Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Real-time Control and Hardware-in-the-loop Simulation for Educational Purposes of Wind Energy Systems

Adrian Gambier*

**Fraunhofer IWES, Fraunhofer Institute for Wind Energy Systems, 27572 Bremerhaven, Germany (Tel: +49 471 14290-375; e-mail: adrian.gambier@iwes.fraunhofer.de)*

Work financed by the Federal Ministry of Economic Affairs and Energy (BMWi)

Abstract: The education of students in the area of wind energy engineering for industrial as well as research activities is every day more and more challenging. Wind energy converters become continuously larger and more complex and therefore advanced expertise is necessary. On the other hand, large machines are not available neither for research nor for teaching. In particular, the field of control of wind turbines is principally affected because control systems have to be proved in real-time and new control engineers are not able to be familiar with such systems and then it is not possible to gain practical experience. To this end, Hardware-in-the-Loop (HiL) simulation and control can play an important role in this sense. In the present contribution, an architecture of a HiL system for real-time control and simulation of wind turbines is presented. The use of the system helping students in the development of their Master thesis is illustrated by some examples.

Keywords: Control education, control of large wind turbines, real-time simulation, Hardware-in-the-Loop

1. INTRODUCTION

At the Leibniz University Hannover (see Gambier and Balzani, (2019)), a Master degree programme in Wind Energy Engineering is running. It combines all aspects of the wind energy systems in a varied course portfolio, including a course about the principles on control of wind turbines. Some students also finish the programme completing a Master thesis in control of wind energy converters using advanced control approaches.

A significant objective of the programme is to prepare human resources for start working in the very competitive wind energy industry. The concrete problem for the control engineering is the fact that it is impossible to reckon with a large wind turbine in order to run real control experiments. A valid alternative is however the use of Hardware-in-the-Loop (HiL) simulators. Hence, students working in the Master thesis in the area of control can test their developments in the more realistic environment given by a HiL facility.

Hardware-in-the-Loop simulation is a concept that allows to test control algorithms in an environment where not only purely mathematical models of components are used but also a real physical control hardware is embedded in the control loop. Hence, the new architecture provides a more realistic testing infrastructure in particular where the control algorithms can be studied under real-time conditions.

HiL simulations is a tool used in research for a long time. A typical discipline is the aerospace engineering, (see e.g. Evans and Schilling, (1984), Bailey and Doerr, (1996)). HiL simulators are also known in the field of power systems. For example, grid phenomena are studied in Roscoe et al., (2010). Wind energy systems are analysed in Steurer et al., (2004). HiL simulators for the study of nacelle test benches have been reported in Neshati et al., (2016) as well as in Leisten et al., (2017).

The effectivity of HiL concepts in control education has been reported in several works (see e.g. Grega, (1999), Temeltas et al., (2005), Sala and Bondia, (2006), Tejado et al., (2016), Sobota et al., (2019)). In the present work, a HiL architecture for supporting advanced master students in the experimentation of control algorithms for large wind turbines is presented.

The attention is set here in the advanced control algorithms and their real-time implementation of in a dedicated hardware, which is often used in commercial wind energy converters. In addition, a high-resolution model implemented in an aeroelastic software, which is also running in real-time, in order to emulate the wind turbine is considered. The paper is organized as follows: In Section 2, the concept of Hardware-in-the-Loop is introduced according to the interest of this work. Section 3 is devoted to real-time systems and real-time simulation with the view in the HiL application. In Section 4, the HiL implementation is explained. Study examples is the subject of Session 5. Some numerical results are presented in Section 6. Finally, conclusions are drawn in Section 7.

2. HARDWARE-IN-THE-LOOP CONCEPT

Depending on the discipline, several interpretations about the idea of Hardware-in-the-Loop are available in the literature. Detailed descriptions can be found e.g. in Sarhadi and Yousefpour, (2015), Bacic, (2005), Bélanger et al., (2010). Starting from a real computer-controlled system as shown in Fig. 1, where wide arrows are used to represent digital data with parallel representation of bits and thin arrows symbolize analog variables, the idea of HiL is derived according to the sense of the present work.

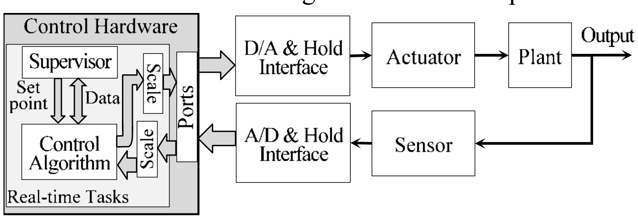

Fig. 1. Scheme of a real control loop

By replacing the chain actuator-plant-sensor of Fig. 1 by the corresponding dynamic models, the Hardware-in-the-Loop scheme of Fig. 2 is obtained.

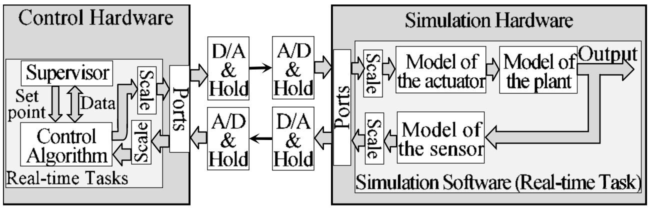

Fig. 2. Hardware-in-the-Loop control system configuration

Because the control system and the simulation system run on different machines, both must be synchronized. Therefore, the simulation must run inside a real time task and the algorithm for the numerical integration has to satisfy time constraints, i.e. numerical computing must meet deadlines.

3. REAL-TIME SYSTEMS AND REAL-TIME SIMULATION

The dynamics of wind energy systems is modelled by a set of algebraic-differential equations, which can be linear, nonlinear and also with partial derivatives. Simulation means then the numerical solution of this set of equations by using a computer program. This program requires a “solver”, i.e., a software implementation of a numerical algorithm to solve differential equations as well as algebraic loops.

On the other hand, a real-time system is a system in which the correctness of a result not only depends on the logical exactness of the calculation but also upon the satisfaction of a previous defined time at which the result has to be made available (see Gambier, (2004)). Hence, real-time tasks must be able to respond deterministically to requests of scheduled tasks as well as to internal and external nondeterministic events. The deterministic system response is thus linked to time constraints associated to tasks in the form of deadlines that must be strictly met.

The concept of real-time changes if it is applied to simulation. In Isermann et al., (1999), real-time simulation is defined as a simulation where input and output signals have the same time dependence as the real running system. Thus, the simulation has to be executed inside a real-time environment in order to satisfy the time dependence, i.e. the dynamic of the real system determines the maximum integration time-step for the solver and the solver is executed in a task with a deadline set at the end of the given integration step. In this manner, the synchronization between real time and simulation time is obtained.

In order to guarantee determinism, the integration step has to be fix in order to avoid iterations and recalculations that normally take place in algorithms with adaptive integration steps making unpredictable the whole computational time.

4. HARDWARE-IN-THE-LOOP ARCHITECTURE

The general architecture of the HiL system implemented here consists of a simulation workstation with interfaces and real time capacity, where an aeroelastic code for the simulation of the wind turbine dynamic behaviour runs, and a distributed hardware for the implementation of the control system. The scheme is illustrated in Fig. 3.

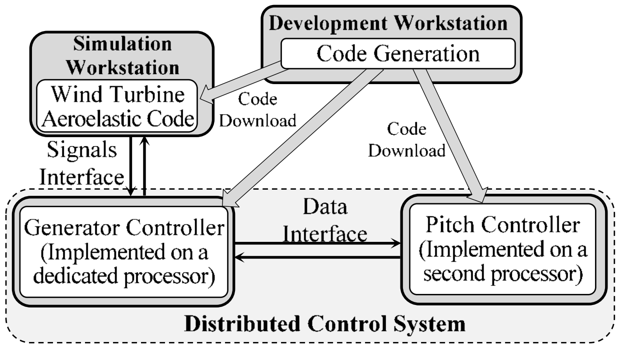

Fig. 3. Proposed general HiL architecture for the large wind turbine

This scheme has been implemented by using a Windows® Workstation with an Intel® Core™ i7 processor as wind turbine simulator equipped with two card MF634 from Humusoft® completing 16 A/D, 16 D/A and 16 I/O channels.

As control hardware, the M1 industrial platform of Bachmann®, which consists of several modules (e.g. MC210 and MX213 as computational units and AIO216 and DIO248 for the interface channels). This is illustrated in Fig. 4. As hard real-time operating system, VxWorks from Wind River is used.

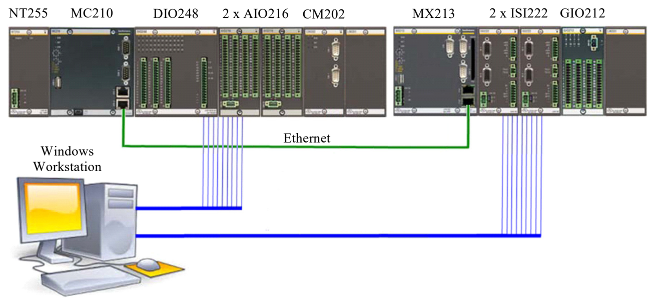

Fig. 4. Implemented HiL architecture

The M1 platform also provides a cross-compiler for Matlab/Simulink such that Simulink block diagrams can be cross-compiled and uploaded to be executed in the Bachmann hardware. This advantage makes possible to use Simulink as development system.

As aeroelastic code for the solution of the wind turbine mathematical model, the software FAST from National Renewable Energy Laboratory, Jonkman and Buhl Jr., (2005), is chosen. It is provided with an Adams-Bashforth (AB4) algorithm as solver, i.e. an algorithm that can be implemented in a real-time task as described in Howe, (1991). In addition, a Matlab/Simulink interface is available such that FAST can be combined with the tools Simulink Coder, Simulink Real Time and Simulink Desktop Real-Time. Thus, FAST can be operated in a soft real time task.

Several large reference wind turbines are available for simulation using FAST, e.g. the NREL 5-MW machine specified in Jonkman et al., (2009), the 10 MW reference turbine from DTU, Bak et al., (2013) and the 20 MW reference turbine from TU-Delft, Ashuri et al., (2016). A summary of most important characteristics of such machines are given in Table 1.

Table 1.Main characteristics of reference wind turbines

	NREL 5MW	DTU 10 MW	TU-Delft 20 MW
Rated wind speed [m/s]	11.40	11.40	10.715
Rotor diameter [m]	126.00	178.30	276.00
Tower hub height [m]	90.00	119.00	160.20
Blade length [m]	61.50	86.35	135.00

The final configuration for the HiL system is shown in Fig. 5.

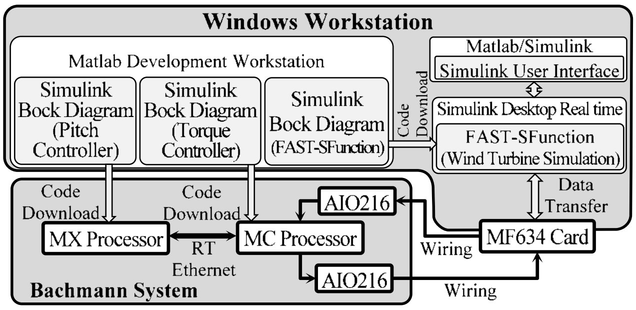

Fig. 5. Implementation of the HiL architecture

The fix integration time-step in FAST is denoted now as h_{FAST} and the necessary real-time to complete the solver calculation of one step is symbolized with h_{rt}. In order to meet real-time requirements for the simulation, the relationship $h_{rt} < h_{FAST}$ must be satisfied. Input signals at the begin of the integration step have to be maintained constant from time-step to time-step until the sampling period finishes and new values of the input signals are required at the begin of the next sampling period. Therefore, the whole sequence of time-steps must finish inside the sampling time but letting enough free margin such that the controller can compute the next values of the control signals before the next simulation step begin. On the other hand, the values of the output signals are delivered by the solver after mh_{rt} steps. Hence, these output values are already available for the control signal calculation when the control tasks are triggered by the wake-up signal. Control tasks have (T_s - m h_{rt}) sec. to deliver the control signals. The time interdependences are given in Fig. 6.

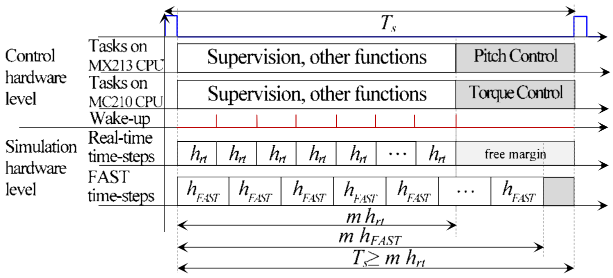

Fig. 6. Time dependences of the HiL system

5. APPLICATION EXAMPLES

The HiL system is used by different Master projects, which have in common following learning objectives:

- becoming familiar with real-time hardware and software
- making experience with real-time control design
- developing advanced algorithms for real-time operation
- real-time testing of control algorithms for wind turbines

The use of the HiL system is illustrated in Fig. 7.

5.1 Implementation of a Supervisory Control System

The fundamentals of supervisory control of wind turbines are described in Gambier et al., (2019). In the present work, a simplified supervisor including only few states is designed as state machine and implemented in Stateflow. The final scheme is compiled and uploaded to be executed in the MC210 processor. The state chart implemented in Stateflow is shown in Fig. 8.

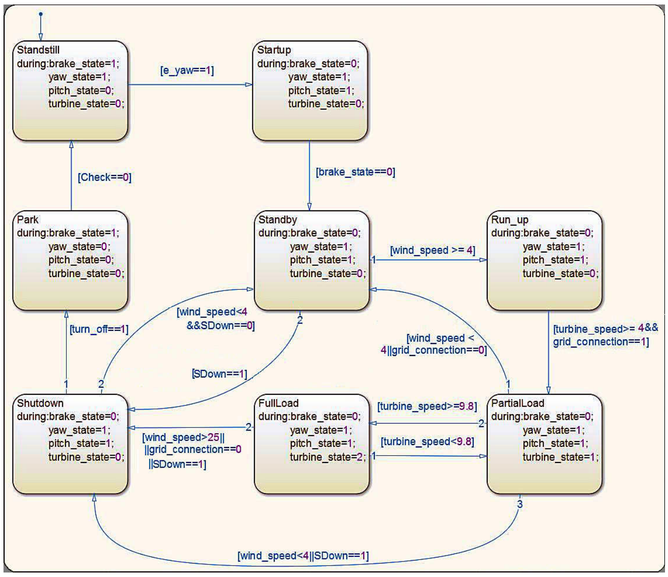

Fig. 8. Stateflow implementation of a supervisor for the wind turbine HiL system

5.2 Implementation of a Nonlinear PID Pitch Control System

Nonlinear PID-Control for the pitch control of wind energy systems has been introduced in Gambier and Nazaruddin, (2018). The control system topology that is implemented to be run in the MX213 processor is given in Fig. 9.

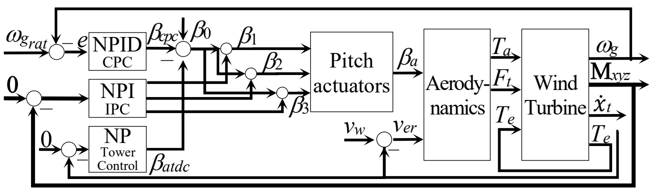

Fig. 9. Nonlinear PID control system for the blade pitch control

A general Simulink block diagram for the implementation of collective pitch control is presented in Fig. 10. Thus, different controllers need only to change the block labelled as PID. This has been used for the nonlinear PID algorithm but also for the controller presented in the next subsection.

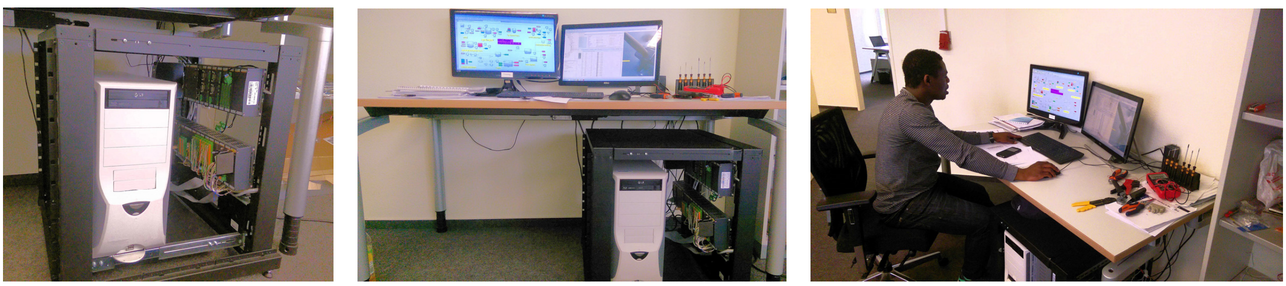

Fig. 7. Hardware-in-the-Loop system for wind turbine control in practical use

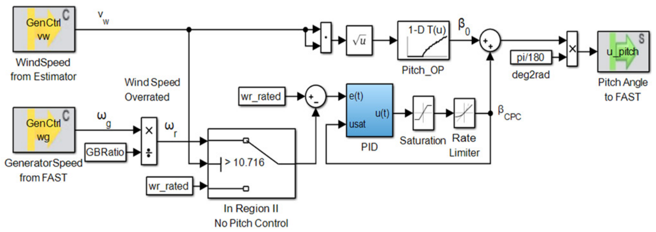

Fig. 10. Simulink block diagram for the implementation of collective pitch control (CPC) in the Bachmann system.

The PID controller is implemented as a time-discrete system. The general I/O representation, which is given in the form

$$u_r(z)=\frac{T(z^{-1})}{P(z^{-1})}r(z) \text{ and } u_y(z)=\frac{Q(z^{-1})}{P(z^{-1})}y(z), \qquad (1)$$

with $u(z) = u_r(z) + u_y(z)$, is realized by using a ladder structure given by

$$\begin{bmatrix}\mathbf{x}_r(k+1)\\ \mathbf{x}_y(k+1)\end{bmatrix}=\begin{bmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}\end{bmatrix}\begin{bmatrix}\mathbf{x}_r(k)\\ \mathbf{x}_y(k)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_r & \mathbf{0}\\ \mathbf{0} & \mathbf{B}_y\end{bmatrix}\begin{bmatrix}r(k)\\ y(k)\end{bmatrix} \text{ and} \qquad (2)$$

$$u(k)=\begin{bmatrix}\mathbf{C}_r & -\mathbf{C}_y\end{bmatrix}\begin{bmatrix}\mathbf{x}_r(k)\\ \mathbf{x}_y(k)\end{bmatrix}+\begin{bmatrix}d_r & 0\\ 0 & -d_y\end{bmatrix}\begin{bmatrix}r(k)\\ y(k)\end{bmatrix}. \qquad (3)$$

Matrices **A**, $\mathbf{B}_y$, $\mathbf{B}_r$, $\mathbf{C}_y$, $\mathbf{C}_r$, d_y and d_u are computed by canonical realization of polynomials *T*, *Q* and *P* from (1).

Controller implementation consists of two periodic real-time tasks. The first task computes the control signal directly after reading and conditioning $y(k)$. The second task is used to update the states after the control signal is sent to the output port. The concept is illustrated in Fig. 11 as a possible implementation.

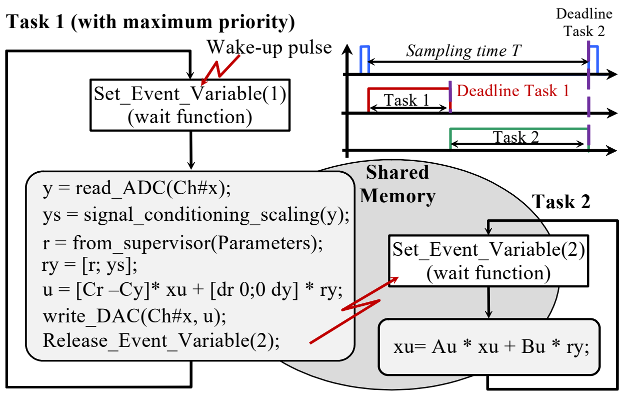

Fig. 11. Controller implemented with two real-time tasks

5.3 Implementation of a Pitch Control System using Fractional Order Mathematics

The fractional PID (FOPID) controller is given e.g. in El-Khazali, (2013). Therefore, only a very short formulation is introduced for the sake of completeness. A standard PID can be transformed in a FOPID by introducing fractional exponents λ and μ to the Laplace variable *s*, i.e.

$$G_{PID}(s)=K_p+K_i(1/s^{\lambda})+\frac{K_d}{(1/s^{\mu})+T_d}, \qquad (4)$$

where $(1/s^{\gamma})$ is an integrator with γ real and positive. In general, a continuous integro-differential operator of positive or negative real order γ can be expressed as

$$X(s)=K\,s^{\gamma}, \quad \gamma\in\mathbb{R}\,. \qquad (5)$$

On the other hand, a fractional-order integro-differential operator is an integer infinite-dimensional operator (see Vinagre et al., (2000)) such that a practical implementation requires an integer order approximation. Here, only the Crone formula (French acronym for *commande robuste d'ordre non-entier*) is used, Cois et al., (2002). It consists in a transfer function of *n* poles and zeros given by

$$X(s)=K_n\prod_{k=1}^{n}\frac{1+s/\omega_{z,k}}{1+s/\omega_{p,k}}, \qquad (6)$$

where K_n is a constant used to obtain a gain equal to 0 dB at the frequency 1 rad/s when $K = 1$. $\omega_{z,k}$ and $\omega_{p,k}$ are zeros and poles within the frequency range $[\omega_l, \omega_h]$ inside of which, the approximation is valid. Poles and zeros are recursively computed by using the sequence

$$\begin{aligned}\omega_{p,k}&=\omega_{z,k-1}\eta, \quad k=1,\dots,n\\ \omega_{z,k}&=\omega_{p,k-1}\alpha, \quad k=1,\dots,n\end{aligned}, \qquad (7)$$

where $\omega_{p,1}=\omega_l\sqrt{\eta}$, $\alpha=(\omega_h/\omega_l)^{\gamma/n}$ and $\eta=(\omega_h/\omega_l)^{(1-\gamma)/n}$. For the real-time operation, (6) is implemented as a difference equation, where the coefficients are computed by using z-transform for a given order *n*.

Notice that the first order filter in the D part of (4) can be avoided for FOPID because the implementation of the fractional order integrators is done by an approximation that normally includes filters such that the limitation of high frequencies is guaranteed. Here, the filter is maintained such that setting the exponents to $\nu = 1$ and $\mu = 1$, the FOPID becomes a PID.

As an anti-windup mechanism, an automatic external reset configuration combined with a back-calculation strategy is used. In addition, parameters of the PID controllers can also be variable, for instance in the case of nonlinear PID control or gain scheduling approaches. Integrating all mentioned characteristics in only one control diagram, the scheme of Fig. 12 is obtained.

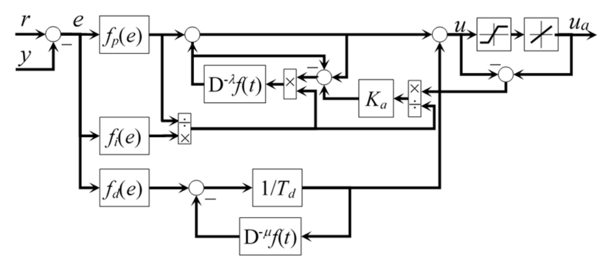

Fig. 12. Generalized PID in automatic reset configuration

This scheme is summarized as Simulink-block is then used instead the block PID in Fig. 10 and cross-compiled for the real-time execution on the MX213.

6. SIMULATION RESULTS

In addition to the examples related to master works of students, which are explained in the previous section, a simple simulation example of a 20 MW reference wind turbine with a collective pitch control system is presented in the following. The machine is operated in Region III (over rated wind speed) for an effective wind speed varying between 11 and 20 m/s with variable turbulence between 5 and 15%. The wind profile is shown in Fig. 13.

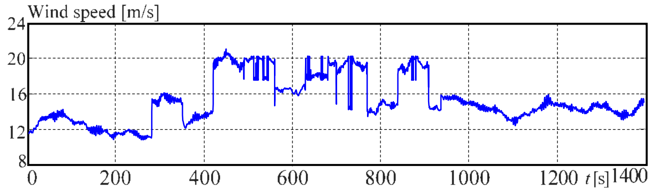

Fig. 13. Wind speed profile for simulation purposes in FAST

The main objective of this simulation experiment is to illustrate the operation of the HiL system. That is, the simulation of a large wind turbine by using a high-resolution model can run in a real-time environment, the communication between simulation hardware and control hardware is correct and maintains the real-time conditions and the available time on the control hardware is enough to compute the control signals. This can be observed in the simulation results (Fig. 14 and Fig. 15). The correct work is appreciated because on the contrary the large machine will behave with very large oscillations or even becomes unstable.

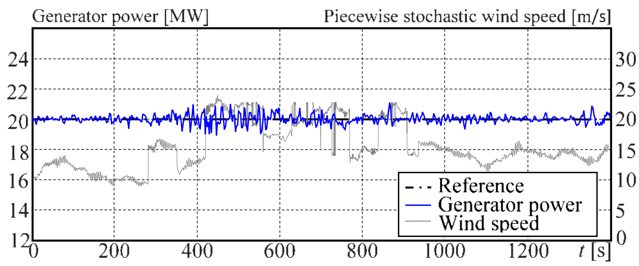

Fig. 14. Generator power maintained constant at 20 MW

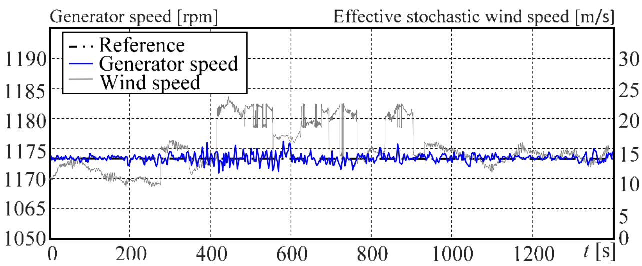

Fig. 15. Generator speed maintained constant at 1173.7 rpm under stochastic wind

Finally, the control signals computed in the MX213 CPU with rate limited anti windup are transferred to the simulation hardware in order to maintain constant the power when the wind speed is over rated. This signal is presented in Fig. 16.

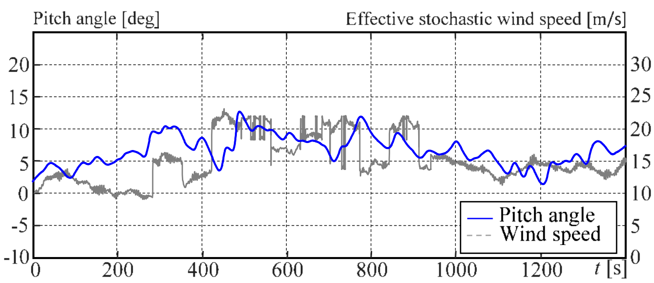

Fig. 16. Pitch angle signal provided by the controller

7. CONCLUSIONS

In this work, a Hardware-in-the-Loop architecture for the real-time simulation and control of wind turbines as support for students working in Master thesis is presented.

Hardware and software chosen for the real-time operation are described. In particular, a Bachmann system as hardware and FAST as software are used.

Three examples are described. The first one is the implementation of a supervisory control system by using Stateflow and cross-compiled for the execution in the processor MC210. The second example is the real-time implementation of a PID controller for collective pitch control in the MX213 processor, implemented as two periodic real tasks. The third example consists of an advanced fractional order PID controller with a special anti-windup procedure.

Practical experience shows that the HiL system satisfies all expectations. On one hand, advanced control algorithms can be tested in a more realistic environment. On the other hand, it is an attractive facility that is able to motivate students and to provide particular experience, which will be useful in the professional activities as control engineers of wind energy systems. The system can now be augmented and improved is several directions, e.g. including real actuators as additional hardware.

REFERENCES

Ashuri, T., Martins, J. R. R. A. and Zaaijer, , (2016), 'Aeroservoelastic design definition of a 20 MW common research wind turbine model', *Wind Energy*.

Bacic, M., (2005), 'On hardware-in-the-loop simulation', *44th IEEE Conference on Decision and Control and European Control Conference 2005*, IEEE, Seville.

Bailey, M. and Doerr, J., (1996), 'Contributions of hardware-in-the-loop simulations to Navy test and evaluation', *Proceedings of the Society of Photo-optical Instrumentation Engineers*, vol. 2741, pp. 33-43.

Bak, C., Zahle, F., Bitsche, R., Kim, T., Yde, A., Henriksen, L. C., Natarajan, A. and Hansen, M., (2013), 'Description of the DTU 10 MW Reference Wind', DTU Wind Energy, Roskilde.

Bélanger, J., Venne, P. and Paquin, J. N., (2010), 'The what, where and why of real-time simulation', *PES General Meeting*, IEEE.

Cois, O., Lanusse, P., Melchior, P., Dancla, F. and Oustaloup, A., (2002), 'Fractional systems toolbox for Matlab: applications in system identification and Crone CSD', *41st IEEE Conference on Decision and Control*, Las Vegas.

El-Khazali, R., (2013), 'Fractional-order PIλDμ controller design', *Computers and Mathematics with Applications*, vol. 66, pp. 639-646.

Evans, M. B. and Schilling, L. J., (1984), 'The role of simulation in the development and flight test of the HiMAT vehicle', NASA, Hampton.

Gambier, A., (2004), 'Real-time control systems: a tutorial', *Asian Control Conference*, ACA, Melbourne.

Gambier, A. and Balzani, C., (2019), 'The challenge of teaching control of wind turbines in a civil engineering school', *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 212-217.

Gambier, A., Li, Z. and Yang, Q., (2019), 'Supervisory control of a wind energy system by using a hybrid system approach', *IEEE*

45th Conference of the Industrial Electronics Society, IEEE, Lisbon.

Gambier, A. and Nazaruddin, Y. Y., (2018), 'Nonlinear PID control for pitch Systems of Large Wind Energy Converters', *2018 IEEE Conference on Control Technology and Applications*, IEEE, Copenhagen.

Grega, W., (1999), 'Hardware-in-the-loop simulation and its application in control education', *29th ASEE/IEEE Frontiers in Education Conference*, IEEE, San Juan.

Howe, R. M., (1991), 'A new family of real-time redictor-corrector integration algorithms', *Simulation*, pp. 177-186.

Isermann, R., Schaffnit, J. and Sinsel, S., (1999), 'Hardware-in-the-loop simulation for the design and testing of engine-control systems', *Control Engineering Practice*, vol. 7, no. 5, pp. 643-653.

Jonkman, J. M. and Buhl Jr., L. M., (2005), 'FAST User's Guide', NREL, Battelle.

Jonkman, J., Butterfield, S., Musial, W. and Scot, G., (2009), 'Definition of a 5-MW Reference Wind Turbine for Offshore System Development', NREL/TP-500-38060, NREL, Golden, Colorado.

Leisten, C., Jassmann, U., Balshüsemann, J., Hakenberg, M. and Abel, D., (2017), 'Design and analysis of a MPC-based mechanical hardware-in-the-loop system for full-scale wind turbine system test benches', *IFAC-PapersOnLine*, vol. 50, pp. 10985–10991.

Neshati, M., Zuga, A., Jersch, T. and Wenske, J., (2016), 'Hardware-in-the-loop drive train control for realistic emulation of rotor torque in a full-scale wind turbine nacelle test rig', *2016 European Control Conference*, EUCA, Aalborg.

Roscoe, A. J., Mackay, A., Burt, G. M. and McDonald, J. R., (2010), 'Architecture of a network-in-the-loop environment for characterizing AC power-system behavior', *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1245-1253.

Sala, A. and Bondia, J., (2006), 'Teaching experience with hardware-in-the-loop simulation', *IFAC Proceedings Volumes*, vol. 39, no. 6, pp. 123-128.

Sarhadi, P. and Yousefpour, S., (2015), 'State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software', *Int. Journal on Dynamics and Control*, vol. 3, pp. 470-479.

Sobota, J., Goubej, M., Königsmarková, J. and Cech, M., (2019), 'Raspberry Pi-based HIL simulators for control education', *IFAC PapersOnLine*, vol. 52, no. 9, pp. 68-73.

Steurer, M., Li, H., Woodruff, S., Shi, K. and Zhang, D., (2004), 'Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation', *IEEE Power Electronics Specialists Conference*, IEEE, Aachen.

Tejado, I., Serrano, J., Pérez, E., Torres, D. and Vinagre, B. M., (2016), 'Low-cost Hardware-in-the-loop testbed of a mobile robot to support learning in automatic control and robotics', *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 242-247.

Temeltas, H., Gokasan, M. and Boğosyan, S., (2005), 'Hardware in the loop robot simulators for on-site and remote education in robotics', *International Journal Engineering*, vol. 22, no. 4, pp. 815-828.

Vinagre, B. M., Podlubny, I., Hernandez, A. and Feliu, V., (2000), 'Some approximations of fractional order operators used in control theory and applications', *Fractional Calculus and Applied Analysis*, vol. 3, no. 3, pp. 231-248.