

# Student-tailored final year project on microcontroller-based hardware-in-the-loop speed control of a wind generator

Gorka Larruscain, Gerardo Tapia, Ana Susperregui and Miren Itsaso Martinez

## Abstract

Student-supervisor cooperation was carried out to design the inter-subject final year project (FYP) reported in this paper. According to the student feedback, this approach allowed gathering his main interests —vector control (VC), microcontroller ( $\mu\text{C}$ ) programming and wind power generation— together, therefore reinforcing his motivation towards his FYP and making him feel particularly responsible for its outcome.  $\mu\text{C}$ -based hardware-in-the-loop (HIL) emulation made possible to combine the three afore-cited student interests. In this context, the virtual prototype of a current-controlled wind turbine-driven 2-MW permanent-magnet synchronous generator (PMSG) is presented, along with the pseudo-code corresponding to the PMSG speed control algorithm programmed in an 8-bit  $\mu\text{C}$ . In addition, tuning formulas are derived for the digital integral-proportional (I-P) controllers commanding both the PMSG current and speed. Detailed descriptions are provided in order to guarantee reproducibility. Implementation of the HIL rig is also tackled, supported by illustrative results obtained when running it. The developed HIL rig is considered suitable for laboratory practices of subjects like digital control and  $\mu\text{C}$  programming.

## Keywords

Digital control, hardware-in-the-loop, microcontroller, permanent-magnet synchronous generator, pole placement, wind power

## Introduction

Final Year Projects (FYPs) represent excellent opportunities to push students to naturally cross imaginary borders between subjects taken during their degree studies. This aspect is essential for their academic training as engineers.<sup>1,2</sup> At the same time, it is desirable to keep motivation of students as high as possible<sup>3</sup> all through the development of their FYPs. The latter favours a positive attitude and willingness to effort when facing the technical difficulties that students will definitely encounter during the course of their FYPs.

A rather intuitive approach aimed at achieving those two goals may consist in designing inter-subject FYPs in collaboration with interested students. This alternative may be regarded as halfway between constraining students to choose from a list of FYPs fully defined by the academic staff and encouraging them to specify the FYPs they would like to undertake. This way, an opportunity is provided to those students who, even though do not exactly know which FYPs they would like to tackle, have a rather clear idea about which subjects they would like to bring together within their FYPs.

From more than a decade ago, the second and third co-authors of this paper have regularly been supervising FYPs designed in collaboration with interested students, whose academic performances range, as a rule, from average to top. This paper reports one of the latter FYPs conceived following that approach. In particular, during an interview, a student taking a Degree in Industrial Electronic Engineering and Automatics explained his interests in depth to his FYP supervisor, somehow assuming the role of a technological client specifying his requirements. In short, those interests might be summarised as follows:

- To study a control technique not taken throughout the degree studies
- To program a microcontroller ( $\mu C$ ) making use of its main resources
- Wind power generation

As a common practice, digital control and  $\mu C$ -based digital electronic systems are tackled separately, within different subjects, during the degree studies.<sup>4</sup> Little effort is traditionally devoted to make their links evident through, say, laboratory practices. However, the first two requirements set by the FYP student can be easily combined, since digital control algorithms are naturally realized via  $\mu Cs$ .<sup>5,6</sup> Moreover, implementing a control loop in a  $\mu C$  requires using, at least, its following resources: an A/D converter, a timer, a timer-driven interrupt, and a D/A converter—if integrated in the  $\mu C$ —or parallel ports.<sup>4</sup>

Similarly, two possibilities spontaneously arise when aiming at combining the first and the last requirements: considering an advanced control technique to govern the pitch of the wind turbine blades,<sup>7,8</sup> or dealing with vector control (VC) of the wind turbine-driven electric generator.<sup>9,10</sup> In the latter case, the full-converter permanent-magnet synchronous generator (PMSG) topology is deemed the most accessible to get started with VC theory.<sup>11</sup>

Regarding combination of the second and third requirements, hardware-in-the-loop (HIL) emulation is an attractive solution allowing to test control hardware on virtual

prototypes of plants not available as laboratory physical prototypes. In this sense, virtual prototypes of complex industrial plants represent an interesting option to provide students with practical insight into real-world control problems at relatively low cost.<sup>12;13</sup>

In a second interview, all the considerations above were presented to the FYP student, who expressed his preference and motivation towards implementing a VC-based algorithm in a  $\mu\text{C}$ , and validating it via HIL emulation on a virtual prototype of a wind turbine-driven PMSG.

Implementation of VC algorithms, whose current control loops run typically at sampling frequencies close to 10 kHz, requires the use of powerful and, generally, dedicated 32- or 64-bit  $\mu\text{Cs}$  or digital signal processors (DSPs).<sup>11;14–16</sup> However, given that, during the degree studies, fundamentals on  $\mu\text{Cs}$  and their programming are usually taught on the basis of simpler 8- or 16-bit devices, substantial additional training is required to bridge the considerable gap existing between programming those devices and programming 32- or 64-bit  $\mu\text{Cs}$  or DSPs.<sup>14;17</sup>

Bearing in mind that VC was also a topic to be learned by the FYP student, in order to keep both the difficulty and duration of the FYP reasonable, the student and his supervisor agreed to somehow make use of the same 8-bit 80C552  $\mu\text{C}$  studied in the degree. Considering the limited computational capacity of the latter, the implementation of the VC-based algorithm for the PMSG was split into two parts by incorporating the fast current control loops to the virtual prototype, while programming the significantly slower speed control loop, in C language, in the 80C552  $\mu\text{C}$ .

A great part of this paper is devoted to describe, in full detail, the work carried out throughout the above-mentioned FYP, so that it is reproducible without difficulty by the interested reader, student or lecturer. The paper ends up with the experience and feedback of the FYP student and a conclusion section.

## Virtual prototype of the “wind turbine + PMSG + current control loops” system

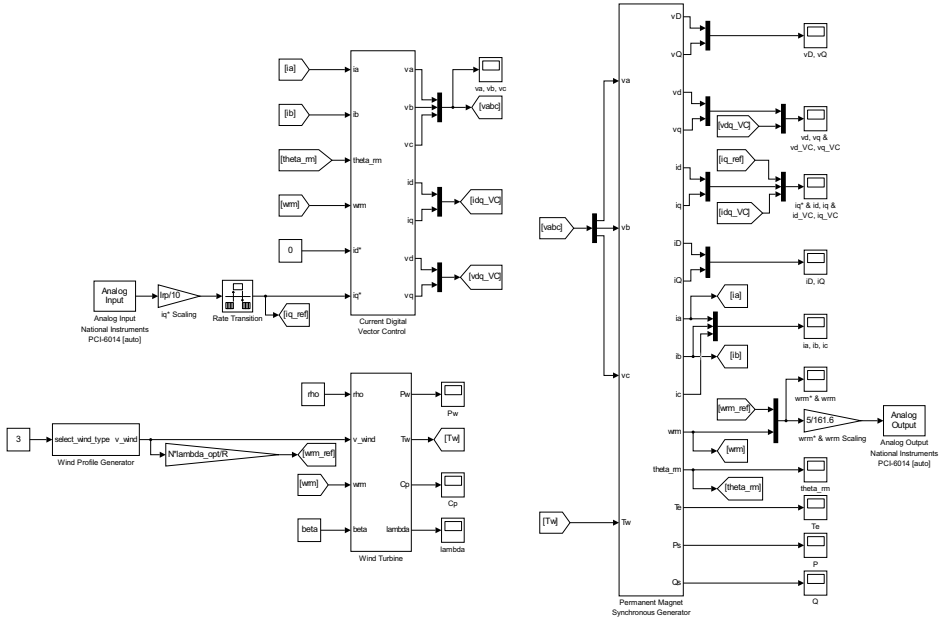
The virtual prototype of the system to be commanded by the 8-bit 80C552  $\mu\text{C}$  was developed using MATLAB/Simulink, and it runs in real time. It is made of the following three subsystems: the wind turbine model, the PMSG model, and the digital current control loops of the PMSG. Each of those subsystems was implemented by means of a C MEX S-function —see Fig. 1—, as described in the following subsections.

### Wind turbine model

The mechanical power captured by the wind turbine,  $P_w$ , is given by

$$P_w = \frac{1}{2} \rho \pi R^2 C_p(\lambda, \beta) v_w^3, \quad (1)$$

where  $\rho$  is the air density,  $R$  represents the blade length,  $C_p$  is the power coefficient, and  $v_w$  corresponds to the wind speed.  $C_p$  is computed by applying the following nonlinear



**Figure 1.** Simulink block diagram of the “wind turbine + PMSG + current loops” virtual prototype.

function of the tip-speed ratio,  $\lambda$ , and the blade pitch angle,  $\beta$ :<sup>9</sup>

$$C_p(\lambda, \beta) = 0.5176 \left( \frac{116}{\lambda_i} - 0.4\beta - 5 \right) e^{-\frac{21}{\lambda_i}} + 0.0068\lambda, \quad (2)$$

with

$$\frac{1}{\lambda_i} = \frac{1}{\lambda + 0.08\beta} - \frac{0.035}{\beta^3 + 1}. \quad (3)$$

Additionally, the tip-speed ratio is calculated as

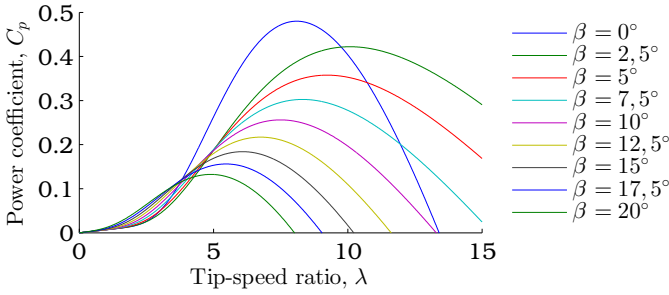
$$\lambda = \frac{\Omega R}{v_w} = \frac{\omega_{rm} R}{N v_w}, \quad (4)$$

where  $\Omega$  and  $\omega_{rm}$  are, respectively, the rotational speeds of the wind turbine and the PMSG, and  $N$  is the ratio of the gearbox connecting the wind turbine and the PMSG shafts.

Based on eqns. (2) and (3), Fig. 2 represents parametric  $C_p$  vs.  $\lambda$  curves for different values of  $\beta$ . It follows that, for the wind turbine under study, an optimum power coefficient  $C_{p_{opt}} = 0.48$  is achieved when  $\lambda = \lambda_{opt} = 8.1$  and  $\beta = 0^\circ$ .

Finally, the wind torque on the PMSG shaft is computed as

$$T_w = \frac{P_w}{\omega_{rm}}. \quad (5)$$



**Figure 2.** Power coefficient vs. tip-speed ratio curves for different blade pitch angles.

As displayed in Fig. 1, the inputs to the wind turbine model are  $\rho$ ,  $v_w$ ,  $\omega_{rm}$  and  $\beta$ , and its output is  $T_w$ . The additional outputs  $P_w$ ,  $C_p$  and  $\lambda$  are provided just for visualisation.

### PMSG model

The electrical model of the PMSG, expressed in the synchronously rotating  $d$ - $q$  reference frame, is given by<sup>10</sup>

$$\frac{di_d}{dt} = \frac{1}{L_d}(v_d - R_s i_d + \omega_r L_q i_q) \quad (6)$$

$$\frac{di_q}{dt} = \frac{1}{L_q}[v_q - R_s i_q - \omega_r(L_d i_d + \Psi_m)], \quad (7)$$

where  $i_d$ ,  $i_q$  and  $v_d$ ,  $v_q$  are, respectively, the direct- and quadrature-axis components of the stator current and voltage,  $R_s$  is the stator resistance,  $L_d$ ,  $L_q$  represent the direct- and quadrature-axis stator inductances,  $\Psi_m$  is the flux of the permanent magnets, and  $\omega_r = P\omega_{rm}$  corresponds to the rotor electrical speed, with  $P$  denoting the number of pole pairs of the PMSG.

$v_d$  and  $v_q$  are derived from the three-phase stator voltages,  $v_a$ ,  $v_b$ ,  $v_c$ , by making use of both Clarke's and Park's transformations as follows:<sup>10</sup>

$$\begin{aligned} \begin{bmatrix} v_d \\ v_q \end{bmatrix} &= \underbrace{\begin{bmatrix} \cos \theta_r & \sin \theta_r \\ -\sin \theta_r & \cos \theta_r \end{bmatrix}}_{e^{-j\theta_r}} \underbrace{\begin{bmatrix} v_D \\ v_Q \end{bmatrix}}_{\mathbf{v}_{DQ}} = \\ &= e^{-j\theta_r} \underbrace{\frac{1}{3} \begin{bmatrix} 2 & -1 & -1 \\ 0 & \sqrt{3} & -\sqrt{3} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix}}_{\mathbf{v}_{abc}} = e^{-j\theta_r} \mathbf{C} \mathbf{v}_{abc}, \end{aligned} \quad (8)$$

where  $\theta_r = P\theta_{rm}$ ,  $\theta_r$  and  $\theta_{rm}$  being, respectively, the rotor electrical and mechanical positions, and  $v_D$ ,  $v_Q$  are the stationary-frame direct- and quadrature-axis components

of the stator voltage. Correspondingly, the three-phase stator currents,  $i_a, i_b, i_c$ , can be obtained from  $i_d$  and  $i_q$  by applying the inverse Park's and Clarke's transformations; i.e.:

$$\begin{aligned} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} &= \frac{1}{2} \underbrace{\begin{bmatrix} 2 & 0 \\ -1 & \sqrt{3} \\ -1 & -\sqrt{3} \end{bmatrix}}_{\mathbf{C}^+} \underbrace{\begin{bmatrix} i_D \\ i_Q \end{bmatrix}}_{\mathbf{i}_{DQ}} = \\ &= \mathbf{C}^+ \underbrace{\begin{bmatrix} \cos \theta_r & -\sin \theta_r \\ \sin \theta_r & \cos \theta_r \end{bmatrix}}_{\mathbf{e}^{j\theta_r}} \underbrace{\begin{bmatrix} i_d \\ i_q \end{bmatrix}}_{\mathbf{i}_{dq}} = \mathbf{C}^+ \mathbf{e}^{j\theta_r} \mathbf{i}_{dq}, \end{aligned} \quad (9)$$

with  $i_D, i_Q$  being the stationary-frame direct- and quadrature-axis components of the stator current.

The electromagnetic torque developed by the PMSG is computed as<sup>10</sup>

$$T_e = \frac{3}{2}P [\Psi_m i_q + (L_d - L_q) i_d i_q], \quad (10)$$

and the simplified one-mass mechanical model given next is adopted:

$$\frac{d\omega_{rm}}{dt} = \frac{1}{J} (T_w + T_e - D\omega_{rm}) \quad (11)$$

$$\frac{d\theta_{rm}}{dt} = \omega_{rm}, \quad (12)$$

where  $J$  and  $D$  are, respectively, the equivalent inertia and viscous friction of the set including wind turbine, low-speed shaft, gearbox, high-speed shaft and PMSG.

Finally, regarding consumed powers as positive, the PMSG active and reactive powers are calculated as follows:<sup>10</sup>

$$P_s = \frac{3}{2}(v_d i_d + v_q i_q); \quad Q_s = \frac{3}{2}(v_q i_d - v_d i_q). \quad (13)$$

As shown by Fig. 1,  $v_a, v_b, v_c$  and  $T_w$  are the inputs to the PMSG model, while  $i_a, i_b, i_c, \omega_{rm}$  and  $\theta_{rm}$  are its outputs. Furthermore, outputs  $v_D, v_Q, v_d, v_q, i_D, i_Q, i_d, i_q, T_e, P_s$  and  $Q_s$  are incorporated with the only purpose of visualisation.

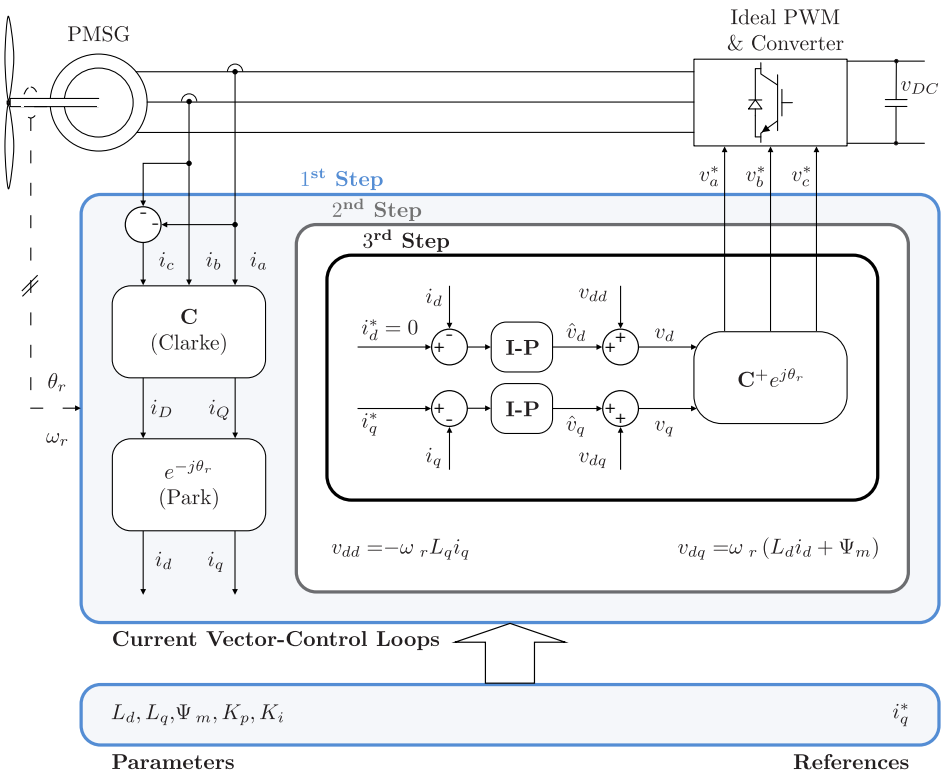
A virtual prototype of a 2-MW PMSG-based wind turbine is adopted for the project, whose parameters are collected in Table 1.

### Digital current control loops of the PMSG

The PMSG current control loops are implemented according to the functional diagram in Fig. 3. It must be considered that any variable present in a given layer of the diagram is also available to the layers inside. Therefore, the control algorithm should be implemented from the outer to the inner layer —labelled, respectively, as ‘1<sup>st</sup> Step’ and ‘3<sup>rd</sup> Step’ at their top.

**Table 1.** Parameters of the 2-MW PMSG-based wind turbine virtual prototype

Parameter	Value
Blade length, $R$	30 m
Gearbox ratio, $N$	45
Equivalent inertia, $J$	562.8955 kg·m <sup>2</sup>
Equivalent viscous friction, $D$	0 N·m/(rad/s)
Stator resistance, $R_s$	10 mΩ
Direct-axis stator inductance, $L_d$	165 μH
Quadrature-axis stator inductance, $L_q$	165 μH
Flux of the permanent magnets, $\Psi_m$	2.5 Wb
Number of pole pairs, $P$	2
Rated r.m.s. line-to-line stator voltage	1000 V
Rated peak value of the stator current	1500 A

**Figure 3.** Functional diagram of the PMSG current control loops.

The '1st Step' is devoted to derive current synchronous components  $i_d$ ,  $i_q$  from three-phase  $i_a$  and  $i_b$  currents by consecutively applying Clarke's and Park's transformations.

Measurement of  $i_c$  is avoided by assuming balanced three-phase current; i.e.,  $i_c = -i_a - i_b$ .

The feedforward decoupling voltage components

$$v_{dd} = -\omega_r L_q i_q; \quad v_{dq} = \omega_r (L_d i_d + \Psi_m), \quad (14)$$

estimated in the ‘2<sup>nd</sup> Step’, allow cancelling the cross-coupling voltage terms present in the PMSG dynamics reflected in eqns. (6) and (7). Knowledge of parameters  $L_d$ ,  $L_q$  and  $\Psi_m$  is required to accurately calculate them.

The  $v_d$  and  $v_q$  voltage components to be applied to the PMSG are derived in the ‘3<sup>rd</sup> Step’ as

$$v_d = \hat{v}_d + v_{dd}; \quad v_q = \hat{v}_q + v_{dq}. \quad (15)$$

It should be noted that substitution of eqns. (15) and (14) into eqns. (6) and (7) leads to the following first-order decoupled dynamics relating  $i_d$ ,  $i_q$  to  $\hat{v}_d$ ,  $\hat{v}_q$ :

$$L_d \frac{di_d}{dt} + R_s i_d = \hat{v}_d; \quad L_q \frac{di_q}{dt} + R_s i_q = \hat{v}_q. \quad (16)$$

Given that  $L_d = L_q = L$  in this case—refer to Table 1—, the two transfer functions arising from eqn. (16) are identical and expressed as

$$\frac{I_d(s)}{\hat{V}_d(s)} = \frac{I_q(s)}{\hat{V}_q(s)} = \frac{b}{s + a}, \quad (17)$$

with  $b = 1/L$  and  $a = R_s/L$ . Accordingly,  $\hat{v}_d$  and  $\hat{v}_q$  are computed via the two identical integral-proportional (I-P) current controllers given next:

$$\hat{V}_d(s) = -K_p I_d(s) + K_i \frac{I_d^*(s) - I_d(s)}{s}; \quad \hat{V}_q(s) = -K_p I_q(s) + K_i \frac{I_q^*(s) - I_q(s)}{s}, \quad (18)$$

with  $K_p$  and  $K_i$  being, respectively, their proportional and integral gains. Another consequence of  $L_d$  being equal to  $L_q$  is that current component  $i_d$  does not contribute to torque production, as evidenced by eqn. (10). Consequently,  $i_d^*$  is set to 0 in order to minimize current consumption.

Replacing eqn. (18) into eqn. (17), the following unity-gain second-order closed-loop transfer function can be derived for  $i_d$  and  $i_q$  current control loops:

$$\frac{I_d(s)}{I_d^*(s)} = \frac{I_q(s)}{I_q^*(s)} = \frac{bK_i}{s^2 + (bK_p + a)s + bK_i}, \quad (19)$$

where  $K_p$  and  $K_i$  are tuned, through pole placement,<sup>18</sup> so that eqn. (19) matches the following target transfer function showing  $\xi$  damping coefficient and  $\omega_n$  natural frequency:

$$\frac{I_d(s)}{I_d^*(s)} = \frac{I_q(s)}{I_q^*(s)} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}. \quad (20)$$



For that purpose, the denominators of transfer functions (19) and (20) are equated coefficient by coefficient, thus giving rise to the tuning equations given next:

$$K_p = \frac{2\xi\omega_n - a}{b} = 2\xi\omega_n L - R_s \quad (21)$$

$$K_i = \frac{\omega_n^2}{b} = \omega_n^2 L. \quad (22)$$

Aiming at avoiding overshoots,  $\xi$  is chosen to be 1. Moreover, once  $\xi$  is fixed,  $\omega_n$  is selected so that a  $t_s = 16.5$ -ms settling time is achieved for current control loops. In particular, according to the 2% criterion,<sup>19</sup>  $\omega_n \cong 5.8/t_s$  when  $\xi = 1$ , which leads to a  $\omega_n$  of 351.5152 rad/s. Replacing those values for  $\xi$  and  $\omega_n$ , together with those for  $R_s$  and  $L = L_d = L_q$  provided in Table 1, into tuning eqns. (21) and (22), it turns out that  $K_p = 0.106$  V/A and  $K_i = 20.3879$  V/(A·s).

Finally, starting from  $v_d, v_q$ , an inverse Park's transformation, followed by an inverse Clarke's transformation, allow deriving the set-points,  $v_a^*, v_b^*, v_c^*$ , of the three-phase voltage to be applied to the PMSG. An ideal power converter is assumed, so that the actual  $v_a, v_b, v_c$  three-phase voltage matches its  $v_a^*, v_b^*, v_c^*$  set-point at any time.

Implementation of a digital version of the current control scheme described above requires the two I-P controllers in (18) to be discretized. Thus, application of Tustin's trapezoidal method<sup>20</sup> to the first I-P in eqn. (18) leads to difference equation

$$\hat{v}_{d_k} = \hat{v}_{d_{k-1}} + K_{pi}(i_{d_k}^* + i_{d_{k-1}}^*) - (K_p + K_{pi})i_{d_k} + (K_p - K_{pi})i_{d_{k-1}}, \quad (23)$$

where index  $k$  stands for  $k$ th sampling instant,  $K_{pi} = K_i h/2$ , and  $h = 100 \mu\text{s}$  is the sample time. Equation (23) is also applicable to the  $q$  axis after replacing subscript  $d$  with  $q$ . Both I-P current controllers are equipped with anti-windup.

Figure 1 evidences that  $i_a, i_b, \theta_{rm}, \omega_{rm}, i_d^*$  and  $i_q^*$  are the inputs to the digital current controller, while  $v_a, v_b$  and  $v_c$  are its outputs. Additional outputs  $i_d, i_q, v_d$  and  $v_q$  are provided just for visualisation.

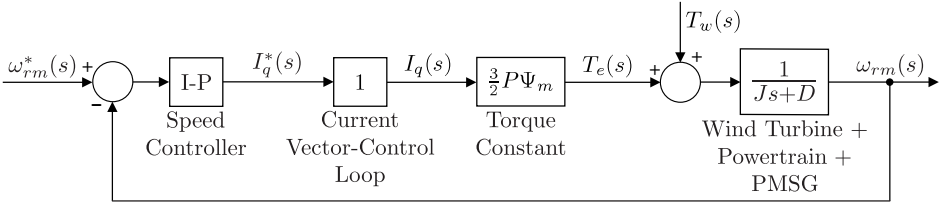
## Digital speed control loop

### Design and tuning

Figure 4 displays a block diagram representing the  $\omega_{rm}$  speed control loop of the PMSG in a simplified manner. An I-P speed controller generates, as control signal, the  $i_q^*$  set-point for the inner  $i_q$  current control loop. The plant to be controlled, placed between  $I_q^*(s)$  input and  $\omega_{rm}(s)$  output, is derived from eqns. (10) and (11) by considering that  $L_d = L_q$ . In addition, given that the dynamics of the speed control loop are much slower—dominant—than those of the  $i_q$  current control loop, the latter is regarded as ideal.

From Fig. 4, the transfer function of the plant to be controlled is derived as

$$\left. \frac{\omega_{rm}(s)}{I_q^*(s)} \right|_{T_w=0} = \frac{b_1}{s + a_1}, \quad (24)$$



**Figure 4.** Simplified block diagram of the PMSG speed control loop.

with  $b_1 = 3P\Psi_m/(2J)$  and  $a_1 = D/J$ . Note that the transfer function in eqn. (24) is analogous to that in eqn. (17), which implies that the method applied to tune the current I-P controllers and, as a result, tuning eqns. (21)-(22) are also valid for the speed I-P controller. Consequently, the proportional and integral gains of the speed I-P controller given by

$$I_q^*(s) = -K_{p1}\omega_{rm}(s) + K_{i1} \frac{\omega_{rm}^*(s) - \omega_{rm}(s)}{s} \quad (25)$$

are adjusted according to tuning equations

$$K_{p1} = \frac{2\xi_1\omega_{n1} - a_1}{b_1} = \frac{2(2\xi_1\omega_{n1}J - D)}{3P\Psi_m} \quad (26)$$

$$K_{i1} = \frac{\omega_{n1}^2}{b_1} = \frac{2\omega_{n1}^2J}{3P\Psi_m}. \quad (27)$$

A  $\xi_1 = 1$  and a  $\omega_{n1} = 4.8333$  rad/s are requested in this case, which lead to a dynamic response showing no overshoots and a  $t_{s1} = 1.2$ -s settling time. Based on those specifications, application of tuning eqns. (26) and (27) yields  $K_{p1} = 725.5098$  A/(rad/s) and  $K_{i1} = 1753.3152$  A/rad.

Finally, the following difference equation, analogous to that in eqn. (23), is programmed to digitally implement the speed I-P controller above:

$$i_{qk}^* = i_{qk-1}^* + K_{pi1}(\omega_{rmk}^* + \omega_{rmk-1}^*) - (K_{p1} + K_{pi1})\omega_{rmk} + (K_{p1} - K_{pi1})\omega_{rmk-1}, \quad (28)$$

where  $K_{pi1} = K_{i1}h_1/2$  and  $h_1 = 60$  ms is the sample time. This controller is also equipped with anti-windup.

### Microcontroller-based implementation

The digital speed controller above was programmed in C language in 80C552  $\mu$ C. The control algorithm itself is implemented within a timer-driven interrupt service routine (ISR), which is scheduled to be executed every  $h_1 = 60$  ms. The main program is just in charge of initializations, staying idle for the rest of the time. The latter is organized as follows:

**1<sup>st</sup> step:** Configure the timer to be used, and load it with the initial value causing its count to roll over—therefore generating an interrupt request—after exactly  $h_1 = 60$  ms.

**2<sup>nd</sup> step:** Enable the timer-driven interrupt and start the timer.

**3<sup>rd</sup> step:** Remain idle using, for example,

```
while ( 1 );
```

Each time a timer-driven interrupt request takes place —thus setting a new sampling instant ( $k$ th sampling instant)—, the following 7 steps are executed within the ISR:

**1<sup>st</sup> step:** If required, reload the timer with the initial value required to generate a new interrupt request  $h_1 = 60$  ms later.

**2<sup>nd</sup> step:** Capture of the current reference and actual speed values,  $\omega_{rm_k}^*$  and  $\omega_{rm_k}$ . Two A/D conversions must be carried out for that purpose. Given that the 80C552  $\mu$ C integrates an 8-channel 10-bit A/D converter (ADC), two integer numbers between 0 and 1023 result from those A/D conversions, which correspond to the [0, 5] V voltage range of the input analog signals.

**3<sup>rd</sup> step:** Based on the  $n_k^*$  and  $n_k$  integer numbers resulting from the A/D conversions,  $\omega_{rm_k}^*$  and  $\omega_{rm_k}$  are derived. In our particular case, the virtual prototype was built so that the [0, 161.6] rad/s speed range is outputted as a voltage signal in the [0, 5] V range —see the detail in Fig. 5a. Consequently,  $\omega_{rm_k}^*$  and  $\omega_{rm_k}$  are derived as follows:

$$\begin{aligned}\omega_{rm_k}^* &= 161.6 * n_k^*/1023; \\ \omega_{rm_k} &= 161.6 * n_k/1023;\end{aligned}$$

**4<sup>th</sup> step:** Computation (update) of the  $i_{q_k}^*$  control signal by applying eqn. (28).

**5<sup>th</sup> step:** If necessary, limitation of  $i_{q_k}^*$  to avoid entering motoring operation — $i_{q_k}^* > 0$ — or exceeding the peak value of the PMSG rated current,  $I_{rp}$ . Given that  $i_d$  is regulated around zero, the peak value of the PMSG current is almost equal to  $|i_q|$  at any instant.

```
if (  $i_{q_k}^* < -I_{rp}$  )
     $i_{q_k}^* = -I_{rp}$  ;
else if (  $i_{q_k}^* > 0$  )
     $i_{q_k}^* = 0$  ;
```

Due to the direct non-canonical implementation<sup>4</sup> provided in eqn. (28) for the I-P algorithm, the limitation above results in simultaneous application of anti-windup.

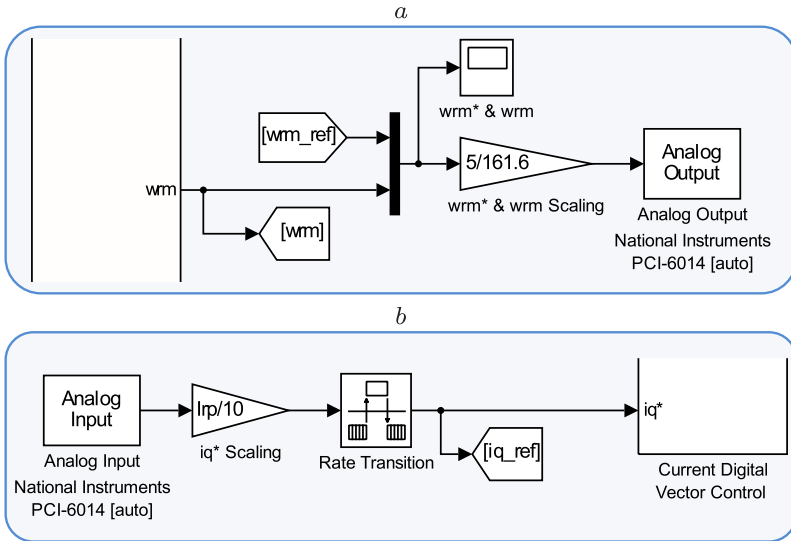
**6<sup>th</sup> step:** Application of the  $i_{q_k}^*$  control signal to the virtual prototype. A D/A conversion is required for that purpose. Since the 80C552  $\mu$ C does not contain any D/A converter (DAC), a 12-bit external DAC providing output voltage in the [−10, 10] V range is used. Accordingly, the integer number to be fed into the DAC is derived as follows:

$$m^* = 4095 * (i_{qk}^* + I_{rp}) / (2 * I_{rp});$$

The output voltage of the DAC is fed into the virtual prototype, which was built so that the  $[-10, 10]$  V range of the DAC output voltage is scaled up to the  $[-I_{rp}, I_{rp}]$  A range of  $i_q^*$ —see the detail in Fig. 5b.

**7<sup>th</sup> step:** Update the discrete states of the I-P controller for the next sampling instant as

$$\begin{aligned} i_{q_{k-1}}^* &= i_{q_k}^* ; \\ \omega_{rm_{k-1}}^* &= \omega_{rm_k}^* ; \\ \omega_{rm_{k-1}} &= \omega_{rm_k} ; \end{aligned}$$



**Figure 5.** Amplitude scaling of reference, feedback and control signals within the virtual prototype. *a* Amplitude scaling of  $\omega_{rm}^*$  and  $\omega_{rm}$ . *b* Amplitude scaling of  $i_q^*$ .

## Hardware-in-the-loop experimentation

The hardware required to perform HIL experimentation, physically observable in Fig. 6, was connected as detailed by Fig. 7.

The virtual prototype consists in the Simulink model of Fig. 1 running in real time in a personal computer connected to NI PCI-6014 data acquisition (DAQ) card. The latter allows outputting  $\omega_{rm}^*$  and  $\omega_{rm}$  from the virtual prototype as analog signals within the  $[0, 5]$  V range. Those signals enter the 80C552  $\mu C$ , included in the DISEN552 module, through channels 0 and 1 of its built-in ADC—pins P5.0 and P5.1.

On the other hand, parallel port P4 and pins P3.2–P3.5 of parallel port 3 bring integer number  $m^*$  —see the 6<sup>th</sup> step of the ISR in the preceding section— from the 80C552  $\mu\text{C}$  to the input of a 12-bit DAC integrated in an interface card. Accordingly, such DAC turns the  $i_q^*$  numerical value computed by the  $\mu\text{C}$  into an analog signal within the  $[-10, 10]$  V range, which enters the virtual prototype via NI PCI-6014 DAQ card.

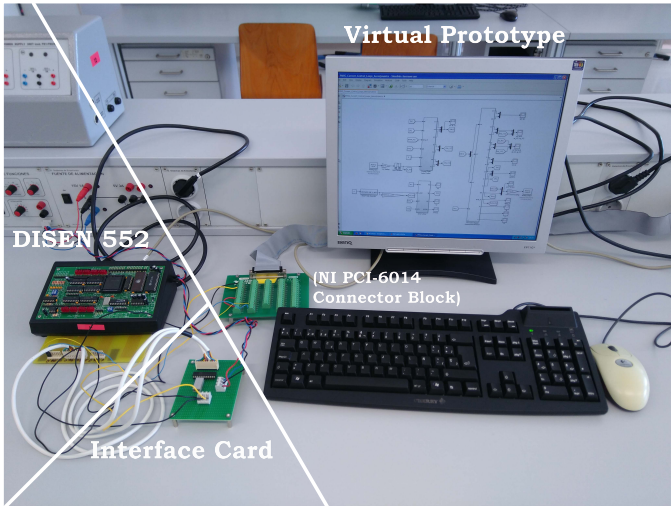


Figure 6. Snapshot of the HIL rig.

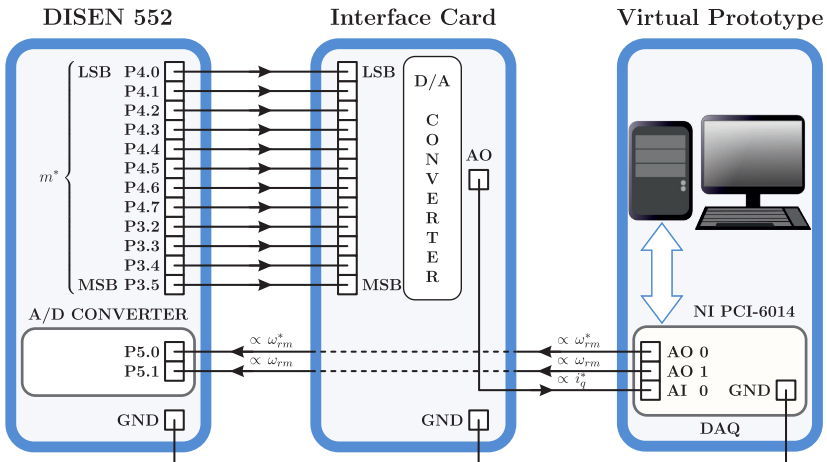


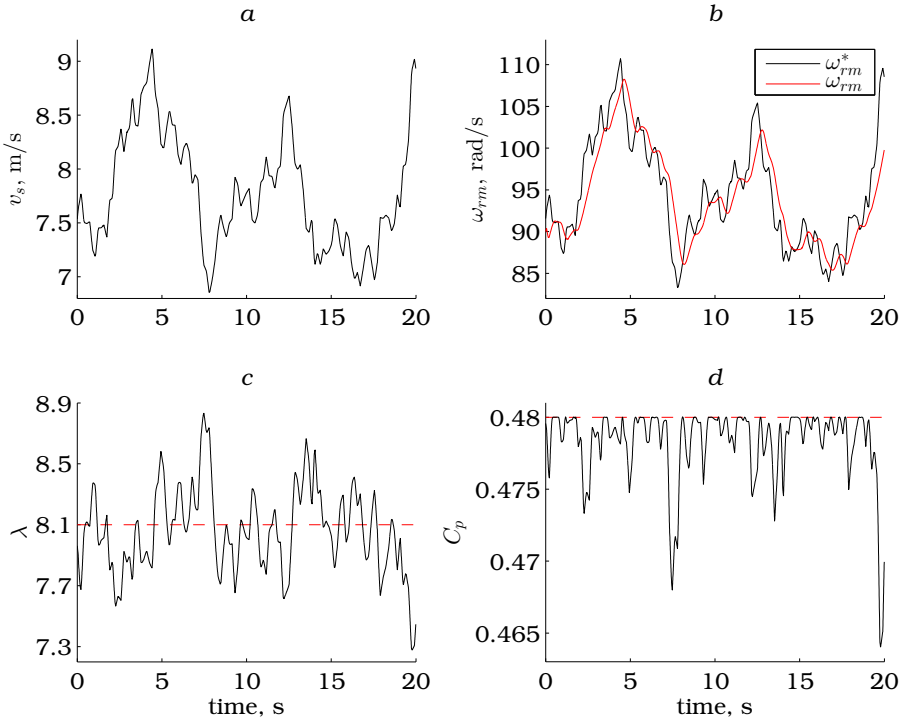
Figure 7. Detailed scheme of the HIL rig connections.

With the purpose of illustrating operation of the HIL rig, the main results of an experiment carried out in the below-rated speed region, where  $\beta = 0^\circ$ , are presented next.

A realistic wind speed profile, displayed in Fig. 8a, was applied. Furthermore, based on eqn. (4) and aiming at reaching the optimum power coefficient, the  $\omega_{rm}^*$  shown in Fig. 8b was computed as

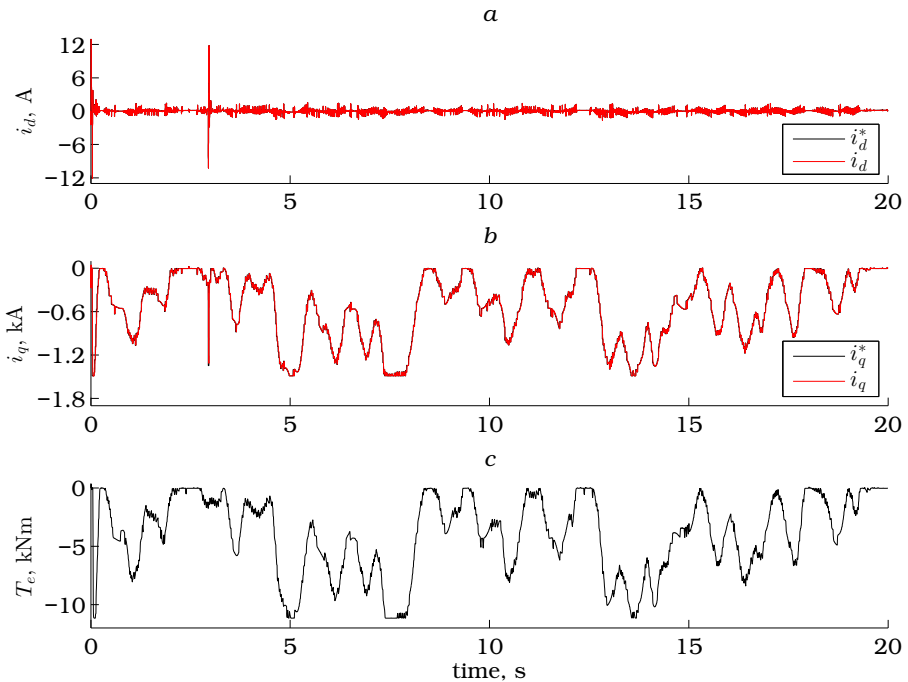
$$\omega_{rm}^* = \frac{N\lambda_{opt}v_w}{R}. \quad (29)$$

Considering both the wind speed variability and the high wind generator inertia, tracking of  $\omega_{rm}^*$  by  $\omega_{rm}$  is more than acceptable, as observable in Fig. 8b. Consequently,  $\lambda$  and, in turn,  $C_p$  remain around their respective optimum values of 8.1 and 0.48, as evidenced by Figs. 8c and d.



**Figure 8.** Performance of the PMSG speed control loop. a Wind speed. b PMSG reference and actual rotational speeds. c Tip-speed ratio. d Power coefficient.

Regarding current VC loops, Figs. 9a and b substantiate, respectively, the satisfactory regulation of  $i_d$  around 0 and the high performance dynamic response of  $i_q$ , resulting in the generating torque of Fig. 9c. The latter leads to the  $\omega_{rm}$  shown in Fig. 8b.



**Figure 9.** Performance of the PMSG current VC loops. *a* Direct-axis current component set-point and actual values. *b* Quadrature-axis current component reference and actual values. *c* Electromagnetic torque.

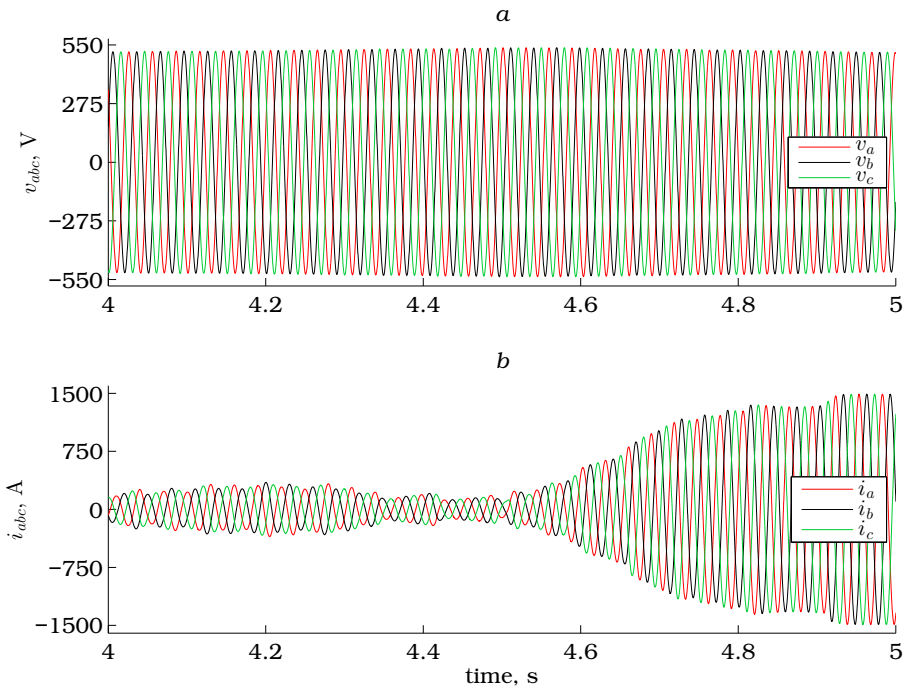
Finally, Figs. 10*a* and *b*, display, respectively, details of the three-phase voltage applied to the PMSG stator and the resulting three-phase current.

## Student experience and feedback

Once the FYP was concluded, the student was kindly asked to freely express his own conclusions in a brief document of no more than 300 words. He handed his feedback over in a sealed envelope, which was opened well after having defended and marked the FYP. Its content is reproduced next.

“I have found highly motivating to actively participate in the design of my own FYP. While allowing me to align it with my main interests, it also pushed me to feel particularly responsible for its final outcome.

The project itself was highly useful in establishing and strengthening some of the fundamental knowledge acquired during my degree studies in Industrial Electronic Engineering and Automatics. That knowledge sets off in control engineering and ends with digital system programming, including principles about electronics and electric



**Figure 10.** Time responses of the three-phase stator voltage and current. *a* Voltage detail. *b* Current detail.

machinery. Consequently, this project, during which I went more deeply into the above-mentioned areas of engineering and became aware about their close connection, could definitely be considered as inter-subject.

Furthermore, I was able to carry out basic research in VC of the PMSG, which was one of the main attractions of the project due to the widespread use of such control technique in industry.

It is also worth mentioning that the software tools adopted proved particularly practical in the development of the project. On the one hand, MATLAB/Simulink allowed me to build a virtual prototype of a current-controlled wind turbine-driven PMSG, capable of running in real time. This made possible bringing together my interests in wind power and  $\mu\text{C}$  implementation of digital controllers with no need for a physical laboratory prototype. Moreover, I both applied and became familiar with the concept of HIL emulation.

On the other hand, the debugger included in the  $\mu\text{Vision}$  software development kit used to implement the digital speed controller eased detection and subsequent correction of all programming bugs within the algorithm. Without this tool, it would have been



a demanding task just correcting all possible bugs, some of which were due to code optimization.

In summary, I feel satisfied both with what I learned during my FYP and with its final outcome.”

So far, the approach for FYP definition put forward in this paper has been evaluated based just on 300-word student commentaries like that reproduced above. Nonetheless, the experience gained up to date suggests that adoption of a more structured methodology would be highly advisable so as to guarantee collection of valuable information on the level of success of the proposed approach. For that purpose, in future FYPs designed via student-supervisor collaboration, it is intended to supplement the aforementioned 300-word commentary with the results of the questionnaire provided in Table 2.

## Conclusions

An inter-subject project, designed by a FYP supervisor in cooperation with an interested student, has been reported. According to the student feedback, this approach allowed not only keeping him motivated all through the development of his FYP, as intended, but also making him “feel particularly responsible for its final outcome.”

The second and third co-authors of this paper have ordinarily been exploring that approach to FYP definition for around a decade. Thus far, it has been confirmed that, in most of the cases, the aim of keeping student motivation while carrying out the FYP was achieved when following the proposed approach. Nevertheless, its main drawback lies possibly in the fact that a higher effort must generally be devoted to the definition of the FYP, when compared to those entirely defined by either the academic staff or the students. In contrast, as far as the development of the FYP is concerned, it is deemed that supervisor workload is not directly related to the way in which that FYP is defined. Other aspects, like its inherent difficulty, the ability of the student to work independently and her/his initiative level, have a prevailing impact on that workload.

The project itself, conceived with the aim of simultaneously covering three topics of interest for the FYP student, has been described in detail, so that it is readily reproducible by appealed readers. In particular, development of a relatively low-cost virtual prototype of current-controlled wind turbine-driven PMSG made possible to experimentally assess, via HIL emulation, a  $\mu$ C-based implementation of its speed controller. The HIL rig implemented is deemed well-suited for laboratory practices of subjects related to digital control and  $\mu$ C programming.

In order to elude bringing further difficulty to the FYP, the student and his supervisor agreed to adopt the same 8-bit  $\mu$ C studied during the degree, which was clearly insufficient to realize the current VC loops of the PMSG, running at 10 kHz. To overcome this drawback, the current VC loops were somehow artificially incorporated to the virtual prototype. However, it is planned to replace the 8-bit  $\mu$ C with a powerful 32-bit one in a future FYP, which will allow transferring the current VC loops from the virtual prototype to the  $\mu$ C. Accordingly, the virtual prototype will emulate just the wind turbine-driven PMSG, while the  $\mu$ C will host the complete control algorithm comprising the outer speed controller cascaded to inner current VC loops.

**Table 2.** Tentative survey to evaluate the success of the proposed approach for FYP definition

Questions	1	2	3	4	5
1. Please, rate the level of difficulty of your FYP. (1: Very low; 3: Average; 5: Very high)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. How did your initial motivation towards your FYP evolve during its development? (1: Substantially decreased; 3: Remained the same; 5: Substantially increased)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. How did your interest in the subjects brought together within your FYP evolve during its development? (1: Substantially decreased; 3: Remained the same; 5: Substantially increased)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. The experience of defining my FYP in collaboration with my supervisor was highly enriching.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. The FYP definition process did not take significantly longer than expected.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. The defined FYP integrated my interests in a natural way.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. My FYP allowed me to identify and strengthen the relationships between subjects taken during my degree studies.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. Participating in the definition of my own FYP pushed me to feel especially responsible for its final outcome.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. How much did you learn throughout your FYP? (1: Much less than expected; 3: As expected; 5: Much more than expected)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. The available laboratory resources were appropriate and sufficient to carry out the defined FYP.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. My supervisor guided me satisfactorily during development of my FYP.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Please, rate your initiative and level of self-sufficiency throughout your FYP. (1: Very low; 3: Average; 5: Very high)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. Overall, I feel satisfied with the developed FYP.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. I deem that the FYP carried out could give me a competitive advantage when seeking employment.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

1: Strongly disagree; 2: Disagree; 3: Unsure; 4: Agree; 5: Strongly agree

## Acknowledgements

Based on the interests of the FYP student, the subject of the reported FYP was envisaged in the context of the DPI2015-64985-R research project, funded by the Spanish Ministry of Economy, Industry and Competitiveness. The student completed his FYP within the “Intelligent Systems and Energy (SI+E)” research group, funded by the Basque Government under the IT677-13 research grant, as well as by the University of the Basque Country UPV/EHU, under both the UFI11/28 unit of formation and research and the GIU16/54 research grant.

## References

1. Baltic M, Krneta R and Rakić A. Interdisciplinary project bridges the gap in electrical engineers’ knowledge of modern control applications. *Int J Elect Eng Educ* 2014; 51: 204–217.
2. Troni G and Abusleme A. Introduction to microbots: A hands-on, contest-driven, interdisciplinary course on mobile robot design in a developing country. *Int J Electr Eng Educ* 2013; 50(4): 395–407.
3. Wong TH. Design of a magnetic levitation control system—An undergraduate project. *IEEE Trans Educ* 1986; E-29: 196–200.
4. Ibrahim D. *Microcontroller based applied digital control*. Chichester: Wiley, 2006.
5. Ibrahim D. Teaching digital control using a low-cost microcontroller-based temperature control kit. *Int J Elect Eng Educ* 2003; 40: 175–187.
6. Gunasekaran M and Potluri R. Low-cost undergraduate control systems experiments using microcontroller-based control of a dc motor. *IEEE Trans Educ* 2012; 55: 508–516.
7. Fragoso S, Ruz ML, Garrido J et al. Educational software tool for decoupling control in wind turbines applied to a lab-scale wind turbine. *Comput Appl Eng Educ* 2016; 24: 400–411.
8. Geng H and Yang G. Robust pitch controller for output power levelling of variable-speed variable-pitch wind turbine generator systems. *IET Renew Power Gener* 2009; 3: 168–179.
9. Zaragoza J, Pou J, Arias A et al. Study and experimental verification of control tuning strategies in a variable speed wind energy conversion system. *Renew Energy* 2011; 36: 1421–1430.
10. Vas P. *Sensorless vector and direct torque control*. New York: Oxford University Press, 1998.
11. Montesinos-Miracle D, Galceran-Arellano S, Sudrià-Andreu A et al. A new low-cost DSP educational tool for a laboratory for motor control. *Int J Elect Eng Educ* 2009; 46: 183–197.
12. Goodwin GC, Medioli AM, Sher W et al. Emulation-based virtual laboratories: A low-cost alternative to physical experiments in control engineering education. *IEEE Trans Educ* 2011; 54: 48–55.
13. Short M and Cox C. RTE-SIM: A simple, low-cost and flexible environment to support the teaching of real-time and embedded control. *Int J Elect Eng Educ* 2011; 48: 339–358.
14. Toliyat HA and Campbell SG. *DSP-based electromechanical motion control*. Boca Raton: CRC Press, 2003.
15. Rodríguez-Reséndiz J, Herrera-Ruiz G and Rivas-Araiza EA. Adjustable speed drive project for teaching a servo systems course laboratory. *IEEE Trans Educ* 2011; 55: 657–666.
16. Rodríguez-Ponce R, Gómez-Lorenzo RA and Rodríguez-Reséndiz J. A project-oriented approach for power electronics and motor drive courses. *Int J Elect Eng Educ* 2015; 52:

219–236.

17. Welch TB, Wright CHG and Morrow MG. *Real-time digital signal processing from MATLAB® to C with the TMS320C6x DSPs*. 2nd ed. Boca Raton: CRC Press, 2012.
18. Tapia G, Tapia A and Ostolaza JX. Two alternative modeling approaches for the evaluation of wind farm active and reactive power performances. *IEEE Trans Energy Convers* 2006; 21: 909–920.
19. Ogata K. *Modern control engineering*. Englewood Cliffs: Prentice Hall, 2001.
20. Kuo BC. *Digital control systems*. New York: Oxford University Press, 1992.