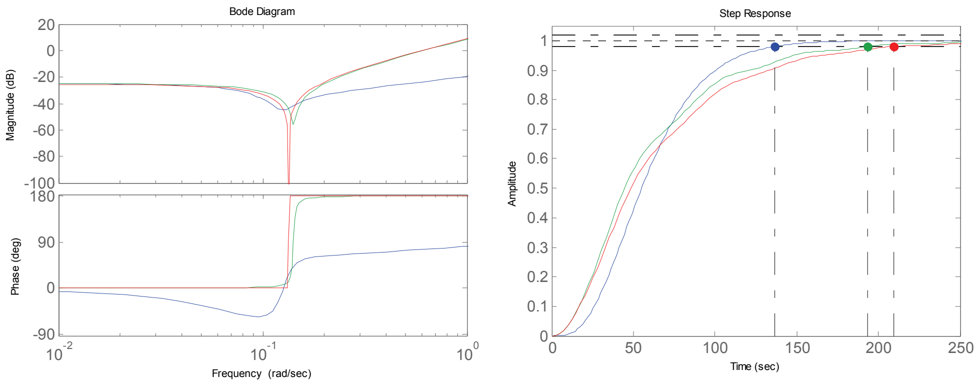

Figure 6. Left: frequency domain approximation and Right: unit step responses for τ=15; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

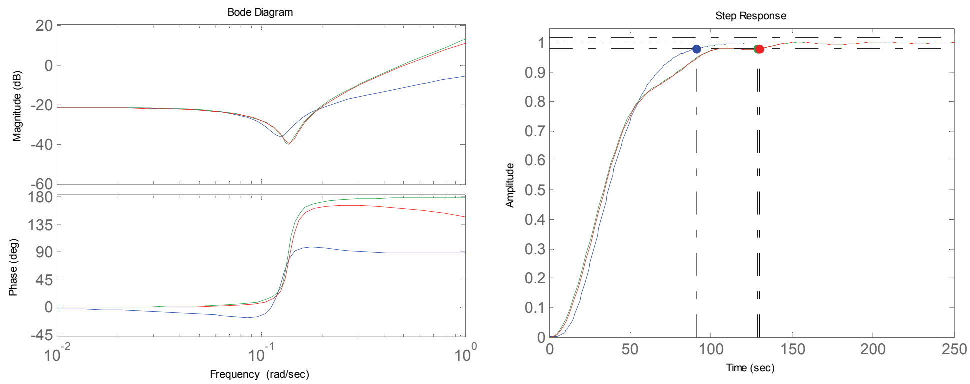

Figure 7. Left: frequency domain approximation and Right: unit step responses for τ=10; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

4. Practical Application: Mechanical Ventilation

One of the novel concepts in control engineering is that of fractals, self-similarity in geometrical structures (Weibel, 2005). Although originally applied in mathematics and chemistry, the signal processing community introduced the concept of fractional order modelling in technical and non-technical areas. A perfect example of fractal structure is that of the lungs. Observations support the claim that dependence exists between the viscoelasitcity and the air-flow properties in the presence of airway mucus with disease and that fractional orders appear intrinsically in viscoelastic materials (i.e. soft lung tissue) (Suki *et al.*, 1994). These mechanical properties are captured in the input impedance, which gives insight upon airway and tissue resistance and compliance.

The respiratory input impedance can be measured non-invasively at the mouth during quiet breathing of the patient, without requiring any special manoeuvres. In this lung function test, forced oscillations are superimposed on the breathing pattern of the patient, in the form of a multisine signal, exciting frequencies in the 4-48Hz range. An I2M (Input Impedance Measurement) device produced by Chess Medical Technologies, The Netherlands (2000) has been used for pulmonary testing. The specifications of the device are those of commercially available i2m devices: 11kg, 50x50x60 cm, 8 sec measurement time, European Directive 93/42 on Medical devices and safety standards EN60601-1. The subject is connected to the typical setup from figure 8 via a mouthpiece, suitably designed to avoid flow leakage at the mouth and dental resistance artifact. The oscillation pressure is generated by a loudspeaker (LS) connected to a chamber (Oostveen *et al.*, 2003). The LS is driven by a power amplifier fed with the oscillating signal generated by a computer (*U*). The movement of the LS cone generates a pressure oscillation inside the chamber, which is applied to the patient's respiratory system by means of a tube connecting the LS chamber and the bacterial filter (bf). A side opening of the main tubing (BT) allows the patient to have fresh air circulation. Ideally, this pipeline will have high impedance at the excitation frequencies to avoid the loss of power from the LS pressure chamber. It is advisory that during the measurements, the patient wears a nose clip and keeps the cheeks firmly supported. Before starting the measurements, the frequency response of the transducers (PT) and of the pneumotachograph (PN) are calibrated. The measurements of air-pressure P and air-flow $Q\ (=\dot{V}$, with V - air volume$)$ during the forced oscillations lung function test is done at the mouth of the patient. Using electrical analogy, whereas the P corresponds to voltage and Q corresponds to current, the respiratory impedance Z_r can be defined as their spectral (frequency domain) ratio relationship:

$$Z_r(j\omega) = \frac{S_{PU}(j\omega)}{S_{QU}(j\omega)} \tag{21}$$

where $S_{ij}(j\omega)$ denotes the cross-correlation spectra between the various input-output signals, ω is the angular frequency and $j = (-1)^{1/2}$, resulting a complex variable. This non-parametric representation can be further identified with parametric models, quantifying some of the mechanical properties of the lung tissue, such as: resistance, compliance and inertance. Depending on the values of these parameters, clinicians can distinguish between healthy and pathologic cases, as well as between various types of lung disease.

Recently, it has been shown that fractional order model characterizing impedance provide better identification results due to their intrinsic nature of capturing variations in frequency domain which are not dependent on the integer multiples of 20dB/dec and π/2 for magnitude and phase, respectively (Ionescu & De Keyser, 2008a). Such a fractional order impedance model can be represented in the form:

$$Z(s) = Ls^{\alpha} + \frac{1}{Cs^{\beta}} \tag{22}$$

with Z the impedance, L the inductance and C the compliance of the total respiratory system and α,β fractional. In this example, the impedance of a patient diagnosed with chronic obstructive pulmonary disease has been used, where L=0.00166 kPa s²/l, C=2.045 l/kPa, α=0.5524 and β=0.5395 (Ionescu *et al.*, 2008b; Ionescu and De Keyser, 2008c). The parameters of the mechanical properties of the lung tissue in these patients can vary during several stages of the treatment applied by clinicians, including medication and ventilatory support. These patients are under mechanical ventilation, to ensure optimal conditions for gas exchange in the body (Behbehani, 2006). The efficiency of the ventilator depends on the optimal matching of the ventilator settings to the mechanical properties of the respiratory system, which may vary significantly in time. The ventilator can be approximated by a 3rd order transfer function of the form:

$$V(s) = \frac{1}{(10s+1)^3} \tag{23}$$

and the total process to be controlled is given by $P(s) = Z(s) \cdot V(s)$.

Due to the fact that (22) is a fractional order model, we evaluate the frequency domain of the process in order to decide upon the frequency band of the controller. Since the reference model can be used to specify the speed of the closed loop, one needs to attain insight on the speed of the process in open loop. For this, integer order approximation is performed using the method described in (Oustaloup *et al.*, 2000) and the step response of the total process $P(s)$ is given in figure 9, left. Based on this information, the reference model has been chosen in the form:

$$R(s) = \frac{1}{(\tau s+1)^4} \tag{24}$$

with τ=10 and 5, respectively. The Bode characteristics of the process and the reference models are given in figure 9, right. Using (13), the controller transfer function is obtained and the problem of nonlinear optimization is solved using **lsqnonlin** for the unknown parameters in (8), in the frequency range $\omega \in \left(10^{-3}, 10^{-1}\right)$. Notice that in practice the process is unknown, so based on the known input and output signals, one may find the frequency response of the controller using (15). After fitting the frequency response with minimum errors (see figures 10-11 left), the resulted set of parameters for the integer-order controller IO_PID from (7) and for the fractional-order controller FO_PID from (8) are those given in Table 3. The corresponding closed loop responses with the respective controllers implemented in the form given by (16) are depicted in figures 10-11, right.

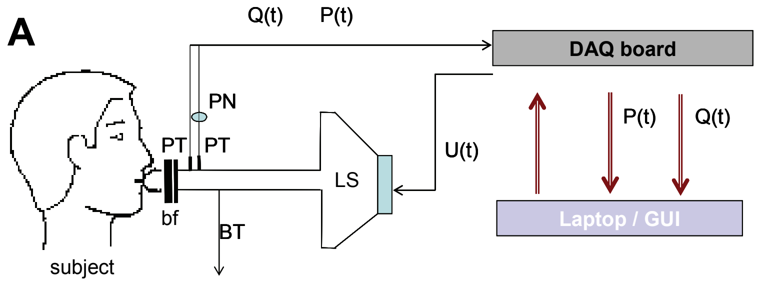

Figure 8. Schematic representation for the forced oscillation lung function testing device; see text for symbol explanation

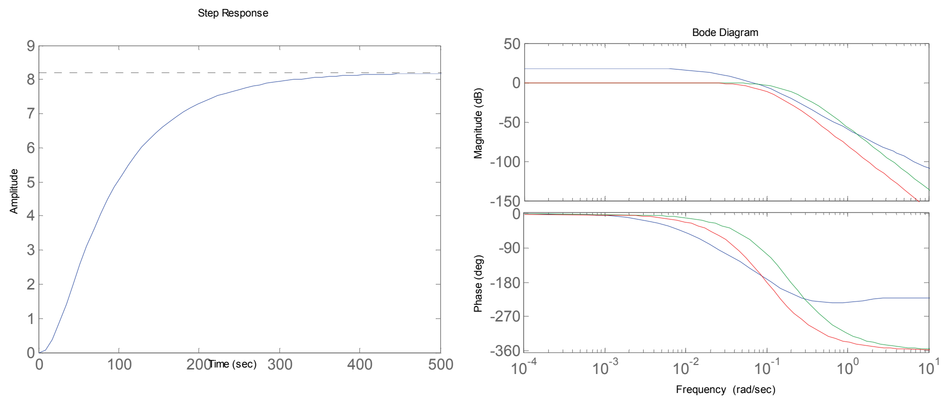

Figure 9. Left: open loop unit step response of the process; Right: Bode characteristics of the process (blue) and the reference models for τ=10 (green) and for τ=5 (red)

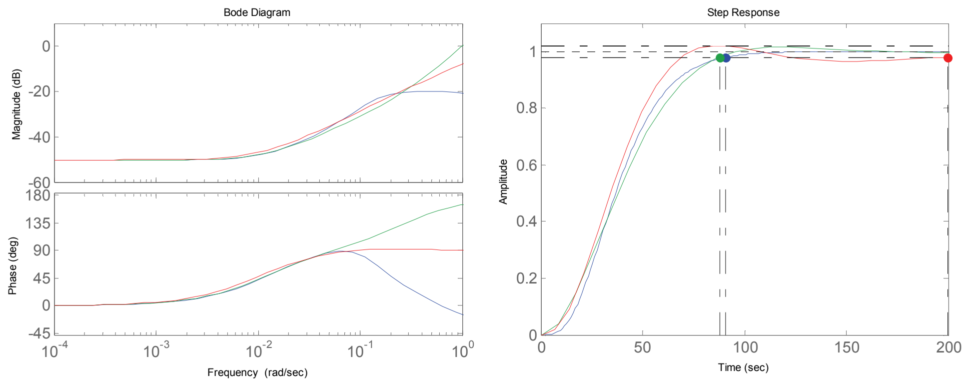

Figure 10. Left: frequency domain approximation and Right: unit step responses for τ=10; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

	τ	c_0	c_1	c_2	λ	μ
IO_PID	10	0.003	0.267	0.971	1	1
FO_PID	10	0.048	0.004	0.386	0.956	0.089
IO_PID	5	0.006	0.576	6.381	1	1
FO_PID	5	0.229	0.021	3.615	0.797	0.623

Table 3. Controller parameters for the mechanically ventilated respiratory system

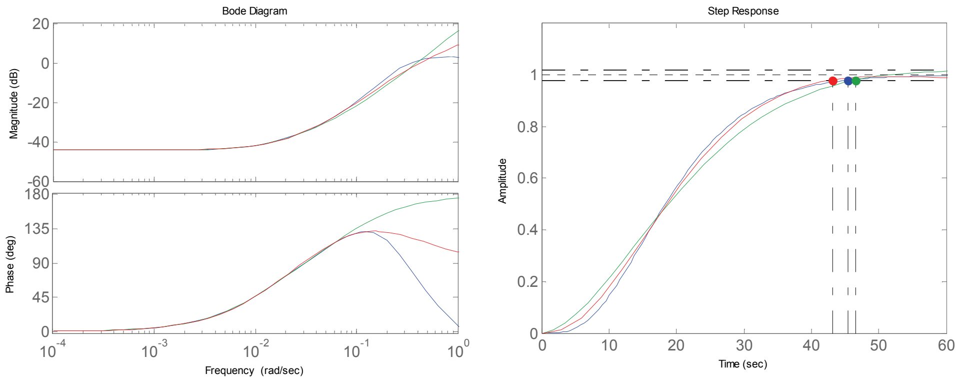

Figure 11. Left: frequency domain approximation and Right: unit step responses for τ=5; reference (blue), IO_PID (green) and FO_PID (red). Circles denote settling times

4. Discussion

4.1 Tuning aspects

Obviously, the first and crucial step in DIRAC is the choice of the reference model $R(s)$. This can be done if some knowledge on the process is available. Some of the rule-of-thumb guidelines can be summarized in the following list:

- if the controller contains integral action, it ensures zero steady state error, which must be reflected in the closed loop gain; the latter should be 1, i.e. $R(1)=1$;
- if both process and controller contain an integrator, it is the case of type 2 control loop (double integrator); this means that the closed loop can track a ramp-setpoint without error and this should be reflected by the choice of the reference model (see, for example, (18));
- if the process contains a dead-time, the closed loop will also be affected by it, therefore in the reference model the presence of the dead-time is necessary and an approximate value suffices to obtain good results;
- if the process is non-minimum phase, the closed loop will also have this property; therefore the reference model should also be non-minimum phase.

In the previous sections it has been stated that the choice of the time constant in the reference model affects the speed of the closed loop. When defining the time constant of the reference model, the actual time constant of the process has to be taken into account. In other words, the desired closed loop speed should be in the same order of magnitude as the open loop settling time of the process. If this condition is not fulfilled, the reference model will ask the closed-loop to behave in an un-realistic way and the process will not be able to follow the actions of the controller, perhaps leading to poor robustness and even instability. Apart from this, the control effort required to fulfil the specifications imposed by the reference model should also be within realistic limits.

As a general observation, the choice of the reference model is not in all situations a 'best' choice. Especially in direct adaptive methods, in which the knowledge of the process is not required, there are uncertainties on the system behaviour. In order to overcome this problem, it is possible to adjust both the reference model as well as the controller parameters. This adaptation must be based on some capability sensing parameters from the process, which would then re-define the reference model to adapt controller parameters to the new achievable specifications. However, the baseline observation is that the reference model is specified such that its output yields a desired, as well as an achievable response.

The second step in the DIRAC algorithm presented here is related to the fact that the reasoning is transposed from time domain (or discrete time domain) to frequency domain. It is clear that a frequency band of interest must be defined, in order to fit the controller's parameters. By definition, it is not possible with a single, linear and simple model to capture the entire frequency response of the desired controller. It is important to choose meaningfully the frequency interval over which the fitting will be done. In this case, one can obtain the actual frequency response of the plant, from the input-output measurements, as from (14)-(15). In this case, the choice of the excitation signal and its frequencies is significant. By looking at the cross-over frequency of the plant and the desired frequency bandwidth of the reference model, one can reason upon the effective frequency interval. Notice that the low frequencies are not important to be perfectly modelled, because the presence of the integrator in the controller ensures steady state error zero (16).

Finally, whether the controller structure is the standard integer order PID from (7) or the more 'flexible' fractional order PID from (8) is a choice of the user. From the presented examples, it appears that there is no guarantee that a fractional-order PID outperforms an integer order PID. Further research will be necessary before a classification can be made upon processes in which FOC is better suitable than standard integer order control.

4.2 Implementation aspects

It is necessary to include here some of the important settings dealing with the implementation of the DIRAC scheme. The fact that in this paper we chose to work in frequency domain is solely due to the fractional order derivatives/integrals which are present. Of course, from a practical standpoint, a discrete time controller is necessary and the discrete-time DIRAC algorithm has been presented in (De Keyser, 1989; De Keyser, 2000).

Firstly, since the representation is in frequency domain, all the necessary transfer functions to calculate the *utopic controller* from (13) have to be dealt with in function of the chosen frequency interval of interest, from which (15) is calculated. Secondly, if the choice of the controller structure is that of an integer order PID, then the Matlab function **fitfrd** can be applied directly to obtain a 2nd order transfer function with relative degree 2 (number of excess poles) and the final controller results as in (16) (MathWorks, 2000b). If the choice of the controller structure is that of a fractional order PID, the nonlinear least squares function **lsqnonlin** is employed, since the function to be minimized (8) is nonlinear in the parameters. Since the choice of the initial values is a critical step in nonlinear optimization, these have been set to the parameters resulted from the integer order PID. This choice is regarded as the best guess upon the final (optimal) values of the parameters to be estimated by the nonlinear estimator. After providing the fitting in the frequency domain with (8), the next step is to convert this polynomial to a stable, integer order transfer function. Again, the use of Matlab functions is not an obvious solution, and care must be taken when choosing the function parameters. To achieve acceptable results, the function **invfreqs** has been employed, delivering the transfer function fitted to the given frequency response (MathWorks, 2000c). The advantage over the **fitfrd** function consists in options parameters, which may be chosen such that the algorithm guarantees stability of the resulting linear system and searches for the best fit using a numerical, iterative scheme. The superior ("output-error") algorithm uses the damped Gauss-Newton method for iterative search (MathWorks, 2000b).

5. Conclusions

A simple and straightforward to understand direct adaptive control algorithm (DIRAC) has been presented in this chapter, from a frequency domain perspective, based on previous work derived for discrete-time DIRAC. Both integer order and fractional order PID controllers have been presented and discussed. Three typical examples have been simulated: i) a fractional order process; ii) a double integrator in the closed loop; and iii) a highly oscillatory process with low damping factor. Although the fractional order controller did not prove to outperform the standard PID controller in the presented examples, the DIRAC method remains available to the control engineering community for further research. It should be noted that the controller structure is not limited to PID; in fact, any transfer function can be fitted to the desired frequency response of the controller, as calculated based on the reference model of the closed loop performance.

Further research may be focused towards the following aspects: i) the relationship between DIRAC and other auto-tuning/adaptive methods; ii) stability and convergence analysis; iii) guidelines on the choice of the reference model; iv) the effect of noise and disturbance on the controller's parameter estimation.

6. References

Åström, K.J. & Wittemark, B. (1995). *Adaptive Control* (2nd ed). Addison-Wesley, ISBN*:* 0-201-55866-1

Åström, K.J & Hägglund, T. (1995). *PID Controllers: Theory, design and tuning*. Instrument Society of America, Research Triangle Park, NC, USA, ISBN 1556175167

Anderson, B.D.O. (2005). Failures of adaptive control theory and their resolution. *Communications in Information and Systems*, 5(1), pp 1-20

Behbehani, K. (2006). Mechanical Ventilation. *Biomedical Engineering Fundamentals,* J. D. Bronzino, B. D. Bronzino (Eds)CRC Press, ISBN 0849321220

Bernstein, D. (2002). Feedback control: an invisible thread in the history of technology. *IEEE Ctrl Syst Mag*, 22(2), pp 53-68

Bueno, S., De Keyser, R., & Favier, G. (1991). Auto-tuning and adaptive tuning of PID controllers. *Journal A*, 32(1), pp 28-34.

Butler, H. (1990). *Model Reference Adaptive Control*. PhD Thesis, Technical University of Delft, Delft, The Netherlands.

De Keyser, R. (1989). DIRAC: A Finite Impulse Response Direct Adaptive Controller. Invited chapter in: S.L. Shah, G. Dumont (Eds), *Adaptive Control Strategies for Industrial Use*, Lecture Notes in Control and Information Sciences, ISBN 354051869X, Springer-Verlag, Berlin, pp 65-88

De Keyser, R. (2000). DIRAC: A Direct Adaptive Controller, *IFAC Conference on Digital Control: Past, Present and Future of PID Control,* pp. 199-204, Terassa, Spain, April 5-7 2000

Gorez, R. (1997). A survey of PID auto-tuning methods. *Journal A*, 38(1), pp. 3-10

Ionescu, C. & De Keyser, R. (2008a). Parametric models for characterizing the respiratory input impedance. *Journal of Medical Engineering & Technology*, Taylor & Francis, 32(4), pp 315-324

Ionescu, C., Nour, B., De Keyser, R., Dugan, V., (2008b). Respiratory pressure level regulation based on a fractional-order model for subjects with chronic obstructive pulmonary disease, *IEEE Proc. 11th Int Conf on Optimization of electrical and electronic equipment (OPTIM08)*, IEEE Cat. Nr 08EX1996C, ISBN 1-4244-1545-4, Vol ID-04, 6pages, May 19-21

Ionescu, C., & De Keyser, R., (2008c). Time domain validation of a fractional order model for human respiratory system, *14th IEEE Mediterranean Electrochemical Conf (MELECON08)*, Ajaccio, Corsica, IEEE Cat Nr CFP08MEL-CDR, ISBN 978-1-4244-1633-2, pp. 89-95, May 3-5

Ilchmann, A. & Ryan, E.P. (2003). On gain adaptation in adaptive control. *IEEE Transactions on Automatic control*, 48(5), pp 895-899

Ljung, L. (1987). System Identification, Theory for the user. Prentice Hall, Upper Saddle River, NJ (*1987*) *ISBN* 0-13-881640-9

MathWorks. (2000a). *Matlab optimisation toolbox*., User's guide. The MathWorks, Inc.

MathWorks. (2000b). *Matlab robust control toolbox*., User's guide. The MathWorks, Inc.

MathWorks. (2000c). *Matlab signal processing toolbox*, User's guide. The Mathworks, Inc.

Melchior, P., Lanusse, P., Cois, O., Dancla, F., & Oustaloup, A. (2002). Crone Toolbox for Matlab. Tutorial Workshop on Fractional Calculus Applications in Automatic Control and Robotics, *41st IEEE Conf on Decision and Control CDC'02*, Las Vegas, Nevada, USA, December 9-13.

Monje, C., Vinagre, B., Feliu, V., & Chen Y. (2008). Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice*, 16, pp. 798-812

Oostveen, E., Macleod, D., Lorino, H., Farré, R., Hantos, Z., Desager, K., Marchal, F, (2003). The forced oscillation technique in clinical practice: methodology, recommendations and future developments, *Eur Respir J*, 22, pp 1026-1041

Oustaloup, A., Levron, F., Mathieu B. & Nanot F. (2000). Frequency-band complex non-integer differentiatior : characterization and synthesis. *IEEE Transactions of circuits and systems – I: Fundamental theory and applications*, 47(1), pp 25-39

Pintér, J. (1996). Global optimization in action. Dordrecht, The Netherlands, Kluwer Academic Publishers, ISBN 0792337573

Podlubny, I. (1999). Fractional Differential Equations--*Mathematics in Sciences and Engineering*, vol. 198, Academic Press, ISBN 0125588402, New York.

Suki, B., Barabasi, A.L., & Lutchen, K. (1994). Lung tissue viscoelasticity: a mathematical framework and its molecular basis. *J Applied Physiology,* 76, pp. 2749-2759

Weibel, E.R. (2005). Mandelbrot's fractals and the geometry of life: a tribute to Benoît Mandelbrot on his 80th birthday, in *Fractals in Biology and Medicine*, vol IV, Eds: Losa G., Merlini D., Nonnenmacher T., Weibel E.R., ISBN 9-783-76437-1722, Berlin: Birkhaüser, pp 3-16