

Design and Validation Methodology of the Control System for a Particle Beam Size Measurement Instrument at the CERN Laboratory

Jonathan Emery¹, Ana Barjau², Bernd Dehning¹, Juan Herranz Alvarez^{1,2},
 Pierre-Jean Lapray³, Matteo Macchini¹

Abstract—A new generation of the Beam Wire Scanner (BWS) instrument is being developed for the Large Hadron Collider (LHC) and its injector chain at the European Organization for Nuclear Research (CERN). This is an electro-mechanical system based on a Permanent Magnet Synchronous Motor (PMSM), a robust resolver and a high precision optical encoder. This system is located in underground installations and has to withstand large irradiation levels provoked by particle beams. The control electronics is situated away from it to avoid the destructive impact of ionizing particles, introducing long distances of up to 250m for powering and measuring the system. Challenges arise when imposing large acceleration to obtain a scanning speed of the wire of no less than 20ms^{-1} and above, with a target position accuracy of $5\mu\text{m}$ rms. This paper details the methodology of the system and control design, which includes performance evaluation benchmark at different design stages and the target electronics development. Finally, the first field results performed inside the Super Proton Synchrotron (SPS) accelerator machine are presented.

I. INTRODUCTION

The European Organization for Nuclear Research (CERN) is designing and operating particle accelerators to perform fundamental research and discovery. To achieve these goals, particle beams are produced, characterized and optimized with specific instruments developed by the Beam Instrumentation group and its international collaborations. As the probability of colliding particles increases as the beam size decreases, the capability of accurately measuring this parameter is essential. The Beam Wire Scanner (BWS) performs this function and helps to calibrate other instruments which may suffer from various limitations.

A. Instrument working principle

The wire scanner is an interruptive beam size measurement device which interacts with proton beams, inside vacuum pipes, by means of a very thin wire of $30\mu\text{m}$ at a displacement speed of no less than 20m/s . This interaction provokes the generation of secondary particles radiation caught by scintillating material which converts it into light. This light is then amplified by a photo-multiplication principle, to finally have a current proportional to the secondary particle concentration at the wire location. The reconstruction of the beam transversal density of particles is performed by plotting the current intensity versus the position of the wire.

B. System architecture description

The wire-scanner system is composed of four parts shown on the block diagram in Fig.1, working together to achieve the measurement.

¹J.Emery, Member, IEEE, B.DeHning, J.Herranz, M.Macchini are with the European Organization for Nuclear Research (CERN), Beam instrumentation group, Geneva, Switzerland
 Contact : jonathan.emery@cern.ch

²A.Barjau, J.Herranz are with the Universitat Politcnica de Catalunya, Barcelona, Spain

³P.J.Lapray is with the Universite de Haute Alsace, Laboratoire MIPS, Ecole Nationale Supérieure d'Ingenieurs Sud Alsace, Mulhouse, France

The electro-mechanics, called the scanner, and the particle sensor on the lower part of the figure are situated in the accelerator underground tunnels where the particle beams are travelling. The control electronics and the acquisition crate shown on the upper part are protected from the damaging effect of ionising radiations on the accelerator surface installations. Details on this architecture can be found in [1]. Connections between surface and tunnel are made of copper cables and fiber optics carrying information and power, to ensure simple and robust design against radiation damages and single event upset to electronics. Moving the electronic to a safe location does not come without drawbacks when the displacement consists of hundreds of meters. The cost to pay is long connections to install and maintain, a limitation in the system bandwidth and a higher sensitivity to noise and perturbations compared to short range systems.

II. METHODOLOGY FOR THE CONTROL DESIGN AND ITS IMPLEMENTATION

The followed methodology is to start with the system modelling and the control simulation, then retrofit the experimental results obtained using a laboratory prototyping platform. The last steps are the implementation into the final electronics. At every development stage, the control performance is evaluated and compared to track degradation introduced by the implementation. To minimise the performance degradation, an optimization of the parameters is performed to compensate for the distortions introduced by technological limitations such as finite precision and quantization errors.

This allows to estimate the future behavior of the scanner, to track errors and inconsistencies in the implementation and to evaluate performance at each stage, i.e. from the modelling to final field tests. Fig.2 presents these stages and their associated tools and optimization techniques. Performance evaluation methods will be discussed in Sec. VII.

A. Modelling of the actuator and control

This allows to experiment the fundamental concepts and solutions by means of simulation, not only for the control system, but also for other areas such as data processing and decision, system

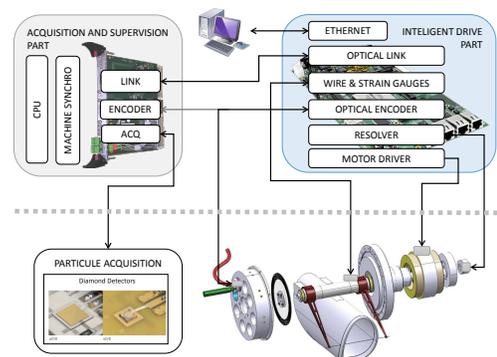


Fig. 1: Wire-Scanner System Overview

condition monitoring and on-line fault detection. The modelling will be described in Sec.IV and the control design Sec.V.

B. Implementation on a laboratory development platform

This second step allows the experimentation of the controller with a real prototype, the parameters tuning, the evaluation of the accuracy of the electro-mechanical system and the electrical modelling. The rapid development platform from DSpace has been selected for its compatibility with the modelling and optimization tools. It provides real world input-output with compiled Simulink code as real-time tasks running on a dedicated platform. At that stage, differences were observed and retro-fitted into the model. Experimental optimization of the control parameters was then possible by means of an iterative optimizing method called Particle Swarm Optimization (PSO) through the Python interface.

C. Digital implementation of the control law and Simulation of individual functions

After validation using Simulink prototyping, the control law has to be translated to a language compatible with the final system. For this purpose, all the individual blocks of the controller have been written in VHSIC Hardware Description Language (VHDL). To verify the behavior of each block, individual test benches have been produced to validate the functionalities at the Register Transfer Level (RTL) using the widely used verification tool called Modelsim from Mentor Graphics. This was done for the position and current acquisition, Clarke/Park transformations, generic PID controller, current decoupler and pulse width generations.

D. Functional testing of the VHDL Control law using Co-simulation Simulink - Modelsim

Once the individual description and testing of each controller block has been done, a global simulation for the controller can be performed. To avoid rewriting the system model into another language, VHDL for instance, Co-simulation between Simulink and Modelsim software is used.

Two copies of the system models were instantiated in Simulink, one controlled by the simulink controller and the other one by the VHDL controller. It was then possible to compare the behavior of all variables in parallel as shown in Fig.3. For instance the system position, speed, phases current, but also internal variables like PID controllers, status and outputs can be compared and the differences can be quantified.

E. Control algorithm of the digital processing platform

The final step consists in the integration of the controller into the digital processing platform based on FPGA. To enable the use of software optimization tools, as in the early stages of the design,

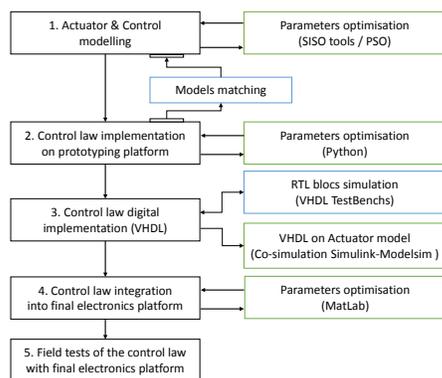


Fig. 2: Modelling and design stages of the control architecture and optimization

we developed the system where MATLAB can read and modify the controller parameters and variables on-line.

III. MOTION PATTERN DESIGN

The selected approach is a movement control using a predefined motion pattern with a setpoint tracking technique. This allows us to design a movement fulfilling the following requirements:

- Wire tangential speed of 20m s^{-1} at beam crossing
- Stabilized speed during beam crossing
- Beam crossed once per movement, no multi-turn system
- Symmetrical movement
- Acceleration below damage limit of the mechanics

The resulting motion pattern implemented is a shaft rotation of π rad, with an acceleration phase, a constant speed phase and a final deceleration phase. Since the prototype has a fork length of 182mm, the rotational maximum nominal angular speed at beam crossing has to be 110 rad/s. To minimize the beam-wire interaction at rest position, the fork initial and final positions are kept far from the beam center. Fig.4 shows this movement according to the mechanical dimensions of the prototype.

The vibrations induced by the motion pattern are minimized by carefully building the profile from sinusoidal jerk. The position is calculated by integrating the profile three times. This approach allows the generation of relatively smooth acceleration from a rather low number of parameters. Fig.5 illustrates this process, from the jerk on top, to the position at the bottom. The motion obtained is a satisfactory starting point, but it suffers from high peak acceleration values, introducing design constraints at the level of motor and shaft sizing. One possible solution is to eliminate the restriction of crossing the beam only once per half scan cycle (see Fig.4 (A)) and allow two beam crossing instead (see Fig.4 (B)). This would provide a larger acceleration and deceleration travel, thereby reducing the peak acceleration.

IV. SYSTEM MODELLING

The actuator of the wire-scanner is a complex system consisting of a Permanent Magnet Synchronous Motor (PMSM), which is driven by a three-phase power inverter. The inverter signal is smoothed by a Sine Filter to reduce the high-frequencies and avoid resonances in the cable. A solid resolver, driven by a custom made circuit and a high precision optical encoder, was included in the system to measure the angular displacement of the shaft. Also, a magnetic brake was used to recover a stable position in case of control or power failure.

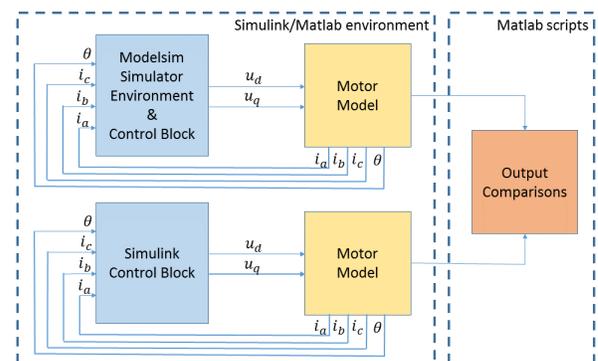


Fig. 3: The co-simulation block diagram. The two system descriptions are evaluated in parallel. θ is the angular position of the rotor, $i_{a,b,c}$ are the 3-phase currents and $u_{d,q}$ are the voltages in the rotor oriented reference.

The system modelling for the control design and its optimization includes:

- Permanent magnet synchronous motor Parker K500150
- Magnetic brake perturbation calculated by simulation
- Solid resolver from Rotasyn
- Current sensors
- Inertias of rotors, shaft, forks and optical disk
- Power inverter stage and its sinus filter
- Power transmission cables

Table I summarizes the main parameters of the system used in the model. Detailed discussion on this subject can be found in [2].

TABLE I: System parameters for the modelling

Characteristics	Values	Units
Motor continuous torque	7.92	$[N \cdot m]$
Motor peak torque	19.5	$[N \cdot m]$
Motor Torque constant K_t	0.39	$[N \cdot m \cdot A^{-1}]$
Motor voltage constant K_e	0.39	$[V \cdot rad^{-1} \cdot s^{-1}]$
Total inertia	$1.35 \cdot 10^{-3}$	$[kg \cdot m^2]$
Static friction torque	0.1	$[N]$
Cable length	10 to 300	$[m]$
DC bus voltage	300 to 400	$[V]$
Control loop frequency	16.129	$[kHz]$

A. Permanent Magnet Synchronous Motor

The PMSM is a 3-phase motor with permanent magnets on the rotor. By using appropriate sequence to supply the stator phases, a rotating field in the stator is created. Due to its high power density and small size, the PMSM has evolved in recent years as the preferred solution for high dynamics speed and position control drives.

To drive it, it is necessary to provide a combination of three stator currents to generate a magnetic flux into the windings, and so a torque. For this kind of devices, it is a common use to refer to the internal state on a two-dimensional axes by means of the Clarke-Park transformation [2].

The motor equations can be summarized as follows:

$$\frac{d}{dt} i_d = \frac{1}{L_d} u_d - \frac{R_s}{L_d} i_d + \frac{1}{L_d} \omega_e (L_q i_q) \quad (1a)$$

$$\frac{d}{dt} i_q = \frac{1}{L_q} u_q - \frac{R_s}{L_q} i_q - \frac{1}{L_q} \omega_e (L_d i_d + \Phi_f) \quad (1b)$$

$$\frac{d}{dt} \omega_m = \frac{(K_T i_q) - \omega_m B}{J} \quad (1c)$$

$$\frac{d}{dt} \theta_m = \omega_m \quad (1d)$$

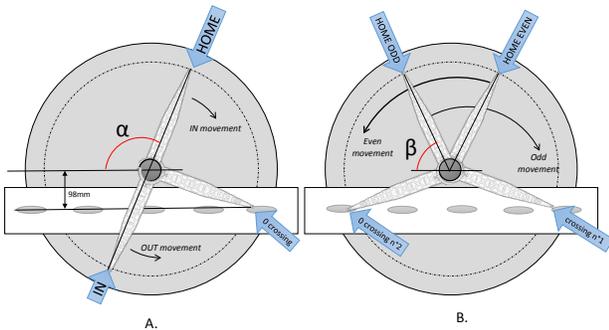


Fig. 4: Wire-Scanner wire strokes

being u_d , u_q , i_d and i_q the stator voltages and currents projected on the d and q axes, R_s , L_d and L_q the equivalent resistances and inductances, Φ_f the flux generated by the stator, K_T the motor constant, J the system inertia, B its damping factor and ω and θ , speed and angular position, respectively, expressed either in the electrical or in the mechanical frame.

B. Resolver

The angular position of the shaft is provided by a resolver, which, with respect to an optical encoder, has the advantage of delivering an absolute measure. One of the major contributions to the accuracy of a resolver is the resolver-to-digital conversion (RDC). In order to perform this operation on the BWS position feedback sensor, the AD2S1210 chip was chosen. After filtering the two input signals properly, an estimated value for θ is computed by the AD2S1210 by means of a feedback loop. Its dynamics are specified in the datasheet and modelled consequently. The dynamic response of the RDC is inversely proportional to the chosen resolution: since the angular speed has to be computed from the angle derivative, both a good resolution and a quick dynamic response are necessary. As a trade off, a 14-bit precision was set in the system according to Table II.

TABLE II: Resolver performance per resolution

n. of bits	10	12	14	16
$\Delta\theta$ [rad]	0.006163	0.001534	0.000383	0.000096
$\Delta\omega^*$ [rad/s]	98.175	24.544	6.136	1.534
T_R^{**} [ms]	0.1	0.25	0.5	2

* with a frequency $f = 16kHz$

** output rise time

C. Power supply

To provide power to the system, a Pulse Width Modulation (PWM) three-phase driver was chosen. It is a typical solution for motor drivers since the PMSM behaves as a passive RLC filter itself, with low-pass dynamics, cutting off the higher frequencies contained in the PWM signal. The amplification is performed by means of a 3-channel inverter realized with six IGBTs. The transfer function of the power supply stage consists in the inverter dynamics, represented by the delay introduced by the 3-phase bridge, τ_d . This can be approximated as $\tau_d = T_s / 2$, being $T_s = 1/f_{PWM}$ the sampling period. For our model, a second-order Padé approximation was performed, hence:

$$e^{-s\tau_d} \approx \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2} \quad (2)$$

where k_1, k_2 coefficients take the values given in Table III.

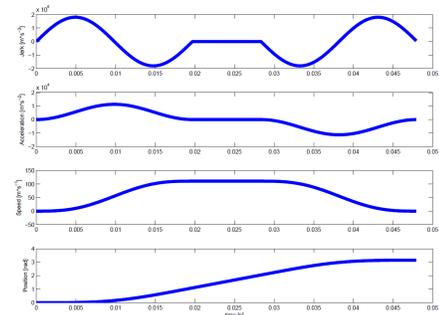


Fig. 5: Position (4th plot), Speed (3rd plot) and Acceleration (2nd plot) profiles obtained from a sinusoidal Jerk pattern (top plot)

TABLE III: Coefficient for the Padé approximation of order 1 and 2

$n = 1$	$k_1 = \frac{\tau_d}{2}, \text{ other } k_i = 0$
$n = 2$	$k_1 = \frac{\tau_d}{2}, k_2 = \frac{\tau_d^2}{12}, \text{ other } k_i = 0$

D. Power transmission cable

Within the CERN facility, several wire scanners are installed in different positions, to guarantee a continuous and reliable diagnostic system. All motors are installed next to the vacuum pipe, while their power supply unit is situated on the surface for maintenance purpose and to prevent it from being damaged by radiations. Due to this, the length of the cable separating the motor and its drive varies between a couple of meters up to 250m. Naturally, the specifications require the system to be entirely stable and to show similar performances for all the possible cable lengths.

This part rises two main complications for the cable modelling and compensation. Firstly, the dynamic response of a power cable, besides being hard to model in all its parameters, is strictly dependent on the load it is connected to. Indeed, when evaluating the effects of a part of an electrical circuit, we usually assume it to be independent from its load, so that no current flows in it. In the case of a PMSM, being $R_S \simeq 2\Omega$, this assumption is not acceptable. We thereby need to model the cable according to the specified motor. Secondly, being the control variables the two voltages provided by the inverter and related to the $d-q$ frame, the cable model will be situated between two nonlinear transformations, and the input-output equation becomes:

$$u_{dq0,out} = K_{C/P} T K_{C/P}^{-1} u_{dq0,in} \quad (3)$$

where $K_{C/P}$ is the Clarke-Park transformation matrix, and T is the transfer function of the cable. Assuming that it present an attenuation behavior on the same phase and a cross-talk effect between phases, we can write:

$$T = \begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix} \quad (4)$$

Developing the calculations (see appendix A of [2]), we obtain:

$$\frac{u_{q,out}}{u_{q,in}} = t_D - t_C \quad (5)$$

which is not only linear time-invariant, but also decoupled in the $d-q$ frame.

The evaluation of the t_D and t_C dynamics was performed through the linear analysis tool of Simulink, and verified experimentally starting from the cable model in Fig.7.

A milliohmeter was used for the verification of the resistance, while the input-output comparison for a frequency-swept signal fed into the system was used to verify the transfer function.

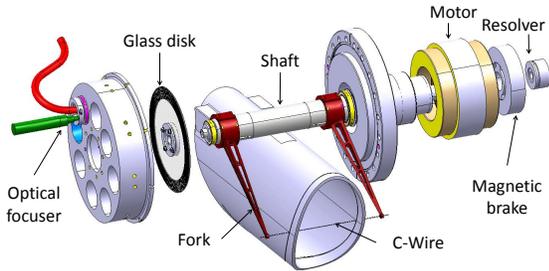


Fig. 6: Wire-Scanner Electro-Mechanical System

Fig.8 shows the final comparison between the cable model and the experimental results; there is a good agreement.

V. CONTROL DESIGN

The type of motor selected and the fast dynamics of the scanning action drove the controller design to a field-oriented architecture, which allows controlling the torque accurately at any shaft position.

A. Field-Oriented control

Field-Oriented Control (FOC), also called Vector control, is a variable-frequency drive (VFD) control method where the stator currents of a three-phase AC electric motor, as well as for a PMSM, are identified as two orthogonal components that can be visualized with a vector.

It acts on three hierarchical levels represented by position, speed and current. On the current level, two controllers are implemented: one for the direct current, which is proportional to flux, and one for the quadrature current, which is proportional to torque. The four control loops are regulated by PID controllers. Besides the typical architecture of a field oriented controller, some complications were added in order to fit the specific design. They include a dynamic decoupler for the motor currents, a Steady-State Kalman Filter to increase the speed acquisition quality, an improved version of PID embedding a variable structure and an anti-windup function, a compensation method for the dead-time in the inverter and a feedforward action to improve the dynamic response of the system.

B. Current decoupling

A coupling exists between the radial (d) and tangential (q) rotor frame currents. Decoupling them allows to increase the control performance by lowering the torque fluctuations. Starting from the current equation of the PMSM in the Laplace domain (6a) (6b), we can eliminate the last contribution of each equation by using feedforward action. For this, the new inputs in Eq. (6c) and (6d) are created.

$$sI_d = \frac{1}{L_d}U_d - \frac{R_s}{L_d}I_d + \frac{1}{L_d}\omega(L_qI_q) \quad (6a)$$

$$sI_q = \frac{1}{L_q}U_q - \frac{R_s}{L_q}I_q + \frac{1}{L_q}\omega(L_qI_q + \phi_f) \quad (6b)$$

$$U'_d = U_d - \omega_e(L_qI_q) \quad (6c)$$

$$U'_q = U_q - \omega_e(L_dI_d + \phi_f) \quad (6d)$$

After implementing the decoupling block using this concept, the current loops assume the following convenient first-order form:

$$I_d = \frac{1}{L_d s} + R_s U_d \quad (7a)$$

$$I_q = \frac{1}{L_q s} + R_s U_q \quad (7b)$$

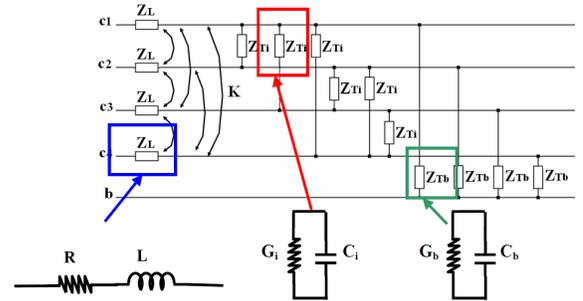


Fig. 7: Discrete model of a power cable

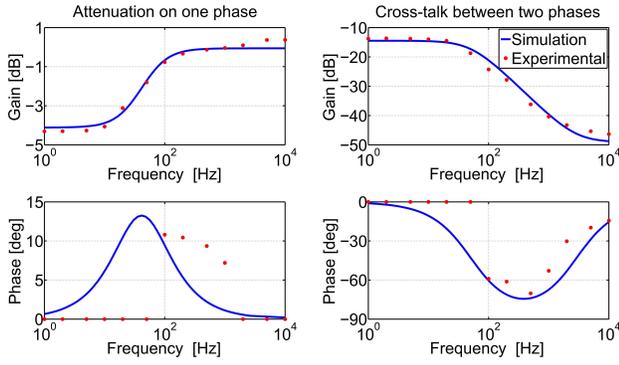


Fig. 8: Effects of the power line on the same phase and on different phases for variable cable lengths

C. Feedforward action

A good model of the process to be controlled allows the implementation of a feedforward action. Feedforward is a popular control technique complementary to feedback. While feedback is necessary to grant robustness towards disturbances in the control, feedforward is used to improve performance by minimizing feedback loop error, a priori. Feeding directly the right profile to follow to both the speed and current regulators, the delay in their action is cancelled and the response of the system is visibly quicker.

The implementation of this technique is actually one of the most important and efficient improvement included in the motion controller. The performance increment allowed the tracking to be faster and more accurate compared to feedback action only.

The implementation for our first prototype uses offline calculation for the feedforward action, presented in Fig. 9 (bottom). In the future, real time calculation will be implemented to allow simple control of the scanner speed operation using only one profile table, as shown in Fig.9 (top).

D. Steady-State Kalman Filter (SSKF)

For position-controlled motors, the most popular technique to compute the speed value, that represents an internal state, is the derivation of the position value. Being a discrete value, this produces a quantization noise that largely overcomes the mechanical vibrations of the system that may reduce the system performance and stability. To solve this problem, some typical techniques were implemented and tested in [2], as well as a Steady-State Kalman Filter, which represents a more complex but effective solution.

Kalman filtering allows to obtain both the advantages of a precise estimation performed following the principles of sensorless control [5] and the robustness of exploiting an actual measure through

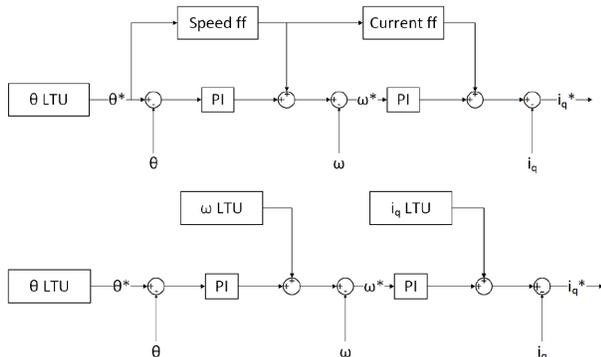


Fig. 9: Feedforward calculated in real time (top) and offline (bottom)

a position sensor (resolver). The Kalman filter is based on the following stochastic discrete-time model of the mechanical system:

$$\begin{aligned} x_n &= Ax_{n-1} + b\tilde{a}_n \\ y_n &= c^T x_n + \eta_n \end{aligned} \quad (8)$$

where

$$x = \begin{bmatrix} \theta \\ \omega \end{bmatrix} \quad A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} \frac{T_s^2}{2} \\ T \end{bmatrix} \quad c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Since the dynamic system given by (8) is linear and time-invariant, the only variable element in the prediction and correction sections of the algorithm is the gain vector g_n . Therefore, it is possible to replace the vector's sequence g_n with its limit g when $n \rightarrow \infty$. In this way a *Steady-State Kalman Filter* (SSKF) is used, with the advantage reducing the number of mathematical operations. The g value is tuned as in [6].

Figure 10 presents a schematic of the filter, while Fig.11 shows a comparison between the performance of usual techniques, which include analog and digital filters, dirty derivators and the proposed method, based on the *Integrated Absolute Error* (IAE).

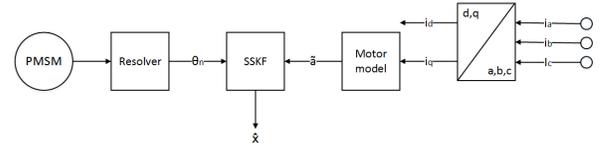


Fig. 10: Conceptual schematic of the Steady-State Kalman Filter

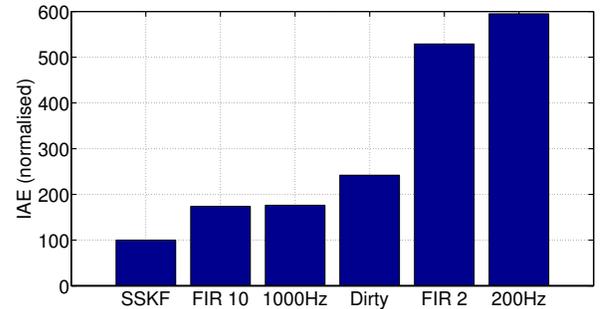


Fig. 11: Speed acquisition methods comparison

E. PID controllers

1) *Conceptual design:* For the BWS control system, several improvements were made on the typical PID architecture in order to achieve better performance and stability. As is usually the case for PID controllers where the integral gain is not zero, an anti-windup function is necessary in case of saturation. In our case, a conditioned anti-windup architecture was chosen, as it is more convenient for hardware implementation in digital controller, and presents slightly better performances compared to the tracking architecture [2].

The imperfect action of the filter also provokes a power distribution on the PWM harmonics and its multiples. These power peaks at high frequencies, amplified by the ringing effects of the cable, affected negatively the stability of the controller, that presented oscillating behaviors when connected to the power line with an 100 m long cable. These effects were mostly observed during the steady-state and the rest position of the actuator and not much during the scanning action. To overcome this issue, a variable architecture for the PID was proposed: the proportional gain, which

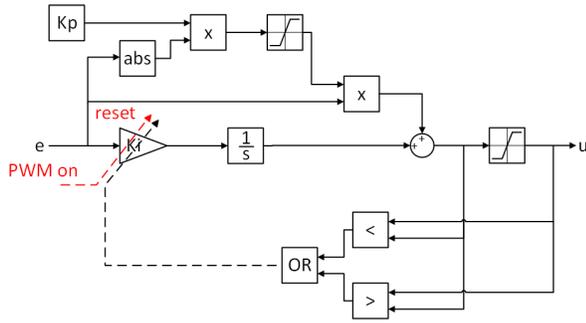


Fig. 12: Architecture of the variable structure, anti-windup PI controller with external reset

is more sensitive to input noise, is set to zero in steady-state and increased linearly with the error until it reaches its saturation point, set to $e = 1$, where it assumes its nominal value. This allowed us to attenuate the disturbances during the slow motion operations.

Finally, a reset function for the integral action was implemented and associated the PWM trigger, in order to avoid the 'kick' effect due to saturation while the drive is disconnected.

The final PID design is shown in Fig.12.

2) *Parameters optimization using MATLAB:* In order to obtain a good dynamic response from the actuator, the gains of the four PID controllers need to be optimized. Considering that only their proportional and integral actions were implemented, and that the two current loops can be considered to be equivalent once decoupled, we need to design 6 independent parameters to describe the controller.

Since a reliable model was available for the process, the SISOTOOL package of Matlab was used for a first, numerical optimization of the control parameters. The model includes all the dynamics of the various components of the scanner, listed in IV, and a second-order Padé approximation of the delays introduced by the control loop and the inverter. The specifications for the controller synthesis can be summarized as follows:

- The steady-state error should be zero for all the three loops.
- The rise time to a step input should be lower than $1ms$ for the current loop, $10ms$ for the speed loop and $1s$ for the position loop.
- The maximum allowed overshoot was set to 10% for the current loop, 20% for the speed loop and zero (no overshoot) for the position loop.
- To guarantee adequate robustness to the system, a phase margin of 40° and a gain margin of $10dB$ were introduced.

The obtained parameters are indicated as $K_{P\ num}$ and $K_{I\ num}$ in Table IV.

3) *Parameters optimization using PSO:* Open-loop and closed-loop PID tuning experimental methods, such as Ziegler-Nichols or ATV techniques, have the common defect of bringing the system to its stability limit. In order to avoid these issues, an iterative optimizing method called Particle Swarm Optimization (PSO) was considered [7] in order to refine the parameters choice by means of an experimental procedure. PSO is a computational method that can be used to find an optimal solution to a problem, based on the minimization of a cost function. It is basically constituted by a particle filter, where each particle represents a control coefficient. The particle position was initialized varying up to 10% the values obtained through SISOTOOL and updated at each iteration using a linear combination of the distance from the best position occupied so far by the particle and the best position found globally. In this case, the cost function is represented by the sum of the Integrated Absolute Errors of various portions of the signals for speed and current compared to their reference. The single IAEs were scaled by means of a weight coefficient to make the criteria

independent from the absolute value of the reference values. The PSO algorithm was implemented in Python on the DSpace machine and used to produce an optimal and reliable set of coefficients for the PI controllers. More details can be found in [2].

The parameters computed by the PSO algorithm are indicated as $K_{P\ PSO}$ and $K_{I\ PSO}$ in Table IV. The same values were also used to validate the model through a laboratory closed-loop test. Fig.13 shows a good agreement between simulated and experimental dynamics. It is worth specifying that the rise time constraints set in the previous synthesis are respected, but the overshoot limit is not due to a more aggressive speed controller.

TABLE IV: Parameters for the PI controller

Loop	$K_{P\ num}$	$K_{I\ num}$	$K_{P\ PSO}$	$K_{I\ PSO}$
Current	0.07	180	0.045	150.72
Speed	0.05	6	0.27	2.38
Position	15	8	24.31	9.47

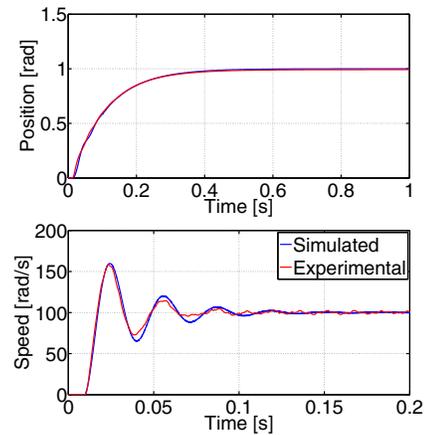


Fig. 13: Validation of the model - step response

VI. INTELLIGENT DRIVE PLATFORM

We developed a dedicated control platform with custom electronics. The principal motivations for this decision were to attain:

- the most specialized system for this application.
- a complete understanding and the hand on all the inner processing without 'black boxes'.
- a lower dependability to third party software and hardware.
- a long term investment by developing in-house know-how for future upgrades.

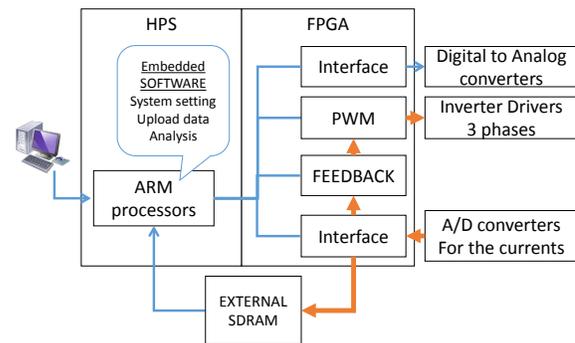


Fig. 14: Overview of the System-On-Chip architecture for the Wire-Scanner.

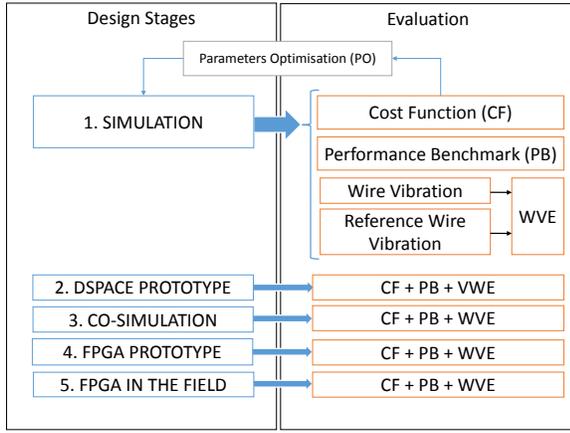


Fig. 15: Performance evaluation at different design stages

Based on these requirements [4], the intelligent drive is built around two parts: a driving/measurement interface board and a central logic platform.

The driving and measurement part provides all necessary information for the central logic to perform properly the control and the monitoring of the scanner. It includes electrical interfaces with all the controlling parts previously described in Section IV. Digitalized values of all these interfaces are provided to the central logic and allow to act on the driver via Pulse-Width Modulation (PWM) techniques. The phases motor currents are measured by Hall effect transducers and digitalized at $250kSPS$ with 16 bit . A long driving and measuring distance resolver is used to measure the shaft angular position.

The central logic platform is the intelligent piece where the control and monitoring algorithms are being implemented. A Field Programmable Gate Arrays (FPGA) with an Altera Arria V chip was selected. It contains logic blocks plus a complete System-On-Chip (SoC) with a dual-core ARM Cortex-A9 processor. Fig.14 shows the internal architecture, where we can see in red the critical block dedicated to measuring and to the control. The motor control was implemented fully in VHDL using logic elements, to avoid the dependability of the critical functions to any running embedded software. On the other hand, the processors are responsible for giving a network user access to the measured values and to the control parameters. The operating frequency of the global system inside the FPGA is $50MHz$. The conversion frequency for generating PWM signal is $16,129KHz$, and the time used to compute a new 3-axes voltage output after receiving the new input current/position is about 128 system clock cycles. This extremely fast computation allows the minimization of the time between a new set of input acquisition to the update of the outputs, and the reduction of eventual instabilities.

For the numerical representation, the VHDL language proposes fixed-point library [11] for computing and data representation, with a 32-bit number format including 16-bit for the fractional part. This representation was selected as it is a good compromise between precision and logic utilization for our application [2].

VII. PERFORMANCE EVALUATION ACROSS DESIGN STAGES

The evaluation of the system performance of each design stage has been performed using three algorithms comparing motion signature against the reference as shown in Fig.15. The inputs to these data analyses are the position, the speed, the torque and the jerk of the shaft shown in Fig.16.

The first algorithm is the Cost Function (CF) developed in [2] to tune the parameters during the control design and based on the specific requirement of the application to track potential instabilities. This method proposes to focus the attention to critical

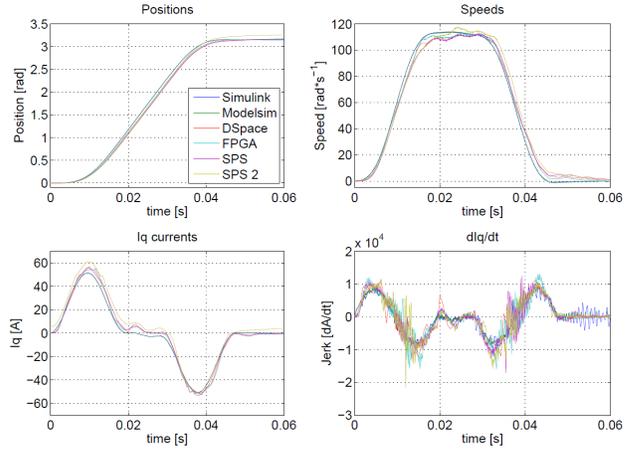


Fig. 16: Motion patterns at different design stages

zones by giving them more weight to them and by calculating the sum of the deviation from the expected position with (9) and (10).

$$ICM = \frac{1}{N_{pts}} \int_{t=0}^{end} |e|dt, ISS = \frac{1}{N_{pts}} \int_{t=t_1}^{t_2} |e|dt \quad (9)$$

$$CostFunc = \frac{1}{Sum(coeff)} \cdot (\alpha_1 \cdot ICM_\theta + \alpha_2 \cdot ICM_\omega + \alpha_3 \cdot ICM_{Iq} + \beta_1 \cdot ISS_\omega + \beta_2 \cdot ISS_{Iq_1} + \beta_3 \cdot ISS_{Iq_2}) \quad (10)$$

A complementary method, called Performance Benchmarking (PB), focuses attention on the action of the actuator on the mechanism and its smoothness during beam crossing. Other regions of interest from the CF algorithm are given less weight. To perform it, the derivative of the rotor current I_q , an image of the torque, is used as the indication of the discontinuities given to the system. The speed stability is also used as described in (11).

$$PB = \frac{1}{Sum(coeff)} \cdot (\gamma_1 \cdot ICM_{\frac{dI_q}{dt}} + \gamma_2 \cdot ISS_\theta + \gamma_3 \cdot ISS_\omega + \gamma_4 \cdot ISS_{Iq_1} + \gamma_5 \cdot ISS_{\frac{dI_q}{dt}}) \quad (11)$$

In the third method, the discrete dynamic model of the wire developed in [3] has been used to evaluate the performance of the controller action on the system. The effective angular motion

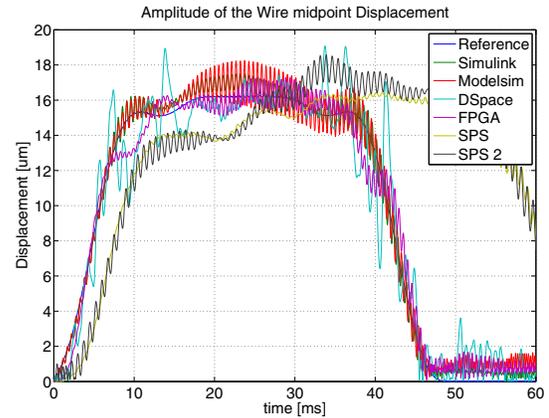


Fig. 17: Wire midpoint displacement at different design stages

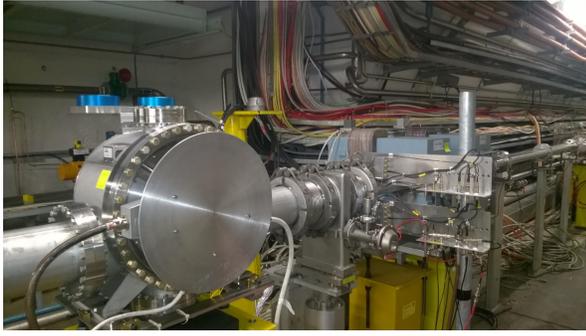


Fig. 18: Wire scanner prototype (left side of the picture) in the SPS accelerator

pattern outputs resulting from each design step are introduced as inputs of the wire dynamic model. The model calculates the motion of the wire central point. Fig.17 shows that motion for each design step. Comparison between each of them against the reference wire motion allows a qualitative evaluation of the design steps performances.

To perform this evaluation, the absolute integral difference between each stage and the reference has been calculated using (12).

$$WVE = \frac{\frac{\alpha_{w1}}{N_{pts}} \int_{t=0}^{end} |e_{r,wire}| dt + \alpha_{w2} \cdot max_{error}}{Sum(coeff)} \quad (12)$$

TABLE V: Performance evaluation results at different design stages using the Cost Function (CF), Performance Benchmarking (PB) and Wire Vibration Evaluation (WVE) methods

Design stages	CF	PB	WVE
Simulation (Simulink)	0.919	253.4	8.992E-07
Co-simulation (Simulink-Modelsim)	0.925	175.4	1.391E-06
DSPACE laboratory prototype	1.166	784.2	3.820E-06
FPGA laboratory prototype	1.596	438.6	1.927E-06
Field tests (SPS)	2.067	450.4	-
Field tests 2 (SPS) - abnormal frictions	3.885	533.0	-

Table V summarizes results for one data set at each design stage. As expected, the CF presents a progressive performance degradation, from simulation to FPGA implementation. On the other hand, Performance Benchmark (PB) reveals an effect of the co-simulation which behaves better than the pure Simulink, a side effect of the precision limitation. This second method and the Wire Vibration Evaluation (WVE) highlight the DSPACE laboratory prototype as the worst system, as it suffers from a limited number of data point with larger noise, that limits its performance.

VIII. CONCLUSIONS AND FUTURE WORK

The increasing need of performance for the instrumentation of the CERN LHC accelerator and its injector chain has driven the design of a high precision beam wire scanner system. To achieve this goal, the electronics and control system have been designed following a precise methodology of defined steps, tools and performance evaluation.

Accurate modelling and control design have been followed to achieve the high performance required for this application up to the realisation of a custom electronics.

The methodology described allows the evaluation of the performance degradation introduced by the controller implementation at each design step. These results may lead to modification or re-design if the degradation would become unacceptable for the system precision. The result of this systematic approach is a high

confidence in the system performance and a precise control of the degradations introduced by the numerous implementations steps.

The wire-scanner control and monitoring electronics have been designed and prototyped. This allowed the concept validation up to the field testing in the CERN SPS accelerator and produced the results presented here. Improvements are still on-going to achieve higher tracking performances and lower measure uncertainties, which will be monitored by the performance evaluation techniques described in this paper.

The systematic design approach presented for the controller will be applied to the other processing parts of the wire-scanner, such as the condition monitoring of the actuator and other measuring devices, the on-line faults detection and decision processing. These parts are foreseen to be introduced in future prototypes for other accelerators of the CERN complex in 2017 and 2018.

REFERENCES

- [1] J. Emery, B. Dehning, C. Morais Pereira, J. Sirvent Blasco, S. Cantin, M. Tognolini, B. Schneider, K. Henzer, M. Starkier, A fast and accurate wire scanner instrument for the CERN accelerators to cope with severe environmental constraints and an increased demand for availability. In : Control Applications (CCA), 2014 IEEE Conference on. IEEE, 2014. p. 1139-1145.
- [2] M. Macchini, Motion control design of a PMSM and FPGA implementation for the Beam Wire Scanner at CERN, Master thesis dissertation, Section BE-BI-BL, CERN, Geneva, CH, 2015.
- [3] A. Barjau, J.Herranz, B. Dehning, Dynamical models of a wire scanner, Journal of Vibration and Acoustics (ASME), to be published.
- [4] J. Emery, Beam Scanner Control, Monitoring and Supplies part, CERN Engineering Specification, EDMS-1318827.
- [5] L. Xu, C. Wang , Implementation and experimental investigation of sensorless control schemes for PMSM in super-high variable speed operation , in Conf. Rec. IEEE-IAS Annu. Meeting, 1998, vol.1, pages 483-489
- [6] A. Bellini, S. Bifaretti, S. Costantini, A Digital Speed Filter for Motion Control Drives with a Low Resolution Position Encoder, UDK 621.313.07:621.371.54
- [7] H. Hu, Q. Hu, Z. Lu, D. Xu, Optimal PID Controller Design in PMSM Servo System Via Particle Swarm Optimization, School of Electrical Engineering, Zhejiang University, China.
- [8] J. Puranen, Induction Motor Versus Permanent Magnet Synchronous Motor in Motion Control Applications: A Comparative Study, PhD thesis dissertation, Lappeenranta University of Technology, 2006.
- [9] K. Florek, Ethernet High Speed with a SoC based FPGA for the Wire Scanner, Bachelor thesis dissertation, Heig-Vd, Switzerland, 2014.
- [10] C. Dufour, V. Lapointe, J. Belanger, S. Abourida, Hardware-in-the-loop closed-loop experiments with an FPGA-based permanent magnet synchronous motor drive system and a rapidly prototyped controller, IEEE International Symposium on Industrial Electronics ISIE, 2008.
- [11] D. Bishop, Fixed point package users guide. Packages and bodies for the IEEE, 2006, p. 1076-2008.