Università di Pisa

Dipartimento di Ingegneria dell'Informazione
Corso di laurea in Ingegneria Robotica e dell'Automazione

Master Thesis

Motion control design of a PMSM and FPGA implementation for the Beam Wire Scanner at CERN

Author:
Matteo Macchini

Supervisors:
Sergio Saponara
Luca Fanucci
Jonathan Emery

2015

Abstract

The *Large Hadron Collider* (LHC) is the world's largest and most powerful particle collider, and the largest single machine in the world. It is designed to collide proton beams with a collision energy of 14 TeV in the center of mass and an unprecedented luminosity of $10^{34} cm^{-2} s^{-1}$.

The strong dependency of the interaction rate on the beam sizes motivates the requirement of an accurate measurement of the beam profiles. The *Beam Wire Scanner* (BWS) belongs to the category of interceptive beam transverse profile measurement instruments. It consists in an electro-mechanical device which measures the transversal beam density profile by moving a thin wire through the beam intermittently. The growing focus on this instrument and the resulting increase in its use during routine accelerator operation has initiated an upgrade of the existing designs.

In order to enable the device to scan particle bunches in the LCH at collision energy, a crossing speed of 20 m/s was included in the specifications for the new design. To provide the necessary torque to the system, a frameless *Permanent Magnet Synchronous Motor* (PMSM) was selected. Several constraints and specifications, including the variable length of the power cable and the necessary minimisation of the vibrations in the wire, make the control of the BWS actuator a challenging task in terms of robustness and performance.

This thesis work concerns the design and the implementation of a digital motion controller for this new improved generation of beam wire scanners. The design process starts with the creation of a dynamic model for the hardware, including the mechanical devices, the various installed sensors and the power supply. Then, a *Field Oriented Control* architecture is proposed and dimensioned to match the design requirements. Several improvements, such as variable structure regulators, Kalman filtering and feedforward actions are included to increase the efficiency and the robustness of the system.

After a preliminary testing phase performed on a prototyping platform in order to validate the design, the firmware for the motion controller was finally implemented on a FPGA board in VHDL language. The application of the proposed control methods during the experimental sessions allowed the new generation of the beam wire scanner to work in nominal conditions for the first time.

Abstract

Il *Large Hadron Collider* (LHC) è l'acceleratore di particelle più grande e potente al mondo. È stato progettato per accelerare due fasci di protoni ad un'energia di collisione pari a 14 TeV nel centro di massa, con una luminosità di $10^{34} cm^{-2} s^{-1}$.

La forte dipendenza della frequenza di collisione dalla dimensione del fascio giustifica la necessità di una misura accurata del suo profilo. Il *Beam Wire Scanner* (BWS) appartiene alla categoria degli strumenti di misura del profilo di fascio trasversale. Si tratta di un dispositivo elettromeccanico che misura il profilo trasversale facendo passare un sottile filamento di carbonio attraverso il fascio di particelle. L'aumento dell'attenzione sullo strumento e l'introduzione di nuove specifiche per il suo funzionamento ha portato ad una proposta di *upgrade* della versione attuale.

Al fine di rendere il dispositivo capace di intercettare *bunch* di particelle a energia di collisione all'interno dell'LHC, le nuove specifice di progetto includono una velocità di picco di 20 m/s. Per fornire la coppia necessaria al raggiungimento di tale velocità, la scelta del'attuatore è ricaduta su un motore sincrono a magneti permanenti (PMSM). Vari vincoli e specifiche, tra cui la lunghezza variabile del cavo di alimentazione e la necessità di minimizzare le vibrazioni nel filo rendono il controllo dell'attuatore del BWS un compito particolarmente complesso dal punto di vista di performance e robustezza.

Questo lavoro di tesi riguarda la progettazione di un *motion controller* digitale per la nuova generazione di beam wire scanner. Inizialmente, verrà elaborato un modello dinamico per l'hardware in questione, comprensivo dei dispositivi meccanici, della sensoristica utilizzata e del modulo di alimentazione. Successivamente, un'architettura di controllo *Field Oriented* sarà considerata e dimensionata per soddisfare le specifiche di progetto. La struttura base del controllore è stata estesa con l'inserimento di una serie di migliorie, inclusi dei regolatori a struttura variabile, un filtro di Kalman e degli anelli di controllo *feedforward.*

A seguito di una fase di testing preliminare su una piattaforma di prototipazione, eseguita per validare il design, il firmware del controllore è stato implementato su un dispositivo FPGA in linguaggio VHDL. L'applicazione del controllo proposto durante le varie sessioni sperimentali ha permesso alla nuova generazione di beam wire scanner di funzionare per la prima volta in condizioni nominali.

Acknowledgements

The last year of my Master degree, carried out at CERN within the BE-BI-BL section, was an exciting journey shared with some marvellous people that I would hereby like to thank.

First and foremost, I would like to express my gratitude to my supervisor here at CERN, Jonathan Emery, for giving me the possibility to start this adventure at a first place. Without his support and his advice, covering a wide field of engineering competences, and the time spent in the detailed revision of these pages, this work couldn't have been carried out. His supervision enriched me both professionally and personally. I would also like to thank Dr. Bernd Denhing for allowing me to join the Beam Loss team, as well as for his valuable interest and feedback on my work. He contributed to make me appreciate the environment and to increase my interest in the field of beam instrumentation.

I can't forget to acknowledge my supervisors in the University of Pisa, Dr. Sergio Saponara, Dr. Luca Fanucci and Dr. Alberto Landi. Their feedback on the ongoing work and on the redaction of this report was truly valuable and useful.

I am also very much obliged to the entire BE-BI-BL section, and in particular to some of its members.
First of all, I would like to thank Emiliano, Luca and William, for their friendship and their help. I would also like to acknowledge Jose and Pierre-Jean, for their great availability and for their precious contributions on some technical aspects. A special thanks needs to be addressed to my colleague and friend Andreas, who was also the main reviser of the aesthetic appearance of this report.

All of my gratitude and love go to the person who revised the final version this work, with the penguins and everything. Most of these words were written sitting at the same table as you, or on the same sofa next to you. Your support and affection were indispensable to achieve this result. This thesis is not only for you, it is because of you.
Thank you Ellen.

This experience also taught me that true friendship can withstand a couple of kilometers of distance. I want to thank my Friends Alessio, Dario, Davide, Federico, Giulio, Matteo and Omar. There is no need to explain why.

Non posso concludere senza ringraziare la mia famiglia, in particolare i miei genitori, i miei adorati nonni, mia sorella e il mio super-nipote Cosimo. È a loro che più di tutti voglio dedicare questo lavoro. Siete sempre la mia più preziosa fonte di orgoglio e motivazione.

Contents

List of Figures

List of Tables

1. Research at CERN

CERN, byname of Organisation Européene pour la Recherche Nucléaire, formerly Conseil Européen pour la Recherche Nucléaire (1952–54), is an international scientific organization established for the purpose of collaborative research into high-energy particle physics. Founded in 1954, the organization headquarters and laboratories sit astride the Franco-Swiss border west of Geneva at the foot of the Jura mountains, and operates expressly for research of a "pure scientific and fundamental character".

From the original 12 signatories of the CERN convention, membership has grown to the present 21 Member States.

CERN has the largest and most versatile facilities of its kind in the world. The site covers more than 100 hectares in Switzerland and, since 1965, more than 450 hectares in France. During its several years of activity, the CERN laboratory grew steadily, building new and more powerful machines and upgrading or redesigning the old ones.

The first accelerators constructed on the site were the Synchro-Cyclotron (SC, 1957) and the Proton Synchrotron (PS, 1959), which played a fundamental role in the development of the theory of the weak force. Experiments at the PS in 1973 demonstrated for the first time that neutrinos could interact with matter without changing into muons; this historic discovery, known as the "neutral current interaction," opened the door to the new physics embodied in the electroweak theory, uniting the weak force with the more familiar electromagnetic force.

They were followed by the Intersecting Storage Rings (ISR, 1971), the Proton Synchrotron Booster (PSB, 1972) and the Super Proton Synchrotron (SPS, 1976). In 1981 the SPS was converted into a proton-antiproton collider based on the addition of an Antiproton Accumulator (AA) ring, which allowed the accumulation of antiprotons in concentrated beams. Analysis of proton-antiproton collision experiments at an energy of 270 GeV per beam led to the discovery of the W and Z particles (carriers of the weak force) in 1983. These results provided experimental verification of the electroweak theory in the Standard Model of particle physics.

In 1989 CERN inaugurated the Large Electron-Positron (LEP) collider, with a circumference of almost 27 km (17 miles), which was able to accelerate both electrons and positrons to 45 GeV per beam (increased to 104 GeV per beam by 2000). LEP facilitated extremely precise measurements of the Z particle, which led to substantial refinements in the Standard Model. LEP was shut down in 2000, to be replaced in the same tunnel by the Large Hadron Collider (LHC), designed to collide proton beams at an energy of almost 7 Teraelectron volts (TeV) per beam. The LHC, expected to extend the reach of high-energy physics experiments to a new energy plateau and thus reveal new, uncharted areas of study, began test operations in 2008, and was the main tool used for the investigation of the Higgs boson, whose existence has been recently proved in 2012.

1.1 LHC and Injection Chain

The Large Hadron Collider (LHC) is the world's largest and most powerful particle collider, and the largest single machine in the world.

It is designed to collide proton beams with a centre-of-mass energy of 14 TeV and an unprecedented luminosity of $10^{34} cm^{-2} s^{-1}$. It can also collide heavy (Pb) ions with an energy of 2.8 TeV per nucleon and a peak luminosity of $10^{27} cm^{-2} s^{-1}$.

Four experimental areas were equipped for observing the proton beams collisions. Two of them consist of general purpose detectors, ATLAS [1] and CMS [2]. The other two are: a detector dedicated also to heavy ions, ALICE [3], and a detector specialized in the physics of the B-meson, LHC-B [4]. The locations of the four areas around the ring are depicted in *Fig. 1.1.* In addition, the experimental setup named TOTEM [5] investigates the p-p total cross section, elastic scattering and diffraction processes. Finally, LHCf[6] experiment uses particles thrown forward by collisions in the Large Hadron Collider as a source to simulate cosmic rays in laboratory conditions.

Despite being the most important and powerful machine in CERN, the LCH is far from being the only relevant attractive of the center. CERN's accelerator complex consists of many different types of linear and circular accelerators and interconnecting transfer lines. At the beginning of the chain, the protons are

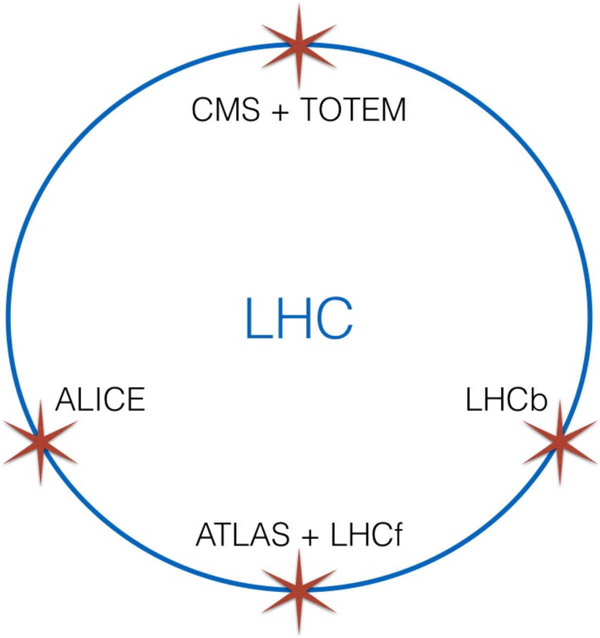


Figure 1.1 Location of the six experiments around the LHC ring

taken from hydrogen gas jet and accelerated in the LINAC2 to the kinetic energy of 50 MeV and transferred to the Proton Synchrotron BOOSTER (PSB)*Fig. 1.2*. The PSB accelerates them to 1.4 GeV and sends them to the Proton Synchrotron (PS). After having reached 25 GeV in the PS, the protons are injected to the Super Proton Synchrotron (SPS) and accelerated to 450 GeV. Finally, they are transferred to the two LHC rings and accelerated for 11 minutes to the energy of 4 TeV. It is foreseen to accelerate to the beam energy of 7 TeV after the stop in 2013 and 2014. The LHC also accelerates and collides lead ions (Pb82+) with the kinetic energy of 2.8 TeV per nucleon. These ions are produced in the LINAC3 and accumulated in the Low Energy Ion Ring (LEIR). Afterwards, they are injected into the PS and follow the same path as the protons up to the LHC.

Several injections from the smaller accelerator are generally needed to fill the subsequent machine so the filing of one LHC ring to the nominal intensity takes in total 4 minutes and 20 seconds. Once the nominal energy is reached, the particles remains circulating in the LHC and colliding inside the four main experiments for several hours *Fig. 1.1*.

The set of the CERN accelerators is very versatile and far from being just the injection system to the LHC. Most of the machines have their own dedicated ex-

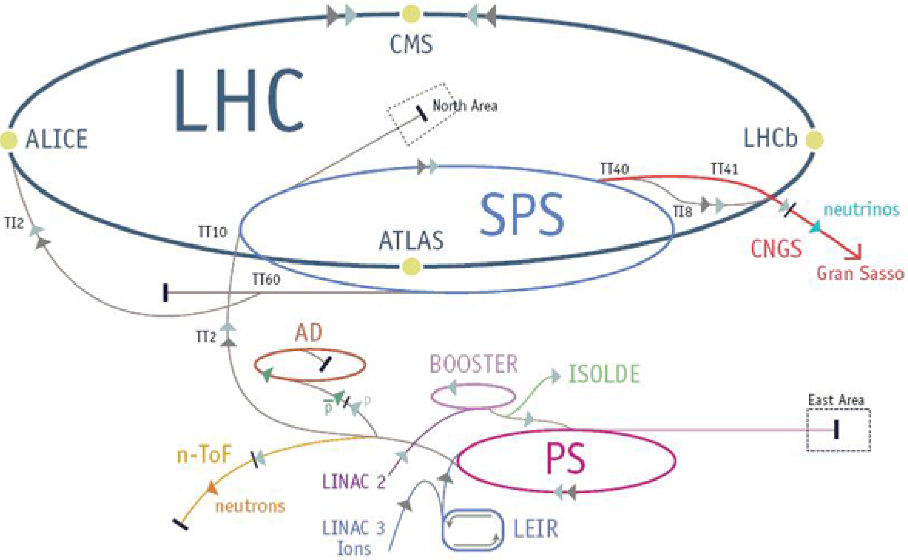


Figure 1.2 The LHC injection system

perimental areas using fixed targets to explore wide range of physics phenomenon. The beam types range from high intensity neutrons for the n-ToF experiment, decelerated anti-protons for anti-matter production to neutrino beams sent to Italy by the CNGS project. *Fig. 1.2* presents a general overview of the system of consecutive accelerators including the LHC.

1.2 Luminosity

In particle physics experiments the energy available for the production of new effects is the most important parameter. The required large centre of mass energy can only be provided with colliding beams where little or no energy is lost in the motion of the centre of mass system (cms). Besides the energy the number of useful interactions (events), is important. This is especially true when rare events with a small production cross section σ_p are studied [7].

The quantity that measures the ability of a particle accelerator to produce the required number of interactions is called the luminosity and is the proportionality factor between the number of events per second $\dot{n}$ and the cross section σ_p:

$$\dot{n} = \mathcal{L}\,\sigma_p \tag{1.1}$$

The unit of the luminosity is therefore $cm^{-2}s^{-1}$. Besides the absolute value of the luminosity, other issues are important for physics experiments, such as the integrated luminosity and the space and time structure of the resulting interactions.

When two bunches containing n_1 and n_2 particles collide, at zero crossing angle, with a repetition frequency f, then the luminosity is given by

$$\mathcal{L} = f \frac{n_1 n_2}{4\pi\sigma_x\sigma_y} \tag{1.2}$$

where σ_x and σ_y are the standard deviation of the Gaussian beam profile.

This shows that the luminosity is inversely proportional to the beam size at the collision points. The beam size must be as small as possible to maximize the number of interaction events.

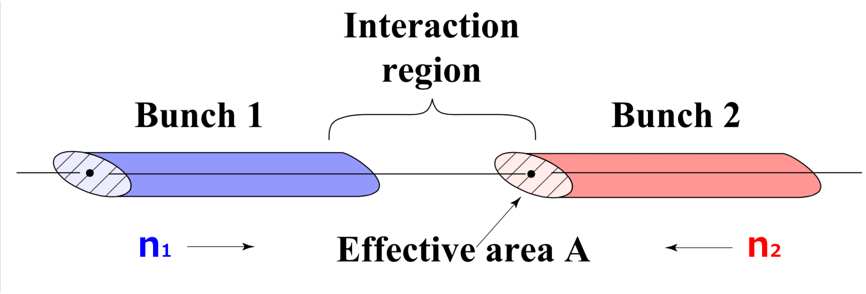
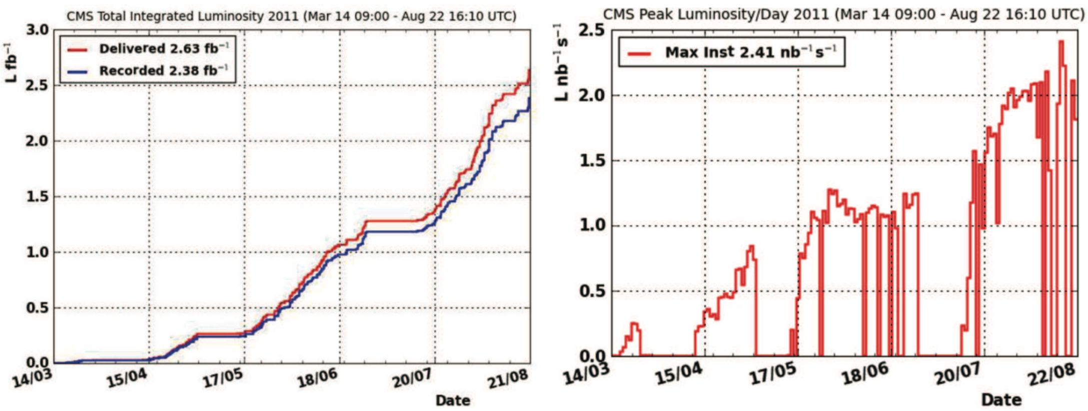


Figure 1.3 Two bunches containing different amount of particles in interaction region

The design luminosity of LHC is $\mathcal{L} = 10^{34}cm^{-2}s^{-1}$ and the total hadronic cross section is $8 \cdot 10^{-26}cm^2$. Consequently the event rate equals to $8 \cdot 10^8 s^{-1}$. *Fig. 1.4* shows the evolution of the luminosity in the LHC during 2011.

Once clarified this, it is logic to understand how a sensor capable of measuring the transverse beam profile, and so its size σ, would be a great tool to estimate the instantaneous luminosity of the beam, which is one of the main performance indexes for a particle accelerator.

Later on, the beam wire scanner will be introduced: it represents a crucial device for beam diagnostic and also the main subject of this work.



(a) *Integrated luminosity* **(b)** *Instantaneous luminosity*

Figure 1.4 Evolution of the luminosity during 2001

1.3 Beam profile measurements

In a particle accelerator, the elements responsible for the guiding of the beam particles along the reference path are the accelerator magnets, often realized through the use of quadrupole magnets. The radiofrequency (RF) cavities, instead, are used to provide acceleration.

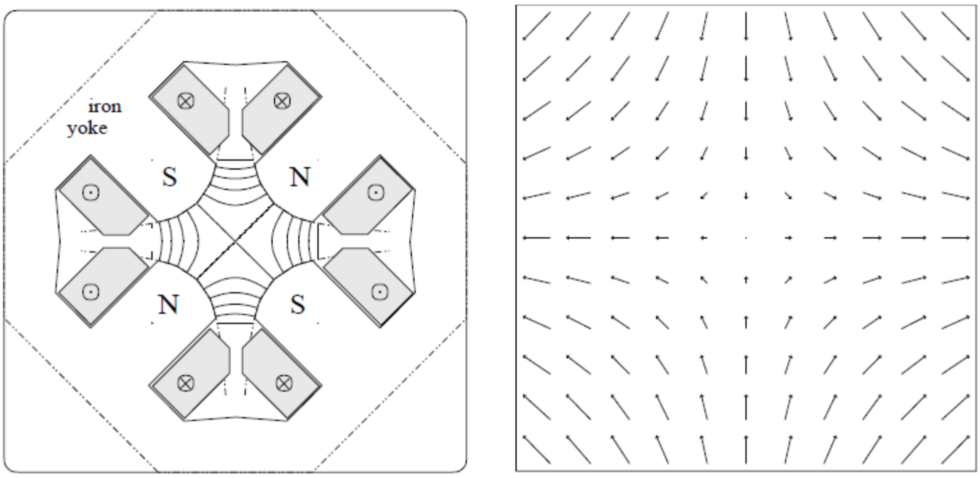


Figure 1.5 Cross section of a quadrupole magnet and its magnetic field

As shown in *Fig. 1.5*, the magnetic field generated by this kind of devices are oriented such that all the particles are affected by it, except for the one in the centre, that remains, anyway, a point of unstable equilibrium. Using this system, then, the particles will be focused on one plane with divergent behaviour, that is why the common design is to use a serie of alternated magnets with opposite polarity, also known as FODO cells. This cells consists on a horizontal focusing quadrupole (F), a drift space (O), horizontal defocusing (D) quadrupole, and another drift space.

The effect of this pattern can be represented like in *Fig. 1.6*.

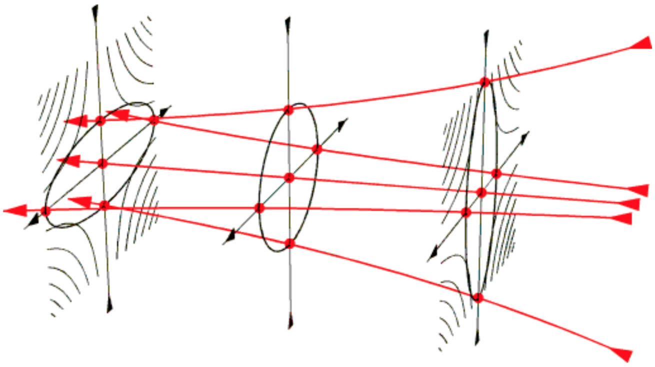


Figure 1.6 Evolution of the beam profile thruogh different focusing magnets

Due to the structure of the focusing magnets, it is reasonable to expect the particles to form a gaussian bidimensional vector on the transversal plane, which two σ components will vary accordingly to the instantaneous magnetic field (i.e. with the position).

The strong dependency of the interaction rate on the beam sizes motivates the requirement of an accurate measurement of the beam profiles. CERN's accelerators are equipped with several beam profile monitors, based on phenomena as:

- secondary emission (SE)
- production of secondaries and Bremsstrahlung
- ionization of rest gas
- synchrotron light

The main devices used for the beam profile measurement within the CERN accelerator system are the following [8]:

- **Beam Wire Scanner (BWS) :** the beam wire scanner belongs to the category of interceptive beam transverse profile measurement instruments.

It uses electro-mechanical parts to move a thin wire of about 30 μm through a particle beam. The secondary particles generated by the interaction of the beam with the wire escape the beam pipe and are detected by a sensor, which in the current system is a scintillator and its associated photomultiplier. The generated current is proportional to the beam intensity at the position of the wire, which allows the reconstruction of the transverse beam intensity profile. This profile is processed to obtain the beam size [9] . It is the most accurate profile monitor adopted in CERN, and because of this reason is it also used to calibrate the other monitors of its category.

- **Beam Gas Ionization monitor (BGI) :** the beam ionizes the residual gas in the vacuum chamber around $8 \div 10$ Pa. An electric field separates the ions and the emitted electrons [10]. Electrons, guided by a magnetic field to limit the angular divergence, or the ions, are collected and the signal is amplified by a multichannel plate (MCP). The electrons produced by the MCP are accumulated onto a screen of phosphorus and one observes the emitted light. This system has a higher efficiency than luminescence but may suffer from space charge effects. This technique is non-perturbative if no gas is injected (as in the case of the LHC), but requires a cross-calibration with the wire scanners.

- **Synchrotron Radiation Monitors (BSRT) :** the synchrotron light monitor profits from the radiation emitted by the accelerated particles to image the transverse beam distribution. This technique avoids any interaction with the beam but, depending on the particles mass and energy, may require the installation of additional bending elements to generate enough radiation. Its absolute calibration is rather difficult but possible by means of using wire scanners [11]. When passing through a magnetic bending element, the beam emits radiation of which the visible spectral part is used and focused on a detector (Intensifier and CCD) by an optical system.

2. Beam Wire Scanners (BWS)

This chapter provides a general description of the beam wire scanner system, from its working principle to the different versions implemented in CERN during the last decades. It is meant to explain the advantages and drawbacks of using such a sensor, the environmental constraints which characterise the installation in a particle accelerator, in particular the Large Hadron Collider, and the typical design specifications for this type of device. Finally, an overview of the limitations of the operational version of the scanner will be given, to justify the motivations of a new, improved design.

2.1 Working principle

The beam wire scanner is described as an electro-mechanical device which measures the transversal beam density profile by means of a thin carbon wire passing through the beam intermittently. When the wire travels through the beam, the interaction between them generates a cascade of secondary particles, which are intercepted by a scintillator coupled with a photomultiplier. Thus, the generated current at the output is proportional to the beam intensity at the position of the wire. The signals acquired for the wire position and the beam intensity are combined in order to reconstruct the transversal beam density profile[12].

The BWS is a perturbative sensor, but its action is limited to the narrow time window in which the scan is performed. It is also the most accurate of the profile monitors, that is why it is used to calibrate the other sensors.

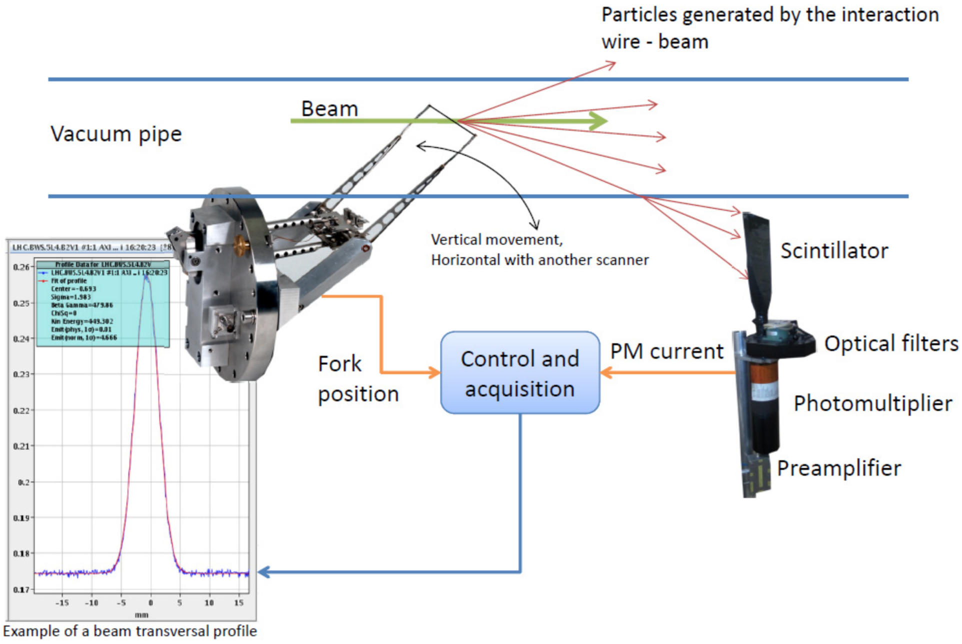


Figure 2.1 Working principle of the Beam Wire Scanner

Another way to acquire information about the particle transversal distribution is to measure the current flowing through the wire. If the wire is made with a conductive material, the traversing particles knock out wire electrons near to its surface (secondary electron emission). This current is also proportional to the number of beam particles present at the wire position.

Two BWS are used to measure beam transversal profiles in horizontal and vertical plans. One wire is horizontal and travels vertically through the beam, producing a vertical profile. The vertical wire that travels horizontally through the beam produces the horizontal beam profile.

2.2 Presentation of the operational system

Before introducing the new conceptual design, it is useful to give an overview of the existing implementation of the system, that has been used in CERN since several years.

The current version is, under the technological point of view, poorer than the new design (the electronics were commissioned in 2007), but it still carried out efficiently its functions until now, which makes its analysis even more fascinating.

2.2.1 Mechanics

Wire scanners are widely adopted in the field of beam diagnostic in all of the main laboratories in the world. One of the common characteristics of these systems is the presence of a mechanical actuator (i.e. a motor) to allow the movement of the carbon wire through the beam pipe and produce interaction between the beam and the wire.

The installation of the actuator in the beam pipe is on itself one of the main challenges of this kind of approach. In fact, the environment inside of a particle accelerator is one of the most extreme ones that can be found on the planet: pressure, temperature and magnetic field are extremely different from the outside, and they represent particularly sensitive parameters to allow the system to work correctly.

Normally, the motion is transferred from the outside to the inside of the vacuum chamber by means of a bellows. This solution is usually preferred to entirely detach the torque-generating parts, in the case of CERN all represented by DC motors, from the actual scanning device, represented by the forks and the wire, limiting the outgassing and avoiding the exposure of sensitive devices like motor windings to the baking procedure.

In terms of mechanics, the wire scanner could be classified in three different types [13]:

- *Linear Displacement Monitors*: Used for a nominal speed of 1m/s. The linear movement is created by a standard dc motor and a rotary to linear mechanism. The position is acquired by a linear potentiometer or by an optical ruler (*Fig. 2.2 left*)
- *Pendulum Movement Monitors*: Also previously called fast wire scanners (FWS), this design can move the wire up to 20m/s (*Fig. 2.2 mid*).
- *Rotating Movement Monitors*: Used for speeds up to 6m/s (*Fig. 2.2 right*).

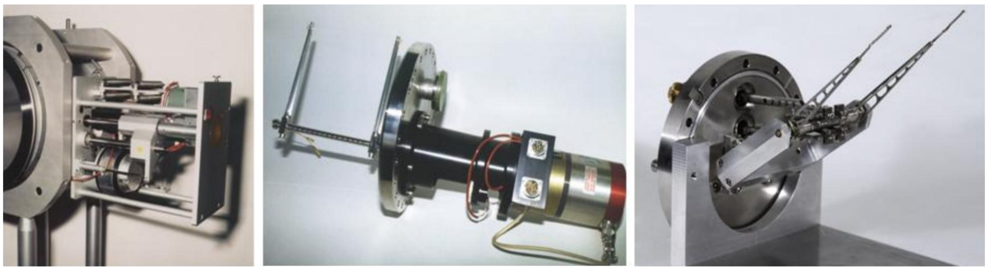


Figure 2.2 Different designs for the existing versions of the BWS

From now on, the overview will be focused on the pendulum movement monitor.

2.2.2 Position sensor

The position measurement is performed by means of a rotary potentiometer, and then converted to a digital value by a 16-bit ADC located in the motion control board [14].

In the case of the FWS, the complex mechanics and the limited accuracy of the potentiometer force us to calibrate this type of scanner. This calibration is performed with a bench that is mounted around the scanner; the particle beam is substituted by a laser whose position is well known [15]. The laser beam is split into two parallel beams of nearly equal intensity distant by 2.7 mm by means of special optics, through which the wire travels at high speed. In the other side a large fixed mirror is positioned. The carbon wire scanning through the beams creates thus twice a shadow detected by one single photodetector,

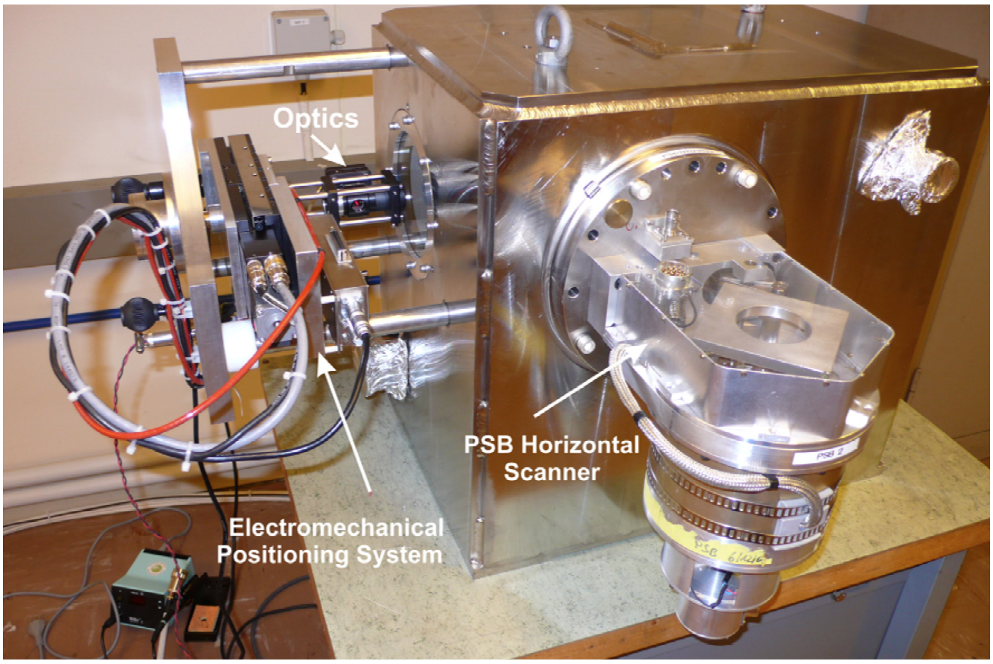


Figure 2.3 Calibration bench with a horizontal PS booster scanner

resulting in two negative peaks and otherwise constant signal. Correlating the measured peak positions with the real laser beam position allows the creation of a correction table.

2.2.3 Power supply

For electromagnetic interference reasons and simpler architecture, two generations of wire scanners are using a linear converter instead of a switching converter as today's state of art. The linear power supply is based on a bridge of two PA50, 40 A, 100 V linear power operational amplifiers, featuring 200 kHz of power bandwidth, a 50 V/s slew rate and a 400 watts of internal power dissipation. An integrated capacitors based energy storage unit allows to balance the peak power during the milliseconds scale scans.

2.2.4 Motion control

The Wire Scanner Motion Control Card (WSMCC) is a unique electronic board designed to control up to 4 multiplexed scanners. When the user requests the action of one of them, the card switches automatically to the appropriate channel.

The system is provided with a position reference, computed through a look up table, and performing the control implementing a single feedback loop with proportional action (P) on the acquired position by means of an analog controller. The controller is implemented on an electronic board and includes 3 FPGAs for digital signal treatment and real-time implementation.

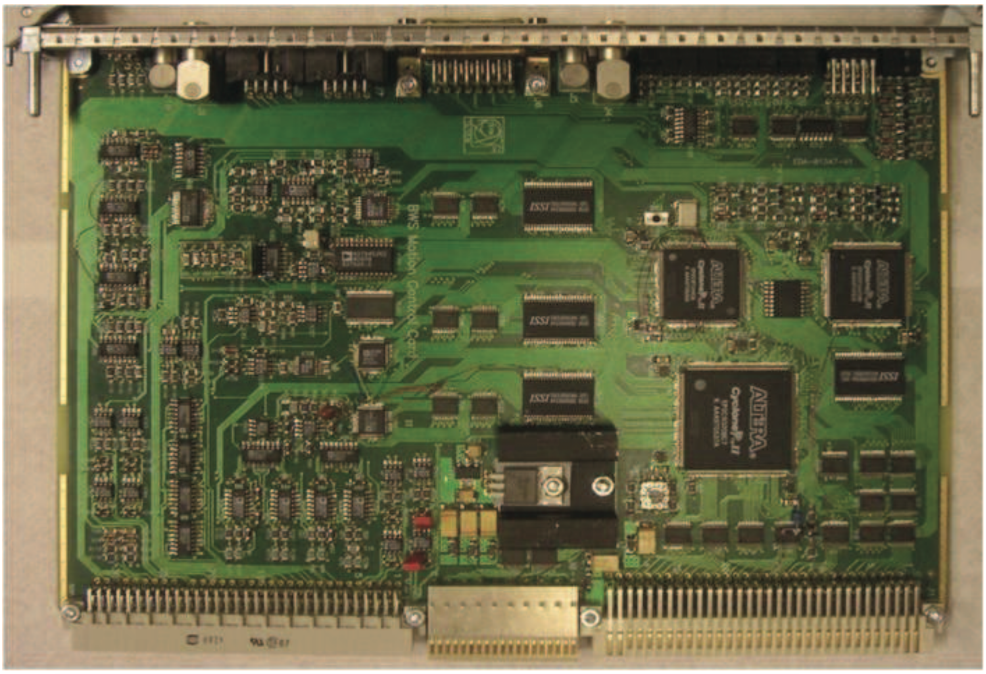


Figure 2.4 The Wire Scanner Motion Control Card

The MCC signal acquisition chain of the photomultiplier current digitise its signal with an ADC after passing through a logarithmic amplifier to extend the dynamic range of the 14 bits initial data width. The software converts it back on a linear scale for display. The low speed acquisition chain is used to determine the average beam size across an accelerator, whereas the bunch by bunch systems give multiple profiles at once.

2.2.5 Scintillator and photomultiplier

The secondary particles produced by the beam-wire interaction are detected by a scintillating material which produces photons. The scintillator is coupled to a filter and a photomultiplier tube (PMT) which has two basic stages: the conversion of photons into electrons and the electron multiplication which enhances the signal level in order to improve the SNR.

2.2.6 Installations

Several wire scanners are installed in the accelerators complex at CERN in different version to fit the requirements set by the acceleration stage. In *Table 2.1*, an overview of the current displacement of the devices is given.

Machine	Scanner type	Number	Scintillators	Max. Speed [m/s]
PSB	Rotating Fast	8	1 for 2 scanners	20
PS	Rotating Fast	5	2 per scanner	15-20
SPS	Rotating	6	1 for 2 scanners	6
SPS	Linear	4	1 for 2 scanners	1
LHC	Linear	8	1 for 4 scanners	1

Table 2.1 Position and characteristics of the beam wire scanners installed at CERN

2.2.7 Limitations

The described system presents some drawbacks, that motivated the Beam Loss section to design an improved version of the scanner:

- The complexity of the mechanical system, especially in the case of the fast wire scanners, prevents an efficient modelling and an accurate measure of the wire position. This is mostly caused by mechanical reducer and cams.

- The mechanical system is not optimized to minimize the vibrations - forks, mechanical play contribute to the uncertainty of the wire position.

- For high intensity beams the energy deposited by the incident particles on the wire may be sufficient to melt or sublimate the wire. This is particularly true for very small beams, in which case the energy transferred from the beam to the wire has potentially damaging effects.

- The wire can melt or sublimate also due to energy transferred by the beam to the wire by means of electromagnetic field coupling inside the cavity created by the scanner tank. The use of ferrite has reduced this effect in the current design.

2.3 Specifications for the new design

In 2008, a new project for the renewal of the beam wire scanner design has been launched. The goal is the realisation of an improved device, that should be able to integrate the perks of the old version with mechanical, electronic and computer upgrades, to match the increasing need of performance and to allow the utilization in the LHC at full power (7 TeV) in safe conditions.

2.3.1 Wire position accuracy requirement

The beam size measurement requirement for the LHC beams is derived from consideration of minimal observed beam size changes and the absolute accuracy needed for the luminosity determination. The accuracy and reproducibility should be better as 1% (δ_{rel}) on the beam size determination. This error consists of a systematic (μ_σ) and statistical (σ_σ) contribution according to:

$$\delta_{rel} = \mu_\sigma + \sigma_\sigma \tag{2.1}$$

If we assume that the systematic error can be eliminated through calibration, we can write:

$$\mu_\sigma \approx 0 \tag{2.2}$$

In the case of a wire scanner, the relative error will depend on two contributions: the error on the amplitude ($\sigma_{\sigma_{amplitude}}$), coming from the secondary shower detector, and the error on the position measure ($\sigma_{\sigma_{position}}$). Assuming that these errors are random and uncorrelated, it leads to:

$${\sigma^2}_\sigma = \sigma_{\sigma_{amplitude}} \pm \sigma_{\sigma_{position}} \tag{2.3}$$

If we can suppose that the two contribution, from position and amplitude, have the same weight on the error of 0.5%, and assuming a SNR of 100 for the photomultiplier, as confirmed by the experiments, the relative statistical error on the beam size is:

$$\frac{\sigma_{\sigma_{amplitude}}}{\sigma_{amplitude}}(SNR_{amplitude} = 100) = 0.005 \tag{2.4}$$

$$SNR_{position}\left(\frac{\sigma_{\sigma_{position}}}{\sigma_{position}} = 0.005\right) \approx 50 \tag{2.5}$$

Finally, considering $\sigma_{LHC_{min}} = 160\mu m$, the minimum beam size expected in the LHC, it yields:

$$\Delta_{position} = \frac{\sigma_{LHC_{min}}}{SNR_{position}\left(\frac{\sigma_{\sigma_{position}}}{\sigma_{position}} = 0.005\right)} = \frac{160\mu m}{50} = 3.2\mu m \tag{2.6}$$

2.3.2 Wire travelling speed requirement

The travelling speed requirement for the wire is chosen to overcome two critical issues. The particle losses generated by wire-beam interaction could compromise the LHC availability if they provoke quenches of superconducting magnets. In order to investigate the quench limits for this loss mechanism, a quench test using a wire scanner has been performed and is described in [16]. Moreover, the energy deposited by the incident particles on the wire might be sufficient to damage the wire. The wire can also be destroyed by the energy transferred by the beam to the wire through its accompanying electromagnetic field. According to [17], a travelling speed of 20 m/s should be sufficient to safely scan particle bunches in the LCH at collision energy ($7TeV$).

2.4 Conceptual design solution

The main idea of this design proposal is to avoid to put any moving parts outside the vacuum chamber. This decision rules out all the devices including sliding contacts, like brushed motors or slip-ring connectors. This way, the use of bellows will be not necessary with benefits on the durability of the device as lower inertia and less mechanical vibrations.

Regarding the motor, a permanent magnets synchronous machine has been selected. This allows the installation of the rotor and the stator, respectively, inside and outside the vacuum, separated by a thin metallic barrier. The same applies for the angular position sensor: the moving parts will be mounted on the motor shaft, while their readout systems will be situated on the outside. The task of this sensor is to deliver an accurate measure of the angular displacement, both to evaluate the wire position while the beam is crossed, and to be fed back into the control loop to track the desired trajectory.

A schematic representation of the design is shown in *Fig. 2.5.*

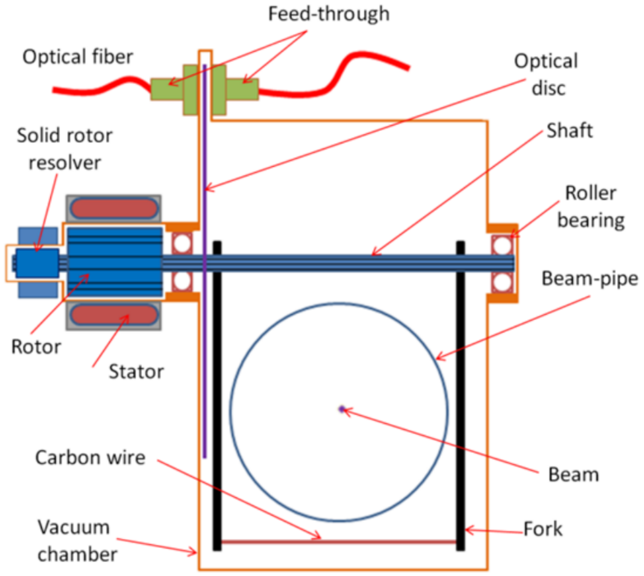


Figure 2.5 Simplified drawing of the future wire scanner. The orange lines represent the vacuum barrier.

2.5 Motion pattern design

When the beam wire scanner is activated, it scans the particle beam with a motion profile consisting of three phases:

- An accelerating phase, during which the desired speed is achieved with high torque in a short time period.
- The crossing phase, in which the particle beam collides on the wire generating secondary particles. This phase is the most sensitive to vibrations, since they can affect the accuracy of the profile evaluation.
- A decelerating phase, also achieved with high torque, to reduce the speed of the rotor to zero and stop it in the second rest position, with a π *rad* displacement.

The pattern is followed twice, in opposite directions, for each scan). This allows the system to return to the start position once the data are collected.

A crucial aspect about the motion pattern design concerns the vibrations in the wire induced by the strong acceleration. Usually, wire scanners use simple trapezoidal velocity models, in which the wire is accelerated until a constant

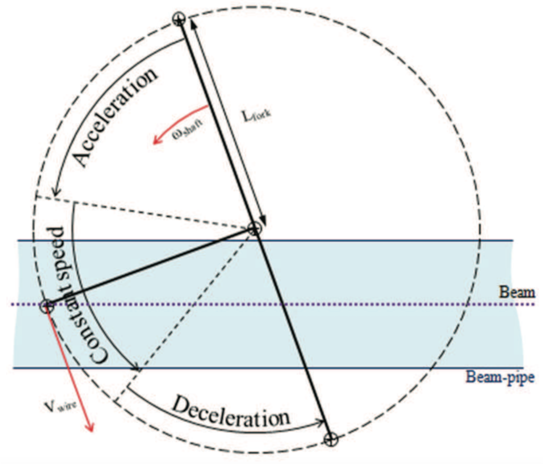


Figure 2.6 Representation of the motion profile during a scan

velocity by means of a constant acceleration and until zero velocity by means of a constant deceleration, as in *Fig. 2.6*. Such a profile implies an impulsive variation of the jerk, which would induce strong excitations into the mechanics of the system, especially into the wire.

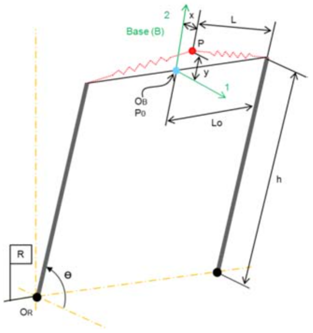


Figure 2.7 Schematic drawing for the dynamic model of the wire

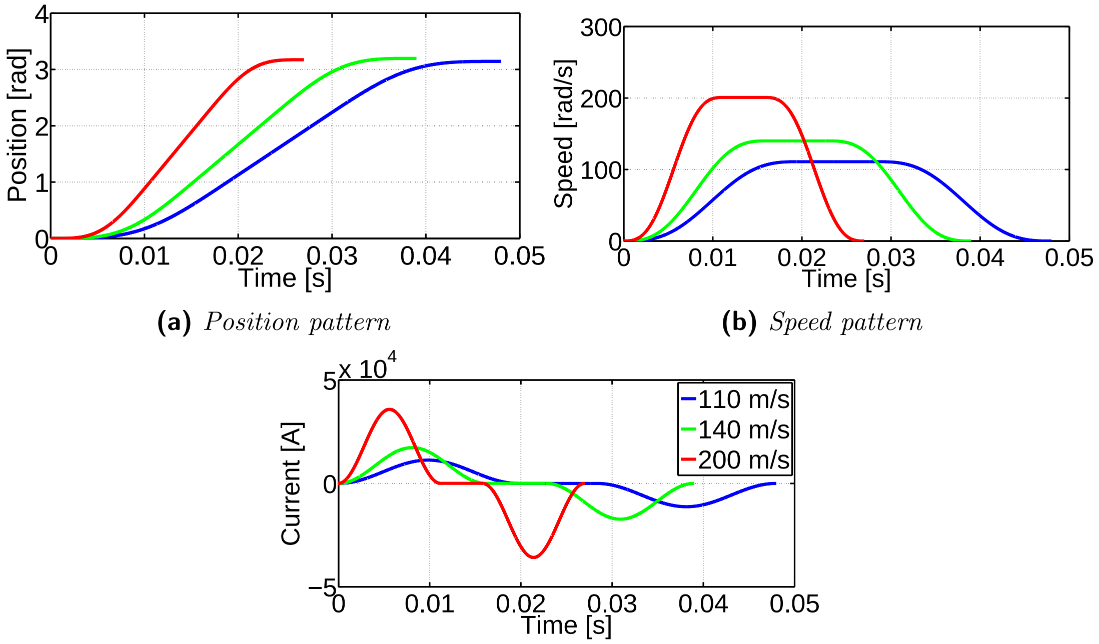
To study this phenomenon, a model of the wire was developed in [18], taking into account the contribution on the tangential and radial displacement of the middle point of the wire and an elastic behaviour toward deformations. The displacement equations for the middle point of the wire can be written as:

$$\ddot{x} = \ddot{\theta}h + \ddot{\theta}y + 2\dot{\theta}\dot{y} + \dot{\theta}^2 x - \frac{kx}{m} \tag{2.7}$$

$$\ddot{y} = -\ddot{\theta}x + -2\dot{\theta}\dot{x} + 2\dot{\theta}^2 h + \dot{\theta}^2 y - \frac{ky}{m} \tag{2.8}$$

where h is the fork length and m is the mass of the wire.

Relying on this model, a scanning profile was generated in order to minimise the vibrations in the wire. The patterns adopted within this thesis come from a trigonometric profile, which is optimal according to the simulations. They are shown in *Fig. 2.8* for different crossing speeds. An iterative optimisation method is ongoing and it is part of the future improvements foreseen for the project.



(a) *Position pattern*

(b) *Speed pattern*

(c) *Acceleration pattern*

Figure 2.8 Trigonometric profiles used in the new design

3. Hardware description and modelling

In this chapter, a detailed description of the design will be given, including all the main hardware components. This overview will consider the mechanical elements, mainly the motor itself, the sensing hardware such as the resolver and the current sensors, as well as the purely electrical devices, represented by the power supply system and the power transmission cable. We will return to the illustration of the regulator, under the hardware and software aspect, in a Chapter 4*. Thus, this part of the work will not be shown here. The choices that were made during the design phase will be motivated for all the selected elements, explaining their working principle, their sizing and their interfaces. Finally, particular care will be taken in the evaluation and computation of their effects on the control: the whole system will be represented as a block diagram, where each component will designate a block consisting in a set of inputs, a set of outputs, a set of internal states, and a transfer function, describing the laws that link these three sets and the update of the state variables.*

3.1 System overview

The beam wire scanner is a complex electromechanical system consisting in several different parts.

In order to design a motion controller for such a complicated device, it is necessary to achieve a global understanding of its subcomponents.

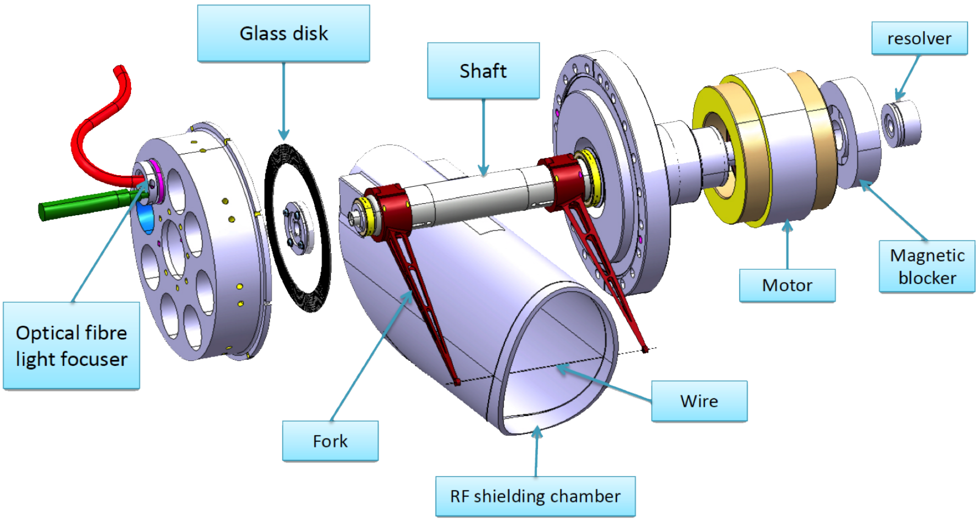


Figure 3.1 3D exploded view of the Beam Wire Scanner

In the next pages, the following components will be analysed:

- Permanent Magnet Synchronous Motor
- Resolver
- Current sensors
- Forks
- Brake
- Power supply system
- Power transmission cable
- Sine filter

3.2 Permanent Magnet Synchronous Motor (PMSM)

The most important and complex device that will be considered is the motor.

To provide the torque needed to drive the wire across the particle beam, a Permanent Magnet Synchronous Motor (PMSM) has been selected. This specific type of motor consists in a solid rotor, on which a set of permanent magnets are fixed to generate a constant magnetic field, and a stator made out of a ferromagnetic material, where the windings connected to the power line are installed.

By changing the currents into the windings, the flux generated by the stator will change in orientation and magnitude. The magnets mounted on the internal rotor will tend to align the two fluxes, resulting in a torque that can be controlled accordingly to the desired motion profile.

3.2.1 Working principle of the PMSM

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. The rotor is attracted and rotates synchronously with the stator rotating magnetic field. The stator windings are distributed in the stator slots, so that the voltage induced by the magnet during rotor rotation (back-EMF) is almost sinusoidal. This differentiates the PMSM from the commonly called Brushless DC Motor (BLDC), in which the back-EMF is trapezoidal [19].

The power density of a PMSM is higher than in other motors with the same rating due to a high magnetic field stored in the permanent magnets. It is more powerful and has both a lower mass and a lower moment of inertia. Due to its high power density and smaller size, the PMSM has evolved in recent years as the preferred solution for high dynamics speed and position control drives.

When we speak about brushless motors, it is always important to distinguish between *mechanical* and *electrical* quantities. Mechanical position and speed represent the absolute values of the shaft displacement and its derivative. Mechanical position, being a rotational quantity, is periodic with a period of 2π. On the other hand, electrical position and speed are referred to the motor frame from the electrical point of view. In other words, they are periodic according to a rotation towards the phase pattern, and their period depends on the number of winding pairs per phase (generally referred to as p). If a motor presents just one pole pair per phase, then the two quantities are equivalent.

Otherwise, the following relationship holds:

$$\theta_e = p\theta_m \tag{3.1a}$$

$$\omega_e = p\omega_m \tag{3.1b}$$

Figure 3.2 Picture of a permanent magnet synchronous motor and its components

Like for the stator of an induction motor, the control is realized by changing the currents into the stator windings.

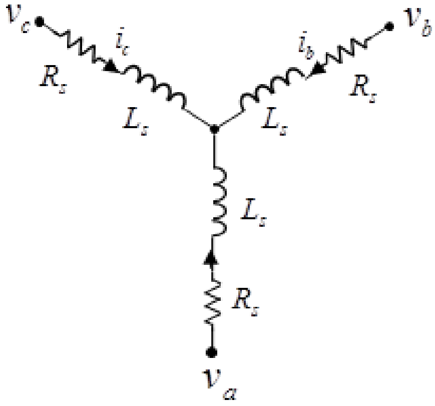


Figure 3.3 Equivalent circuit of the electric motor

Since the electrical equivalent of the motor can be modelled as in *Fig. 3.3*, the flux in a certain phase Φ_s will then vary proportionally to the currents according to:

$$\Phi_s = L_s i_s \tag{3.2}$$

Exploiting this relationship, and supposing the inductance L_s to be the same for the 3 phases, it will be possible to express the generated torque, proportional to the flux, as a vector combination of the 3 currents, scaled by a gain which is dependent on the geometry and the used materials.

Again, referring to *Fig. 3.3*,

$$u_A(t) = L_s \frac{d}{dt} i_A + R_s i_A(t) + u_{BE_A}(t) \tag{3.3a}$$

$$u_B(t) = L_s \frac{d}{dt} i_B + R_s i_B(t) + u_{BE_B}(t) \tag{3.3b}$$

$$u_C(t) = L_s \frac{d}{dt} i_C + R_s i_C(t) + u_{BE_C}(t) \tag{3.3c}$$

Where $u_{BE_x}(t)$ represents the back-EMF generated on winding x. The currents can be computed accordingly, in the frequency domain, as:

$$I_A(s) = \frac{1/R_s}{L_s/R_s s + 1}(U_A(s) - U_{BE_A}(s)) \tag{3.4a}$$

$$I_B(s) = \frac{1/R_s}{L_s/R_s s + 1}(U_B(s) - U_{BE_B}(s)) \tag{3.4b}$$

$$I_C(s) = \frac{1/R_s}{L_s/R_s s + 1}(U_C(s) - U_{BE_C}(s)) \tag{3.4c}$$

The electromagnetic torque contribution of one of the phases is:

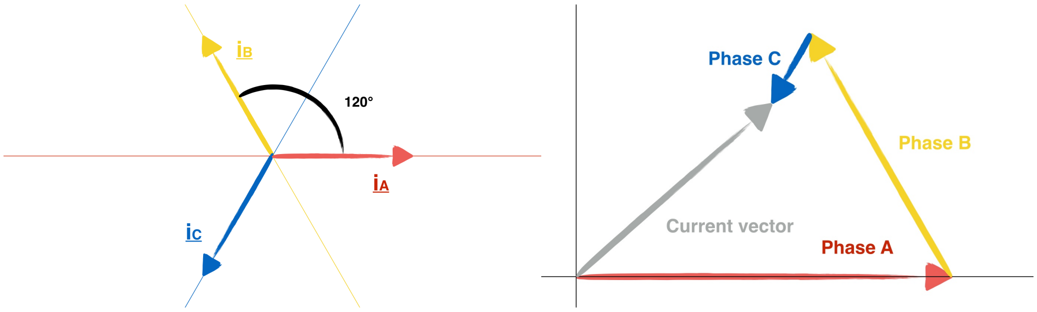
$$T = p\sqrt{3}K_E I cos(\theta) = pK_T I cos(\theta) \tag{3.5}$$

where p is the number of pole pairs, K_E and K_T are, respectively, the voltage and the torque constants of the motor, and ψ is the angle between the phase axis and the orientation of the rotor.

Now, we need to compute the full contribution of the 3 currents. This can be achieved by considering the vector sum of the currents, like in *Fig. 3.4*. Doing the maths, *(3.5)* becomes:

$$T = -K_T \left(sin(\theta_e) i_A(t) + sin\left(\theta_e - \frac{2\pi}{3}\right) i_B(t) + sin\left(\theta_e + \frac{2\pi}{3}\right) i_C(t) \right) \tag{3.6}$$

where with θ_e, we referred to the electrical position of the rotor.



(a) *Stator currents* **(b)** *Vector sum*

Figure 3.4 Vectorial representation of the stator currents and their sum

3.2.2 Selected device

Due to the mechanical and environmental constraints, the choice of the device has been quite challenging.

After an accurate analysis [20], the selected device was the Parker K500150 frameless PMSM. An overview of its parameters is given in *Table 3.1*

As it can be noticed, all of the specifications are met except for the peak torque. This problem was due to an underestimation of the total inertia of the rotating part during the pre-design phase.

The consequence of this fact will be the impossibility to drive the system to the nominal speed of 200 rad/s during the scan period. This is especially important in regard to the experimental results shown later on: all of the test have been made with a peak speed of 140 rad/s, to keep the motor working with a feasible amount of torque.

Latest analysis have led to a complete redesign of the motor [24]. It will be included in the next prototyping.

3.2.3 Coordinate transformation

Logically, the electro-magnetic torque generated by the motor is a function of the flux generated by the three-phase currents injected into the windings, so $T(\Phi_A, \Phi_B, \Phi_C)$. For *(3.2)*, the flux generated by a winding is directly proportional to the corresponding current, so $T(i_A, i_B, i_C)$. What is less intuitive, and actually marks the difference between a DC motor and an AC motor, is the absence of a real contact between the rotor and the stator.

PMSM parameter	Value	WS requirement
Stator resistance R_S	$0.245\ \Omega$	
Stator resistance L_S	$1.365\ mH$	
Voltage constant K_E	$0.3911\ Vrad^{-1}s^{-1}$	
Motor constant K_T	$0.3904\ NmA^{-1}$	
Pole pairs p	4	
Maximum frequency f	$426\ Hz$	
DC bus voltage U_{dc}	$300\ V$	
Nominal current I_N	$18\ A$	
Peak current I_p	$53\ A$	
Rotor inertia J_{rot}	$3.4 \times 10^{-4} kg\ m^2$	
Max. mechanical speed Ω_m	$670\ rad\ s^{-1}$	$210\ rad^{-1}$
Nominal mechanical speed Ω_N	$355\ rad\ s^{-1}$	
Peak torque T_p	$19.5\ Nm$	$> 30\ Nm$
Nominal torque T_N	$7.92\ Nm$	$2.98\ Nm(RMS)$
Maximum temperature	250^0C	200^0C
Air-gap thickness e_0	$0.8\ mm$	$0.3\ MM$

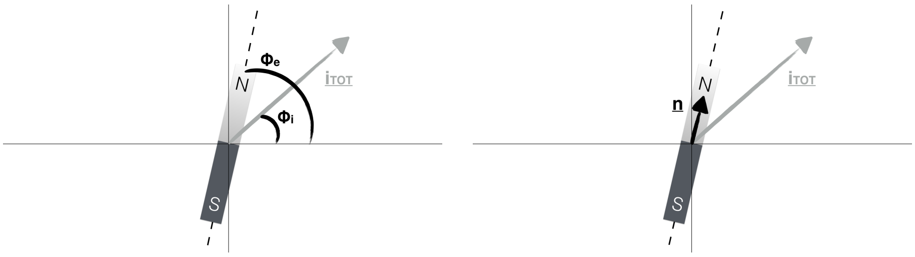
Table 3.1 Construction parameters of the selected PMSM

The coupling is purely magnetic, for no brushing contacts are included.

This implies that, since there are no different armatures between which the rotor could switch during the motion, the torque will also depend on the rotor's instantaneous position, $T(i_A, i_B, i_C, \theta_e)$. This concept is usually simplified by considering the vector sum of the currents: given the resulting current vector and the position of the shaft, we can write

$$T = K_T \; i_{tot} \; sin(\theta_i - \theta_e) \tag{3.7}$$

where the multiplication for $sin(\theta_i - \theta_e)$ can represent a vector product between $\underline{i_{tot}}$ and the position unit vector of the shaft, $\underline{n}$, as well as the equivalent projection of $\underline{i_{tot}}$ on a the plane normal to $\underline{n}$. In fact, it is immediate that when the flux is exactly aligned with the rotor magnets the torque is zero, and the only force generated is on the axial direction. However, when the angle between the shaft and the current (i.e. the flux) is 90^o the torque will be maximised.



(a) *Stator currents*

(b) *Vector sum*

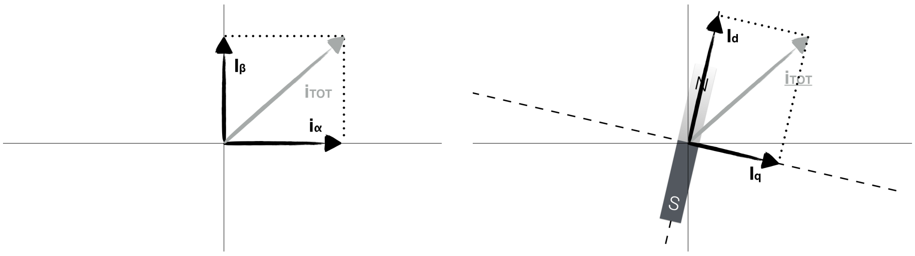
Figure 3.5 Interpretation of current-torque equation

As a vector product, *(3.7)* becomes:

$$\underline{T} = K_T \; \underline{i_{tot}} \times \underline{n} \tag{3.8}$$

where the vector $\underline{T}$ is the torque vector, interpreted with the right hand rule.

The latter equations demonstrate how the torque generated in the motor is proportional to the projection of the vector current on a plane normal to the rotor. Thus, the current can be projected on the axes identified by $\underline{n}$ dividing the $\underline{i_{tot}}$ in two components, as shown in *Fig. 3.6*: $\underline{i_d}$, aligned with the rotor magnet and $\underline{i_q}$, perpendicular to it.



(a) *Stator currents* **(b)** *Vector sum*

Figure 3.6 Representation of current after frame transformation

The transformations used to compute i_d and i_q from the three-phase currents are called *Clarke/Park transformations*, and are defined as:

$$\begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{3.9a}$$

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} c_\theta & s_\theta & 0 \\ -s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \\ i_0 \end{bmatrix} \tag{3.9b}$$

$$\begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} c_\theta & c_{\theta-\frac{2\pi}{3}} & c_{\theta+\frac{2\pi}{3}} \\ -s_\theta & -s_{\theta-\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \tag{3.9c}$$

or, in compact form:

$$\underline{i_{\alpha\beta 0}} = K_C \ \underline{i_{abc}} \tag{3.10a}$$

$$\underline{i_{dq0}} = K_P \ \underline{i_{\alpha\beta 0}} \tag{3.10b}$$

$$\underline{i_{dq0}} = K_{C/P} \ \underline{i_{abc}} \tag{3.10c}$$

here, K_C represent the Clarke transformation, and corresponds to the calculation of the components of $\underline{i_{tot}}$ with respect to the original frame while K_P, the Park transformation, is the projection of these components on a frame identified by $\underline{n}$ (*Fig. 3.6*). Finally, $K_{C/P} = K_P \ K_C$.

The described transformations are extremely useful in the field of motor control because they turn a set of 3 currents, which effect is hard to evaluate, into 2 terms that describe exactly their practical effects. i_q, normal to the field generated by the stator, is named *quadrature current* and is proportional to the torque. On the other hand, i_d is parallel to this field and its contribution to the torque is zero. Its only effect is to "push" the rotor against the stator, increasing the friction. This is why we want i_d to be minimised, possibly zero.

Finally, it is possible to write:

$$T(i_d, i_q) = \frac{3}{2}p(\Phi_f i_q + (L_d - L_q)i_d i_q) \tag{3.11}$$

where Φ_f is the flux generated by the stator, and L_d and L_q are the equivalent inductances relative to the d and q axes. For surface PM motors whose $L_d = L_q$, like in our case, the above equation simplifies into:

$$T(i_d, i_q) = \frac{3}{2}p\Phi_f i_q = K_T i_q \tag{3.12}$$

3.2.4 Transfer function

To describe a model for our PMSM, we first need to define the inputs and outputs of the system: the input set will be composed of u_d and u_q, that is, the $d-q$ transformation of the three-phase voltages, while the output set will contain the currents i_d and i_q, the angular speed ω_m and the angular position θ_m. The internal states were chosen to correspond to the outputs.

Recalling (3.3), transforming with (3.10) and extending the value of $u_{BE_x}(t)$, we obtain:

$$u_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_e(L_q i_q + \Phi_f) \tag{3.13}$$

$$u_q = R_s i_q + L_q \frac{di_q}{dt} + \omega_e(L_d i_d) \tag{3.14}$$

from which it is possible to extract the differential equation describing the behaviour of the current, in the Laplace domain:

$$\frac{d}{dt}i_d = \frac{1}{L_d}u_d - \frac{R_s}{L_d}i_d + \frac{1}{L_d}\omega_e(L_q i_q) \tag{3.15}$$

$$\frac{d}{dt}i_q = \frac{1}{L_q}u_q - \frac{R_s}{L_d}i_q - \frac{1}{L_q}\omega_e(L_d i_d + \Phi_f) \tag{3.16}$$

Finally, we can write down the last two components of the transfer function, relative to the speed and position. Recalling (3.12)

$$\frac{d}{dt}\omega_m = \frac{(K_T\ i_q) - \omega_m B}{J} \tag{3.17}$$

$$\frac{d}{dt}\theta_m = \omega_m \tag{3.18}$$

where B is the damping factor. Its value will be investigated further in these pages.

Together, (3.15), (3.16), (3.17) and (3.18) entirely describe the behaviour of a general permanent magnet synchronous motor.

3.3 Position and speed measurement

The absolute angular position of the shaft is provided by a resolver consisting in a solid stator, in which the excitation and the acquisition windings are installed, and a rotoric disk made out of a nonhomogeneous material. The resolver has the advantage of delivering an *absolute* measure: this means that, the moment the system is turned on, a value for the angular displacement is immediately available. This guarantees better robustness for the control point of view, and for this reason the measure coming from the resolver is used for the control feedback loop.

3.3.1 Working principle of the resolver

A resolver is a rotary transformer where the magnitude of the energy through the sensor windings varies sinusoidally as the shaft rotates. A resolver control transmitter has one primary winding, the *reference* winding, and two secondary windings, the *sin* and *cos* windings. The reference winding is located in the rotor of the resolver, the sin and cos windings in the stator. The sin and cos windings are mechanically displaced 90 degrees from each other, like in *Fig. 3.7.*

In general, in a control transmitter, the reference winding is excited by an AC voltage called the reference voltage u_r.

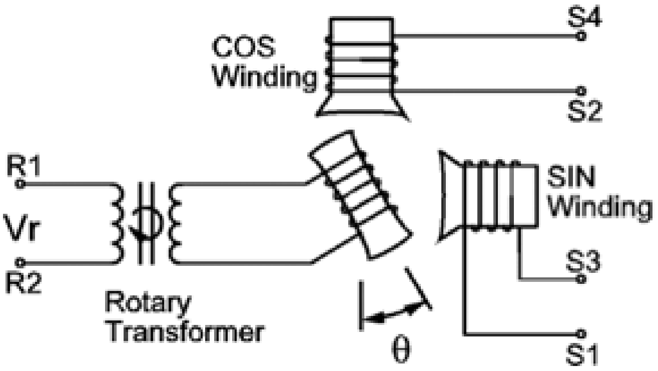


Figure 3.7 Brushless resolver control transmitter schematics

The induced voltages in the sin and cos windings are equal to the value of the reference voltage multiplied by the sin or cos of the angle of the input shaft from a fixed zero point, so:

$$u_s = K_{res} u_r sin(\theta) \tag{3.19a}$$

$$u_c = K_{res} u_r cos(\theta) \tag{3.19b}$$

where u_r is the excitation voltage and K_{res} is the transformation ratio, a constant depending on the magnetic coupling. Thus, the resolver provides two voltages whose ratio represents the *tan* of the absolute position of the input shaft.

Since

$$\frac{u_s}{u_c} = tan(\theta) \tag{3.20}$$

Then

$$\theta = atan_2(u_s, u_c) \tag{3.21}$$

In a brushless resolver, energy is supplied to the reference winding (i.e. the rotor) through a rotary transformer. This eliminates brushes and slip rings in the resolver and the reliability problems associated with them.

3.3.2 Selected device

The Rotasyn (*Fig. 3.8*), unlike the traditional brushless resolver, has both primary and secondary windings in the stator and thus no rotary transformer is required. The transferred energy remains magnetic from the primary coil through the air gap to the sinusoidally shaped poles of the solid rotor. The rotor works as a magnetic valve completing the flux path. Since the total flux through the gap is constant, the rotor determines the angular position within the stator bore where the coupling occurs, and thus the relative amplitudes of the output signals.

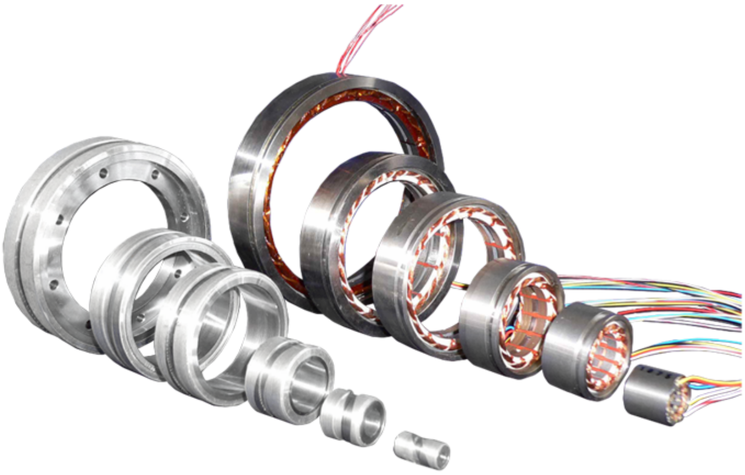


Figure 3.8 Family of *Rotasyn* resolvers

This design gives the Rotasyn unique advantages over traditional brushless resolvers: since the solid rotor has no windings, it is easy to make it vacuum compatible and resistive to radiation and temperature. As for the motor, the stator of the Rotasyn can be placed outside the vacuum chamber and the air-vacuum barrier is obtained by means of a low magnetic permeability thin wall located in the air-gap.

As we said before, the choice of a resolver instead of an incremental sensor like an encoder for the control loop is due to the fact that, during the design phase, the robustness granted by having an absolute measure for the full time of scan was considered more important than, for example, having a more accurate measure or to be able to acquire it in open-loop.

The parameters for the chosen device are listed in *Table 3.2.* They all fit the required specifications except for the accuracy: ±60 $arc-minutes$ corresponding to 17.45 $mRad$. Even though the accuracy does not match the requirements specified in [27], it can be largely improved by calibrations, allowed by the high reproducibility of the measure.

Resolver parameter	Value
Mass	$85\ g$
Maximum speed	$50\ kRPM$
Moment of inertia	$20\ g\ cm^2$
Accuracy	$\pm 60\ arc - minutes$
Radial air gap	$0.3\ mm$
Excitation frequency	$5 \div 50\ kHz, 10\ kHz\ nominal$
Excitation amplitude	$2 \div 12\ Vrms$
Transformation ratio	1.75×10^{-4}

Table 3.2 Construction parameters of the Rotasyn resolver

3.3.3 Resolver-to-digital conversion

One of the major contributions to the accuracy of a resolver is the *resolver-to-digital conversion* (RDC).

Differently from incremental sensor, indeed, it is much more complicated to convert a signal generated by a resolver into digital. The principle of the conversion is the application of (3.21) on the two input voltages, measured on the differential pins of the sin and cos windings. Generally, this is how, the conversion in open-loop can be implemented. Anyway, the most common way to perform the digitisation is represented by the implementation of a tracking feedback loop based on oversampling.

In order to perform this delicate operation on the BWS position feedback sensor, the *AD2S1210* chip was chosen. After filtering the two input signals properly, an oversampled estimated value for θ is computed by the *AD2S1210* by means of a feedback loop. This value is then compared to the current output angle and this error is regulated to zero by the type II tracking loop shown in *Fig. 3.9*, where the represented parameters are constants depending on the desired resolution.

The disadvantage of such a method is that, differently from a simple open-loop acquisition, the tracking loop itself has a dynamic. This means that the resolver cannot be modelled as a simple wire in our block diagram, but it will own a proper *transfer function*, that can be faster or slower with respect to the number of bits adopted for its representation.

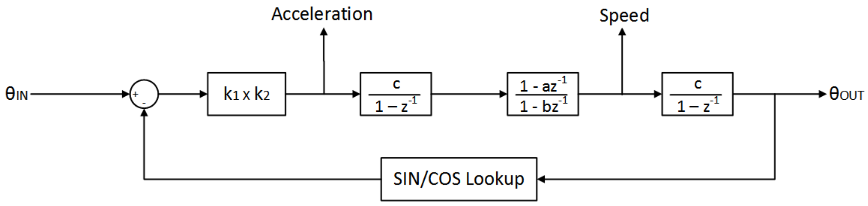


Figure 3.9 Tracking loop for resolver-to-digital conversion

As it is reasonable to expect, the higher the resolution, the slower the dynamic will be, resulting eventually in unacceptable delays in the control. The used RDC allows to select the desired resolution within a range of 10 to 16 bits.

The rise time of the resolver digitiser has been one of the criteria used to choose the resolution for the system: a comparative of the step responses corresponding to the various resolutions allowed by the board [21] is shown in *Fig. 3.10.*

If we consider a top speed of 200 rad/s, we can compute:

$$200\frac{rad}{s} \approx \frac{11460°}{s} = \frac{11.46°}{ms} \tag{3.22}$$

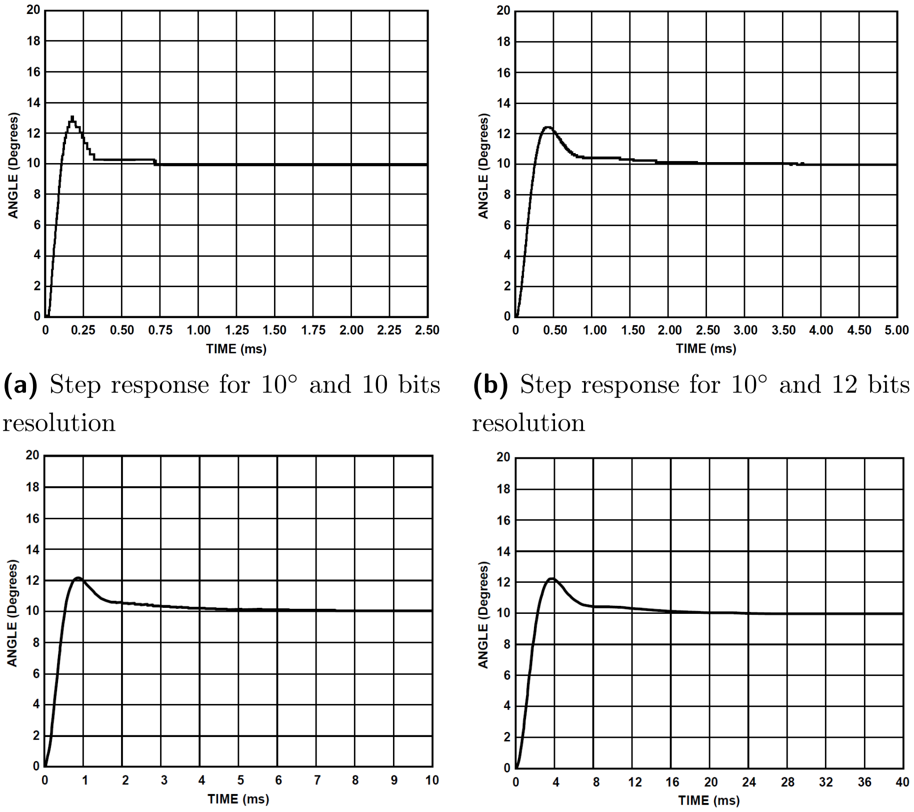
thus, the rise time should not be higher than 1 ms for a step of 10°. The latter observation excludes the top resolution, 16 bits, from the set of the feasible ones due to the results shown in *Fig. 3.10.*

An interesting work was developed in [22] on the acquisition system from the resolver and its implementation on a FPGA board. Unluckily, the open-loop architecture of this design couldn't provide enough accuracy for the position and speed feedback. A study on the possibility to include the RDC action in the motion controller design will be object of future work.

3.3.4 Speed measurement

Another, and probably the most important drawback of the choice of a relatively low-resolution position sensor, concerns the evaluation of the speed.

If the position resolution can satisfy the need of a high precision motion application, this is not always true for the speed measurement.



(a) Step response for 10° and 10 bits resolution

(b) Step response for 10° and 12 bits resolution

(c) Step response for 10° and 14 bits resolution

(d) Step response for 10° and 16 bits resolution

Figure 3.10 Step response of the RDC for different resolutions

In fact, the first can be written as:

$$res_\theta = \frac{2\pi}{2^{res_{bit}}} \tag{3.23}$$

where res_{bit} is the resolution expressed in bits, while the second is equal to the minimum angle step (i.e. res_θ) over the time interval running between two samples:

$$res_\omega = \frac{res_\theta}{\Delta t} \tag{3.24}$$

but $\Delta t = 1/f$ so:

$$res_\omega = res_\theta f \tag{3.25}$$

for now, let's assume $f = 16\ kHz$. The reasons for this choice will be clarified in *section 3.6*.

In *Table 3.3* the different resolutions in position and speed are listed as a function of the resolution in bits.

res_{bit}	$res_\theta\ [rad]$	$res_\omega\ [rad/s]$
10	0.006136	98.175
12	0.001534	24.544
14	0.000383	6.136
16	0.000096	1.534

Table 3.3 Resolution of the resolver as a function of the used bits

The observation of the table clarifies how an acceptable loss of resolution in position can translate in a totally unacceptable loss in speed. As an example, assuming a borderline accuracy of 10 bits, still sufficient to perform a decent position control, the speed resolution will be 98.175 rad/s. This means that, if the motor spins at a constant speed of 50 rad/s, the error in its measure will be always nearly 100%, entirely unacceptable for a high-performance drive like the Beam Wire Scanner.

Due to the latter observations, the resolution of the resolver was set to 14 bits, which is the best crossover between accuracy and dynamic response.

3.3.5 Transfer function

In the case of the position sensor, the input and output variables are clearly represented by the actual value of the angular displacement and its measured value. The input is a real-world variable, which is continuous both on the time domain and in magnitude, while the output is a digital one, discrete under both these aspects. The resolver is an intrinsically continuous sensor (gives voltages as outputs), but, since the digitisation of the signal is an essential part of its use, it will be included in the description of its dynamic behaviour.

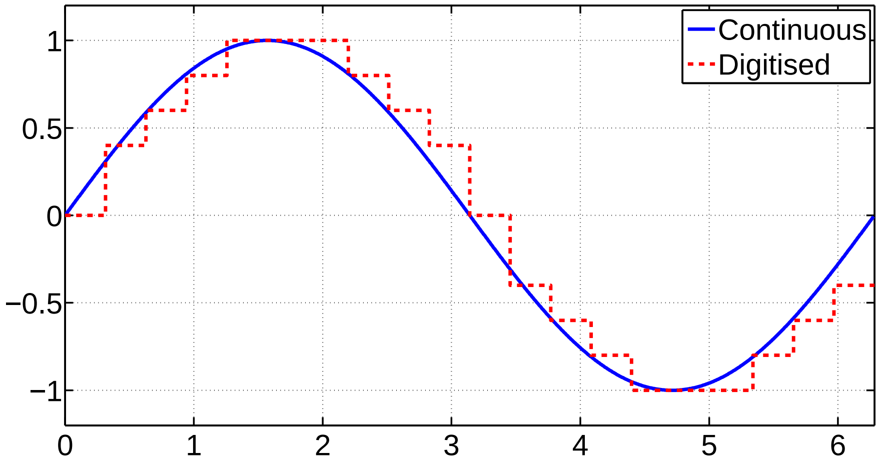


Figure 3.11 A continuous function and its digitisation

The discretisation on the time axes is due to the *sampling*, and it has been modelled like a simple sample-and-hold module. It is also called ZOH (*Zero-Order Hold*), and it basically perform the variable transformation form the continuous Laplace domain to its discrete equivalent, the Z domain. On the other hand, the discretisation on the vertical axes (*quantisation*) is an effect of resolution; this concept was widely discussed within this chapter and modelled with a simple quantisation block.

Apart from the digitalisation problem, the actual dynamic of the resolver can be computed from *Fig. 3.9*.

The equation for the open loop system resolver can be written (neglecting higher-order dynamics) as:

$$G_{r_{ol}}(s) \approx \frac{K_a}{s^2}\frac{1+st_1}{1+st_2} \tag{3.26}$$

where

$$\begin{aligned} t_1 &= \frac{t_s(1+a)}{2(1-a)} \\ t_2 &= \frac{t_s(1+b)}{2(1-b)} \\ K_a &= \frac{k_1k_2(1-a)}{a-b} \end{aligned}$$

where t_s indicates the sampling period, and a, b, k_1 and k_2 are constants depending on the selected resolution. Thus, the closed-loop *tf* of the resolver is simply:

$$G_{r_{cl}}(s) = \frac{G_{r_{ol}}(s)}{1+G_{r_{ol}(s)}} \tag{3.28}$$

since the sin/cos lookup has unity gain.

3.4 Current measurement

The second important sensor in the BWS actuator system is represented by the current sensing devices.

The current regulation is indeed the basic principle of motion control for electric drives, since in the majority of them , especially the asynchronous ones, the torque generated by the stator windings is directly proportional to the vector sum of the currents flowing into them. Since the stator windings are closed on a star connection, it is always possible to compute the third current even using just two sensors, like:

$$i_C = -(i_A + i_B) \tag{3.29}$$

due to Kirchhoff's law.

Still, it was chosen to use three sensors, one for each phase, to get the possibility to exploit this redundancy in case of a hardware failure in one of them or in its acquisition system.

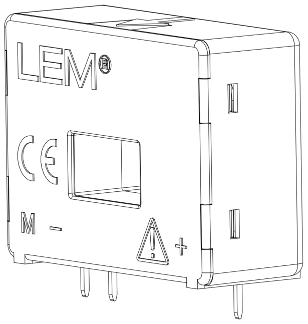


Figure 3.12 LEM LA 100-P current sensor

3.4.1 Chosen device

The selected device was the Current Transducer LA 100-P by *LEM*. It is basically a compensated transimpedance amplifier with a tunable gain, which can be selected by means of a resistor. Its parameters are listed in *Table 3.4*.

Current sensor parameter	Value
Measuring range	$\pm150\ A$
Conversion ratio (internal gain)	$1 : 2000$
Accuracy	$\pm0.45\%$
Max offset current (room temperature)	$\pm0.10\ mA$
Rising time t_r	$< 1\mu s$

Table 3.4 Parameters of the LEM LA 100-P

3.4.2 Transfer function

Due to compensation, the current sensor needs to be modelled like something more complicated than a simple wire, as well as the resolver. In this case, the dynamic will be fully continuous, since no digitisation is included in the sensor itself. The discretisation effect will of course be added later, once the sampling hardware is chosen.

The transfer function of the LEM LA 100-P can be easily approximated with a first-order low pass filter dynamics considering *Table 3.4* and recalling the usual rule-of-thumb linking the rising time of a system and its cut-off frequency (i.e. its bandwidth):

$$f_c = \frac{0.35}{\tau_r}$$

considering the worst-case performance, thus assuming $\tau_r = 1\ \mu s$, it yields

$$f_c = \frac{0.35}{10^{-6} s} = 350 kHz \tag{3.30}$$

so, the time constant of the filter can be written as:

$$\tau_c = \frac{1}{2\pi f_c} = 4.55 \times 10^{-7} s \tag{3.31}$$

which corresponds to the following transfer function

$$G_{lem}(s) = \frac{2.2 \times 10^6}{s + 2.2 \times 10^6}$$

3.5 Contribution of the mechanical design

Apart from the rotor and the shaft, there are several other contributions to the global dynamics of the system. The mechanical contributions to the dynamic of the system are mostly nonlinear and they represent the main complications to the modelling. Their effect is computed on-line by a real-time algorithm and compensated with a feedforward controller.

3.5.1 Inertia

First of all, we shall consider the various contribution to the final inertia. Recalling the complete mechanical design of the beam wire scanner (*Fig. 3.1*), the total inertia was computed in [23] and summarised in *Table 3.5*.

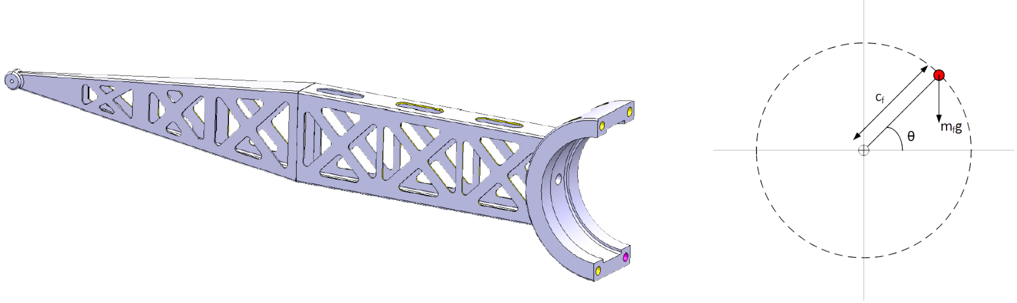
3.5.2 Forks

The forks installed on the beam wire scanner were designed in [23]. They were designed to be as light as possible, to withstand a high quantity of torque and to minimise the vibrations on the tip.

Component	$J\ [kg\ m^2]$
Bearing 1	1.96×10^{-5}
Bearing 2	2.44×10^{-6}
Disc ($t = 3mm$)	2.51×10^{-4}
Disc holder	2.68×10^{-4}
2 Forks	1.75×10^{-4}
Resolver	3.99×10^{-6}
Rotor	3.40×10^{-4}
Shaft ($t = 5mm$)	2.91×10^{-4}
Total	1.35×10^{-3}

Table 3.5 Contribution of the various elements to the total inertia of the system

A lateral view of the forks is shown in *Fig. 3.13 (a)*. Since they are still in a prototyping phase, it was chosen to 3D-print them in a titanium alloy.



(a) *Lateral view of a BWS fork*

(b) *Torque contribution of the forks*

Figure 3.13 Forks mechanical design and torque effect

The impact of the forks on the system dynamics are two: the inertial contribution, already considered in *Table 3.5*, and the gravitational effect, only present in the vertical scanners. The forks produce a torque on the shaft depending on their position: modelling as in *Fig. 3.13 (b)*, we can write:

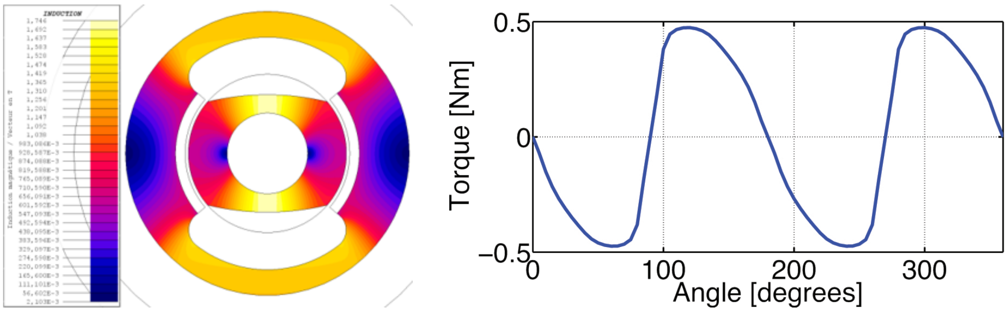
$$T_f = 2\ c_f m_f g\ cos(\theta) \tag{3.32}$$

where $m_f = 0.066\ kg$ is the mass of a fork and $c_f = 17.5\ mm$ is the distance of the centre of mass of the fork from the shaft axis.

3.5.3 Magnetic brake

To prevent the actuator from stopping in an unknown position in case of failure, a magnetic brake was included in the design [24]. The brake consists in a stator-rotor system where two permanent magnets are installed: when the shaft is moved from its 'home' position, the two magnets will tend to align to one other until a stable position is reached again. The two stable positions correspond to the start and the end of the scanning routine.

Fig. 3.14 (a) shows a schematic realisation of the brake, while in *Fig. 3.14 (b)* its torque contribution to the system is plotted as a function of the position.

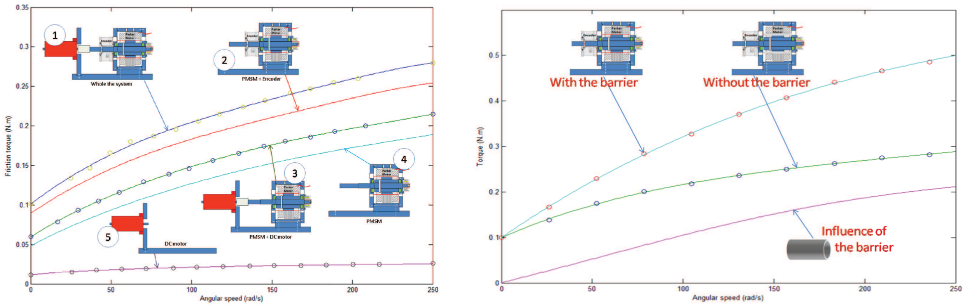


(a) *Schematic representation of the brake* **(b)** *Torque produced by the brake vs angular position*

Figure 3.14 Brake schematic and effects

3.5.4 Friction, damping and vacuum barrier effects

The impact of dynamic friction, damping and the effect of the vacuum barrier on the electromagnetic field are all attenuations that should be taken into account when modelling the PMSM. They were widely investigated, both theoretically and experimentally, in [20]. The experimental testbench was set up connecting a DC motor on the same shaft as the PMSM, running it at constant speed and computing the necessary amount of torque. The results are shown in *Fig. 3.15*, and they will be used in all of the following chapters.



(a) *Identification of the friction and dumping constants*

(b) *Qualification of the vacuum barrier effects*

Figure 3.15 Experimental qualification of the friction, damping and barrier effects

3.6 Power supply

One of the main points of the design for the new beam wire scanner is the full separation between the electronics, the power supply and the control board - which are going to be installed in the surface - and the actual motor - which will be installed, of course, in the accelerator chamber. Since the distance between these two parts could be relevant (up to 300 m), a special care has been taken to select the type of driver for the motor. The power supply design was more complicated than just selecting the right kind of amplifier to drive the PMSM: it consists in a complete electronic system including:

- The actual generator
- The power amplifier and its interface
- Voltage and current sensors and their power supply
- A charge storage stage

aAll of the previously listed devices are installed in a single box which constitutes the entire power electronic compartment of the system.

A preliminary study has been conducted in collaboration with the Heig-vd school to validate such architecture in the case of long cabling [25].

3.6.1 Overview

The voltage generator is a *Delta Elektronika* 150 SX 200-200. The characteristics are listed in *Table 3.6*.

Power supply parameter	Value
Voltage range	$70 - 400\ V$
Max current	$0.3\ A$
Efficiency	84%
Ripple + noise (max)	$100\ mV$ p-p
Output impedance (max)	$1\ \Omega$

Table 3.6 Characteristics of the voltage generator

For the voltage amplification, a Pulse Width Modulation (PWM) three-phase driver has been chosen. It represents a typical solution for motor drivers since the motor, outlined by the circuit in *Fig. 3.3*, behaves like a low-pass filter itself, cutting off the higher frequencies contained in the PWM signal. On top of this effect, the mechanical dynamics are much slower than the electric ones, for obvious physical reasons, resulting in an additional low-pass action.

The amplification is performed by means of a 3-channel inverter realised with six IGBTs. The IGBTs specifications can be seen in *Table 3.8*

Item	Maximum ratings
Collector-Emitter voltage	$600\ V$
Collector maximum current	$200\ A$
Collector-Emitter saturation voltage	$2.45\ V$
Input capacitance	$14.0\ nF$
Turn-on time	$1.20\ \mu s$
Turn-off time	$1.20\ \mu s$

Table 3.7 Parameters for the used IGBTs

The rack also includes a cooling fan, a driver circuit to regulate the switch timing of the power transistors and a charge storage stage, implemented by means of a set of capacitors, used to store the charge during the non-scanning phase, in

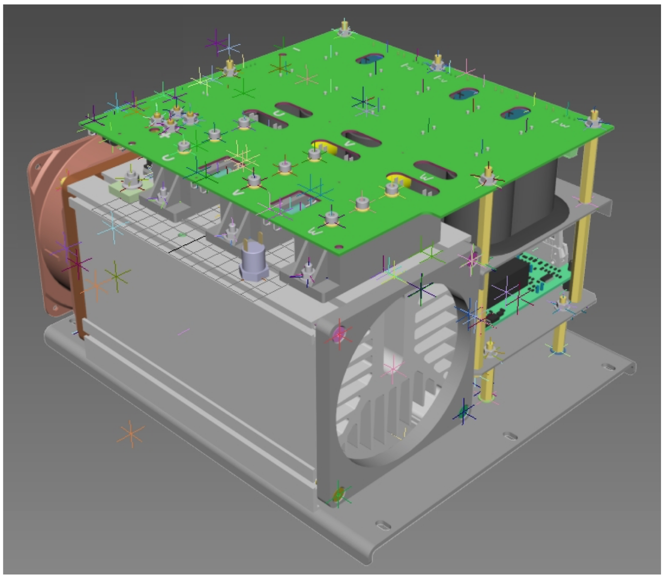


Figure 3.16 Power inverter board

order to avoid large voltage drops when the scan is performed and the current flows into the power lines.

The PWM carrier frequency was designed to be $f_{PWM} = 16\ kHz$, while the sampling period T_s was set equal to its inverse, to perform the measures always with the same phase and avoid to include switching noise in the current measures. The frequency was chosen due to a set constraints, including the maximum guaranteed working frequency for the switch ($f_{MAX} = 25\ kHz$), and the minimum frequency for the proper behaviour of the power switch ($f_{MIN} = 4\ kHz$). Furthermore, this value makes the digital implementation easier, being approximated to a multiple of 2. A switching frequency between 10 kHz and 20 kHz is a typical choice for motion controllers.

During the movement, the expected energy consumption will be higher than the capability of a standard main plug (230V, 10A). Capacitors on the DC bus should provide the instantaneous charges needed by the motor [26]. The dimensioning of this part takes in account the power needed to accelerate and the maximum voltage drop at the end of this phase.

$$Q = C_{bus}V \tag{3.33}$$

where C_{bus} is the total capacitance on the DC bus and $V = 300\ V$

If we consider an average current during the scan routine $\overline{I_s} = 32\ A$ and an acceleration duration of 16 ms we get:

$$Q_{scan} = \overline{I_s} t_{scan} = 32\ A \times 16 \times 10^{-3} s = 0.512\ C \tag{3.34}$$

and so, considering a voltage drop of 10% on the power line, the capacitors can be sized like:

$$C = \frac{\frac{100}{10} Q_{scan}}{V} \approx \frac{5}{300} = 16.66\ mF \tag{3.35}$$

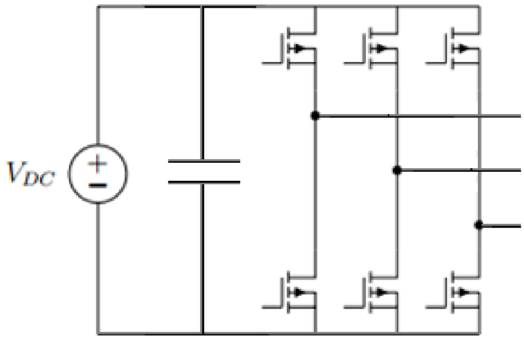


Figure 3.17 Schematic of the power line

Since the turn-on and the turn-off time are not zero, an additional problem could rise with the power transistors: if during the switch both the voltages on the gates are interpreted as "high", the two transistors will saturate and short-circuit the V_{CC} line to *gnd*. A solution to this problem is to include in the PWM generator the so-called *dead time* function, that is a short period of time during which both the transistors are in a low-state, allowing the safe *high-low* transition of one before the *low-high* transition of the other. This complication is explained in *Fig. 3.18*.

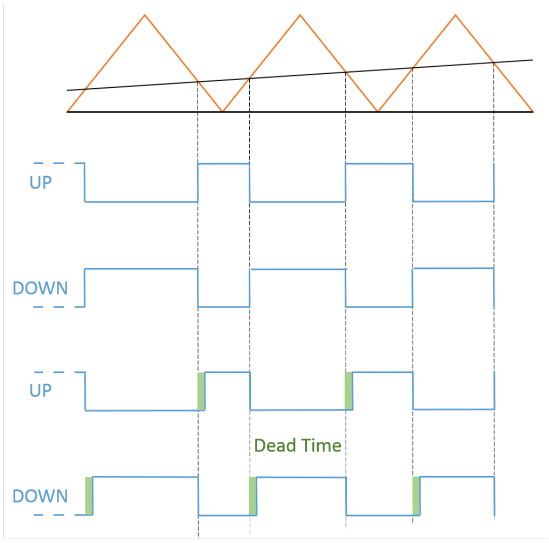


Figure 3.18 PWM generation with dead time

In the case of our inverter, the dead time delay was set to be twice the rising-falling time of the IGBTs:

$$T_d = 2T_{on/off} = 4.8\ \mu s \tag{3.36}$$

3.6.2 Transfer function

The transfer function of the power supply stage is basically represented by the inverter dynamic. Considering the PWM effects (introduction of high frequencies) compensated by the motor and by the power filter, the remaining dynamic of the inverter is basically represented by the delay introduced by the 3-phase bridge. This, accordingly to [28], can be approximated to $T_s/2$, being T_s the sampling period.

Recalling that the frequency-domain representation of a τ_d delay is $e^{-s\tau_d}$, the linearised version of this dynamic can be obtained using the Padè approximation. For our model, a second-order Padè approximation was performed:

$$e^{-s\tau_d} \approx \frac{1 - k_1 s + k_2 s^2}{1 + k_1 s + k_2 s^2} \tag{3.37}$$

with k_1, k_2 coefficients of Padè table.

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

Table 3.8 Coefficient for the Padè approximation of order 1 and 2

3.7 Power cable

The beam wire scanner is a very powerful and versatile instrument used for beam diagnostic. Because of the possibility to use it for different kinds of beam and at a different stage of acceleration, CERN adopted many sensors of this kind in its facility, as seen in *Table 2.1*. Moreover, the necessity of calibration for the other sensors, for which the BWS is used, justifies its installation in several positions within the accelerators structure.

Because of these reasons, the length of the cable separating the PMSM and its drive is variable between a couple of meters up to $300m$. As it is natural to expect, the specifications require the system to be entirely stable and to show similar performances for all the possible cable lengths. This task is particularly challenging to achieve, since power cables are not usually easy to model in all of their parameters, on which their dynamic behaviour depends.

This section will show the main steps of the characterisation of the three-phase power cable modelling, from the theoretical modelling approach to its experimental verification.

3.7.1 The power transmission problem

Power transmission lines are a common problem in the case of motors that are installed far away from their drivers. This kind of situation often happens in tough environments, such as mines and subaqueous applications, that make the placement of the driver in a closer position or the use or short cables especially hard.

Nevertheless, in the latter situations, the motors are usually part of fans to extract air or gas from a cavity, or drills, or propellers. This means that they are often requested to perform a 'simple' task from the regulator point of view, such as constant-speed turning, maybe with a tunable speed, also allowing vibrations and errors that are not acceptable in case of the precision position control implemented for the beam wire scanner. This means that, for this particular application, the complication of a long power cable and of a precise position control at high-speed will be both present.

The main effects of cables in power transmission are:

- **Attenuation** - Minly due to resistive effects into the conductors, increases linearly with the cable length and consists in a voltage drop between the *in* and *out* pins of the same phase. It stays in principle constant when the frequency changes.
- **Cross-talk** - This effect is provoked by the leakage currents between the insulator used to separate the conductors. Its contribution is greater at higher frequencies due to parasite capacities.
- **Ringing** - R inging behaviours occur when a cable is excited in its resonance frequencies. This is most likely to happen when a PWM driver is used, since many frequencies higher than the desired ones,are introduced by the modulation and can be uncontrollably amplified, possibly damaging shielding and insulation.

3.7.2 Power filter

The first typical countermeasure to the undesired dynamics introduced by cabling is the use of a *power filter*, that is, a low-pass filter capable to carry the current needed to drive the motor. A popular version of this kind of filters is the *sine filter*.

Sine filters are realised like the circuit in *Fig. 3.19*, and basically consist in a three-phase low-pass LC filter where the ground is substituted by a common floating node. This circuit cuts down the high frequencies that are due to modulation and noise, resulting in a smoother voltage fed on the motor windings.

During the switch, mostly high frequency noise is introduced into all three outputs at the same time. When a high frequency switch is turned on in the motor controller, a pulse of energy is driven into the ground plane inside the printed circuit board. The ground plane ideally would have the same potential everywhere, but due to real world limitations such as copper resistivity and plane inductance, the ground plane has slightly different voltages across its section. When the motor controller switches on a power component, the ground plane

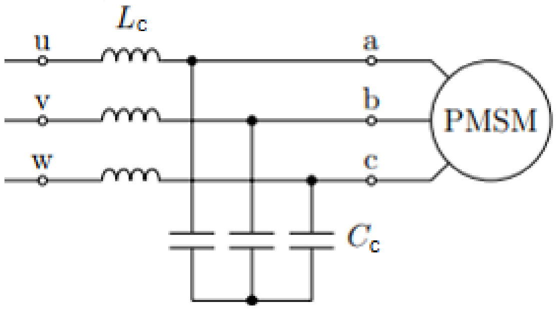
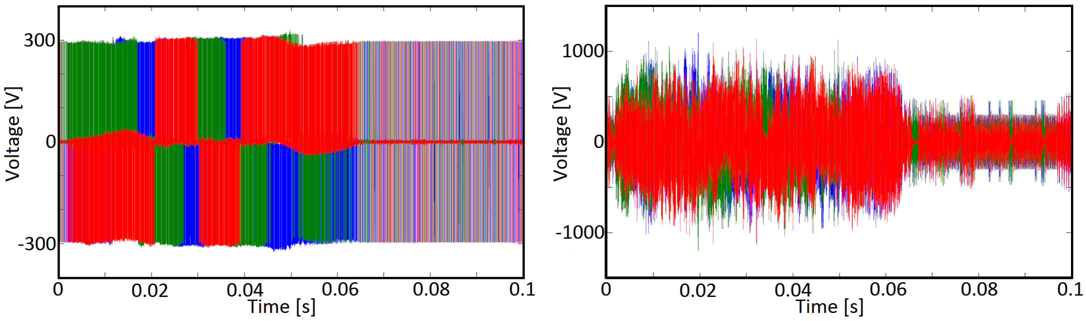


Figure 3.19 Schematic of a sine filter



(a) *Cable input*

(b) *Cable output*

Figure 3.20 Simulation of the line voltages with no filter and a 250m cable

voltage rises sharply. This produces ground bounce and it is a source of high frequency common mode noise. The sine wave filter squashes this high frequency common mode noise.

The chosen filter is a *Schaffner* FN5040-24-84 , shown in *Fig. 3.21*.

Figure 3.21 Sine filter by FN5040-24-84 *Schaffner*

3.7.3 Cable model

Two main complications occur when trying to include the model of the cable in the overall system scheme, especially considering the control aspect.

The first problem is a purely electric one. Normally, to extrapolate a transfer function from a circuit, the ratio between the output and the input pins is computed while these are considered floating, that is, the output load is hypothesised to be big enough such that no current is flowing into it. Then, the effects of each input on each output are usually calculated using the superposition principle. This hypothesis is standard and usually verified in the case of low-power and signal electronics. In the case of a PMSM, however, the input load is not big at all: its order of magnitude is the same as the winding resistances, which in our specific case are $R_s = 2.2\ \Omega$. This value is definitely not enough to suppose that no current is flowing into them. This fact is obvious considering that the motor is driven in current, and so it is more than reasonable to expect it to present a low resistive load.

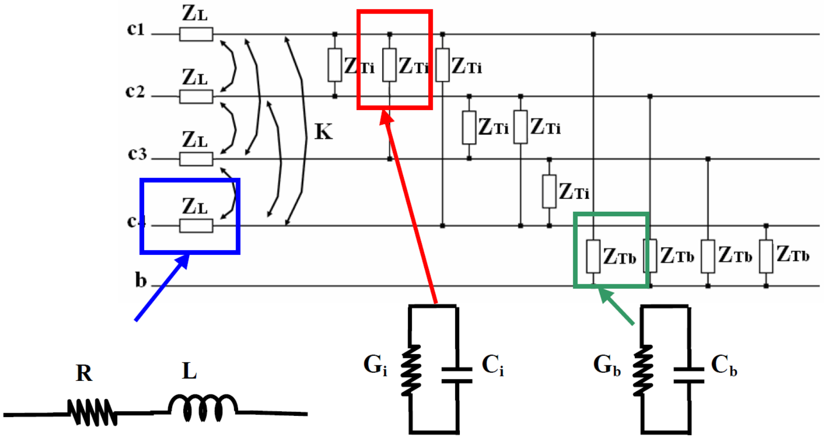


Figure 3.22 Discrete model of a power cable

The values of the passive elements in *Fig. 3.22* depend on the materials and the geometry used to physically build the cable. To model them, the following equations [29] were considered:

$$R_c = \frac{\rho l}{\pi}\left(r_1^2 - \left(r_1 - \frac{1}{\sqrt{\pi\mu\sigma f}}\right)^2\right) \tag{3.38}$$

$$C_c = \frac{2\pi\epsilon}{ln(\frac{15D}{16r_1})} \tag{3.39}$$

$$L_c = \frac{\mu}{2\pi\ ln(\frac{3D}{4r_1})} \tag{3.40}$$

where r_1 is the radius of one conductor, ρ is the resistivity of the material, μ is the magnetic permeability, σ is the conductivity of the conductor, ϵ is the permittivity of the dielectric, f is the frequency of evaluation and D is the distance between two conductors.

Secondly, since the cable can be modelled like the passive circuit shown in *Fig. 3.22*, its transfer function can be expectedto be LTI *Linear Time-Invariant*.

To model the behaviour of the power line we need to consider the three-phase currents i_A, i_B and i_C, while the model of the motor, as well as the controller, are referred to the direct and quadrature currents i_d and i_q. A coordinate transformation will be needed in order to refer the proper effects to the right quantity. This means that the model of the cable will be situated in between two nonlinear transformations, seen previously and described by *equation 3.10*.

To describe the entire transfer function of the cable, a 3 × 3 matrix will be needed, as the description of the effects of each phase on the other three must be given. On the other hand, no voltage sensors were included in the design so the compensation of this block cannot be performed directly on its inputs and outputs. This is a control-related complication which could significantly complicate the design of the regulator.

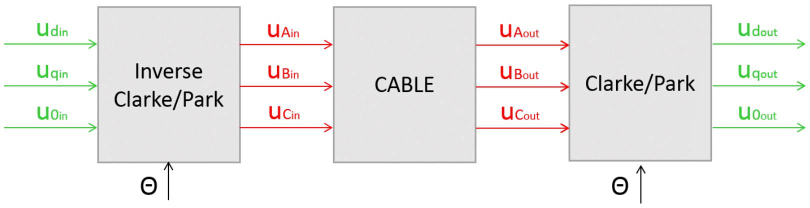


Figure 3.23 Situation of the cable model block

This second problem will be the first to be faced. In fact, if the effect of the cable on the voltage related to the control, u_q, is not clear or violently nonlinear, a complex and accurate model of the three-phase cable itself would be of a little use.

For now, the cable will be indicated as a 3 × 3 matrix:

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \tag{3.41}$$

since the elements on the diagonal represent the direct effect (on the same phase) while the other ones are a representation of the cross-talk, we can write:

$$T = \begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix} \tag{3.42}$$

Then, referring to *Fig. 3.24*,

$$\begin{bmatrix} u_{A_{in}} \\ u_{B_{in}} \\ u_{C_{in}} \end{bmatrix} = \begin{bmatrix} c_\theta & c_{\theta-\frac{2\pi}{3}} & c_{\theta+\frac{2\pi}{3}} \\ -s_\theta & -s_{\theta-\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}^{-1} \begin{bmatrix} u_{d_{in}} \\ u_{q_{in}} \\ u_{0_{in}} \end{bmatrix} \tag{3.43}$$

or, in a compact notation,

$$u_{ABC_{in}} = K_{C/P}^{-1} u_{dq0_{in}} \tag{3.44}$$

and, consequently,

$$u_{dq0_{out}} = K_{C/P} T K_{C/P}^{-1} u_{dq0_{in}} \tag{3.45}$$

recalling that the matrix associated with the Clarke/Park transformation is orthogonal, it yields:

$$\begin{bmatrix} u_{d_{out}} \\ u_{q_{out}} \\ u_{0_{out}} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} c_\theta & c_{\theta-\frac{2\pi}{3}} & c_{\theta+\frac{2\pi}{3}} \\ -s_\theta & -s_{\theta-\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix} \begin{bmatrix} c_\theta & -s_\theta & \frac{1}{2} \\ c_{\theta-\frac{2\pi}{3}} & -s_{\theta-\frac{2\pi}{3}} & \frac{1}{2} \\ c_{\theta+\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_{d_{in}} \\ u_{q_{in}} \\ u_{0_{in}} \end{bmatrix} \tag{3.46}$$

Given this, we can now compute the transfer function of the cable referred to the quadrature voltage $u_{q_{out}}/u_{q_{in}}$, supposing that the $u_{d_{in}}$ is always zero. Indicating the Clarke/Park matrix like:

$$K_{C/P} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{3.47}$$

we get

$$\frac{u_{q_{out}}}{u_{q_{in}}} = \frac{2}{3}(t_D((k_{21})^2 + (k_{22})^2 + (k_{23})^2) + 2t_C(k_{21}k_{22} + k_{21}k_{23} + k_{22}k_{23})) \tag{3.48}$$

substituting the actual value of the parameters:

$$\frac{u_{q_{out}}}{u_{q_{in}}} = t_D - t_C \tag{3.49}$$

the 4.10 shows how the relation between the input and the output quadrature voltages on the cable is a linear, time invariant function having the cable transfer functions as variables. Since the latters are linear, time invariant functions themselves, we can conclude that the dynamic of the quadrature voltage in the cable is LTI. This result is encouraging from the control point of view, since linear systems are easier to model and to compensate. The entire computation of the effects of the cable on the $d-q$ axis is described in [APPENDIX A].

The evaluation of the t_D and t_C dynamics was performed through the *linear analysis* tool in Simulink. This tool extracts a transfer function from a Simulink schematic, in which the input and output sets are defined. For the model of the power cable, the schematic in *Fig. 3.24* was used. This schematic includes the motor model as a part of the design since, as seen before, it cannot be approximated with an open circuit.

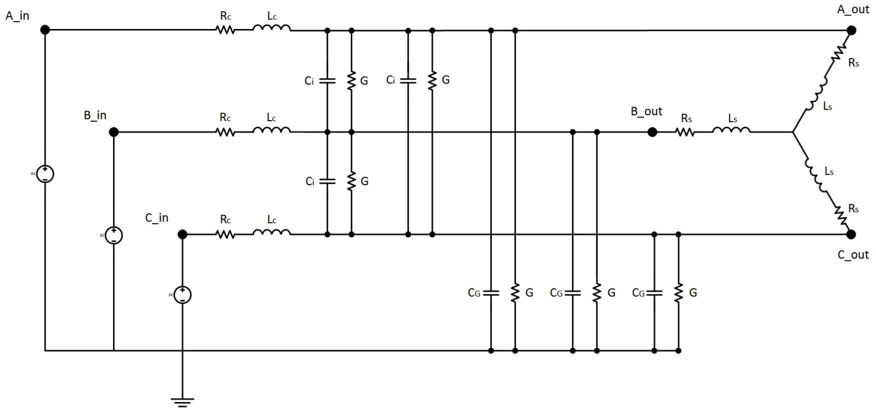


Figure 3.24 Electric schematic of the power line

Fig. 3.25 shows the bode plots relative to the model in *Fig. 3.24* for attenuation and cross-talk effects. The actual shape of the frequency response is difficult to understand because of the major contribution of the resonance peaks in the plot. These peaks become lower in magnitude and move to lower frequencies as the length of the cable increases.

3.7.4 Experimental cable model verification

The goal of this experimental session is to qualify the reliability of the previously described model for the power cable. It is articulated in two main parts:

- Experimental verification of the passive elements value.
- Experimental verification of the cable transfer function.

Since an experimental computation of the cable capacitance and inductance was challenging with the available hardware, only the value of R_c was considered.

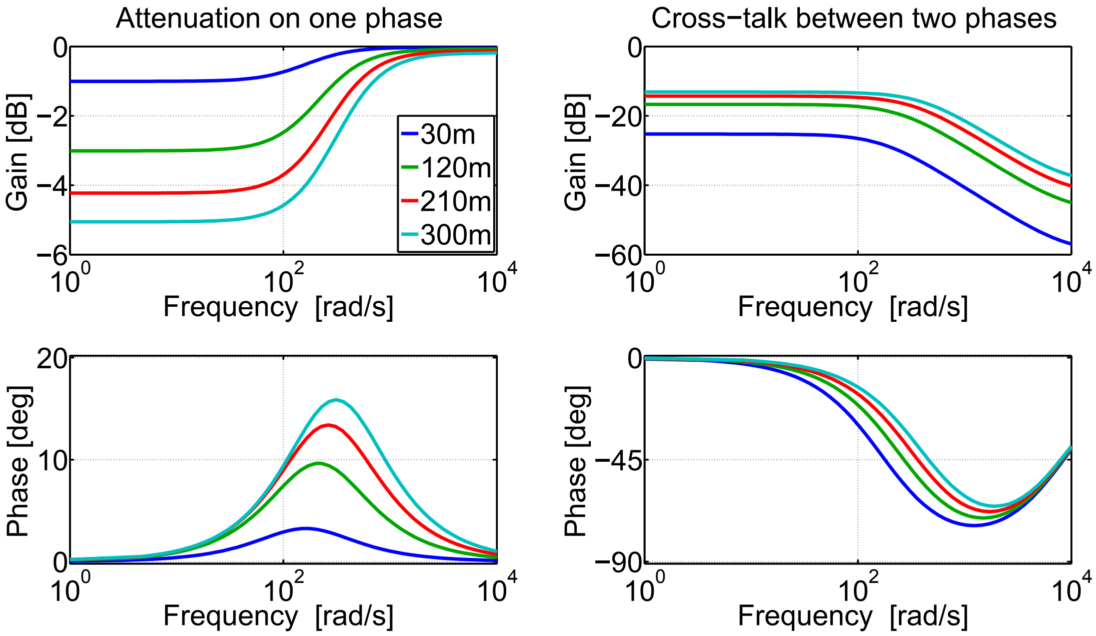


Figure 3.25 Effects of the power line on the same phase and on different phases for variable cable lengths

The test produced anyway interesting results, showing values well fit with the considered model. This was useful to validate the model at least for the resistance estimation.

Due to the low value of the cable resistance, the use of a milliohmeter was necessary. It was realised by means of a voltage divider, using a precision 1 Ω resistor with a 1% uncertainty as a reference.

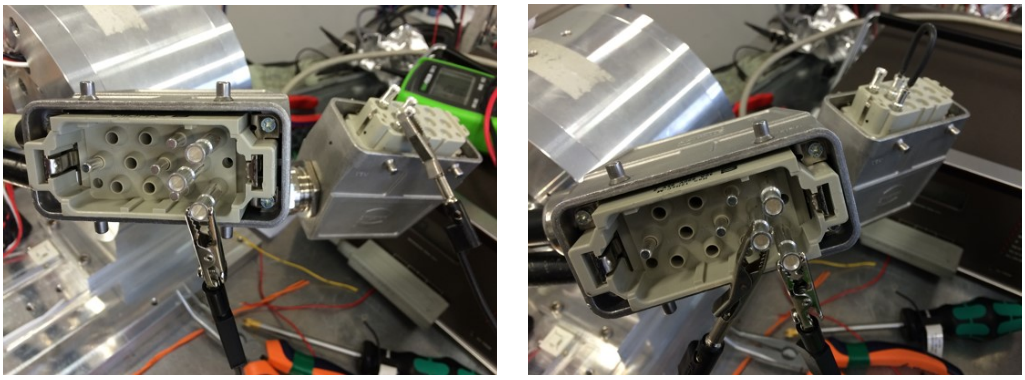
Then, the R_c value was computed as:

$$R_c = \frac{V_m}{V_{cc} - V_m} \tag{3.50}$$

To make the experiment more reliable, several measures were performed on several conductors configuring them in series or in parallel, as in *Fig. 3.26*. The measures were repeated for both the 10 m and the 100 m cables.

Finally the obtained values were organised in *Table 3.9* and the average value was considered. The table shows the fit between the modelled value and the experimental one.

After the measurement of the resistance of the conductors, a second experiment was run to confirm the cable model. It consisted in the evaluation of the frequency response of the cable.



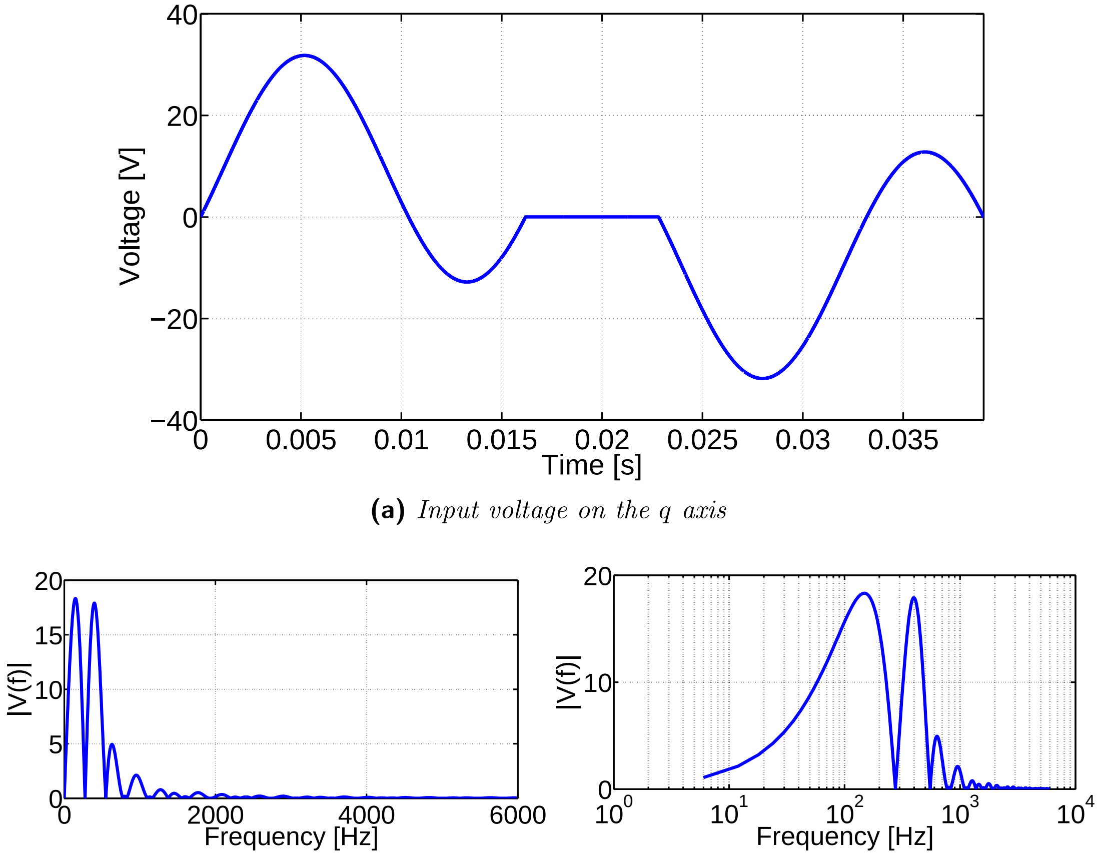
(a) *One conductor* **(b)** *Series of two conductors*

Figure 3.26 Different configurations of the conductors for resistance estimation

Conductors	Measured 100m	Model 100m	Measured 10m	Model 10m
1	0.3089 Ω		0.0329 Ω	
2 series	0.6761 Ω		0.0670 Ω	
3 series	0.9519 Ω		0.0957 Ω	
2 parallel	0.1575 Ω		0.0159 Ω	
3 parallel	0.1206 Ω		0.0147 Ω	
Average	0.3198 Ω	0.3169 Ω	0.0325 Ω	0.0317 Ω

Table 3.9 Measured values for the conductors resistance

Normally, for this kind of tasks, a network analyser is used. Commercial network analysers, though, typically work on range of frequencies which are too high for our goals: what we want to measure is the frequency response of the cable in the spectrum occupied by the voltage signal during a scan. This signal was simulated and its spectral content is represented in *Fig. 3.27*.



(a) *Input voltage on the q axis*

(b) *Single-sided amplitude spectrum*

(c) *Single-sided amplitude spectrum (logaritmic)*

Figure 3.27 Shape and FFT of the input voltage signal

The plot shows that the interested frequencies are much lower than the first harmonic introduced by the drive, corresponding to the PWM carrier at 16 kHz. The bound for the 'useful' frequencies was set to 10 kHz. Above this value, we will consider the signals to be cut by the power filter.

The testbench for the *tf* evaluation was built as in *Fig. 3.28.* A signal generator was used to feed a pure sine wave into one phase of the motor, connected with the 100 m long cable. The input signal, as well as the signal on two different phases of the motor were observed by means of an oscilloscope to qualify the attenuation and the cross-talk effects due to the power line.

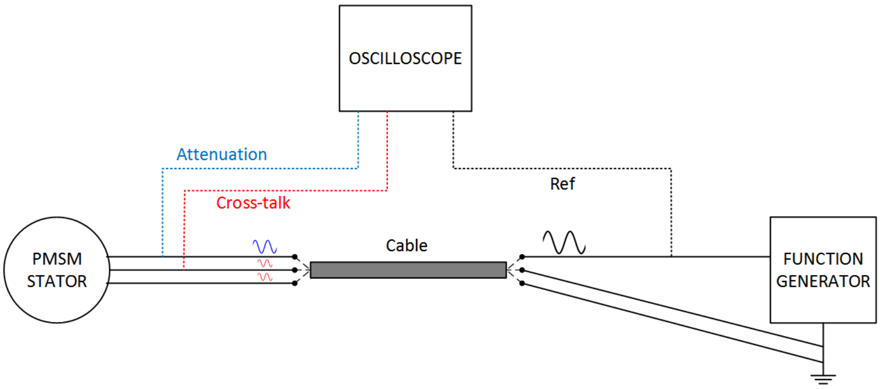


Figure 3.28 Schematic of the circuit used to evaluate the frequency response of the cable

By plotting the output/input ratios of the acquired voltages and comparing them with the simulations, *Fig. 3.29* was obtained.

The simulations and the experimental data show a good fit, except for some phase values that were more challenging to be evaluated due to small delay and harmonic distortions. These results show that the model of the cable is reliable - at least for low frequencies - and it will then be considered in the controller design part for simulations, where a compensation method will be suggested.

3.7.5 Experimental filter qualification

The last part of this section is dedicated to the qualification of the power filter. In particular, the low-pass effects of the filter towards to the spectral components generated by the PWM will be considered. The experiments were set up like in *Fig. 3.30*.

The voltage signal was acquired through a high-impedance voltage divider to prevent the oscilloscope from being damaged by the 300 V line.

Fig. 3.31 shows the differential voltage between two phases of the motor when they are driven with a pure sinusoidal signal through the inverter. The following *Fig. 3.32* shows the FFT of the previous signal both in linear and logarithmic scale. As a conclusion on this investigation we can observe that the effects of the sine filter are evident, since all of the harmonic components at high frequency are attenuated except for the carrier of the PWM, at $16kHz$.

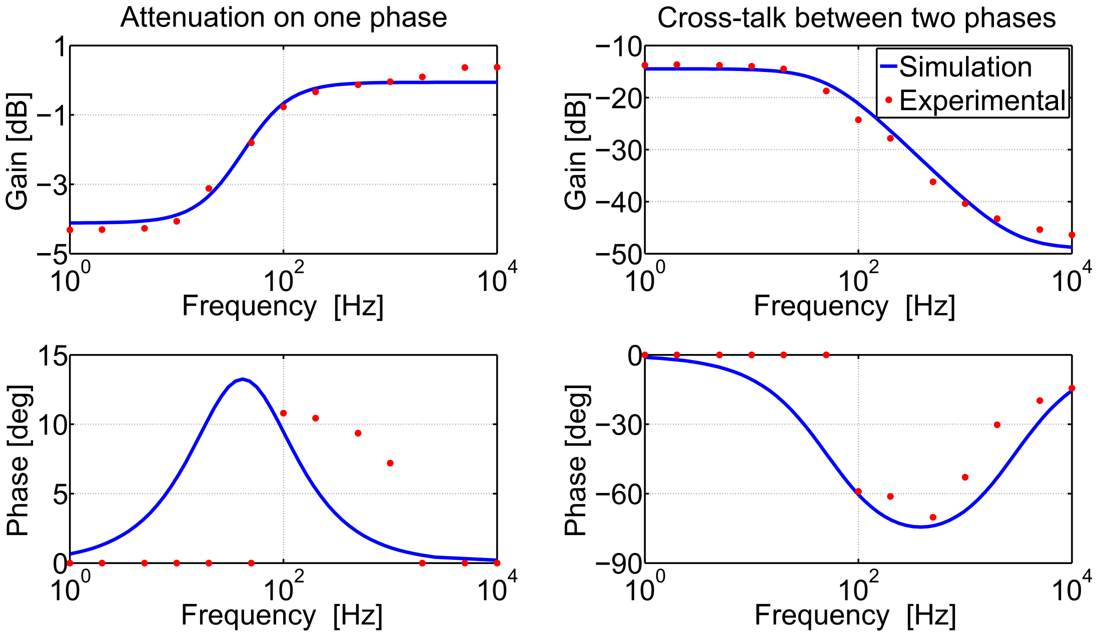


Figure 3.29 Effects of the power line on the same phase and on different phases for variable cable lengths

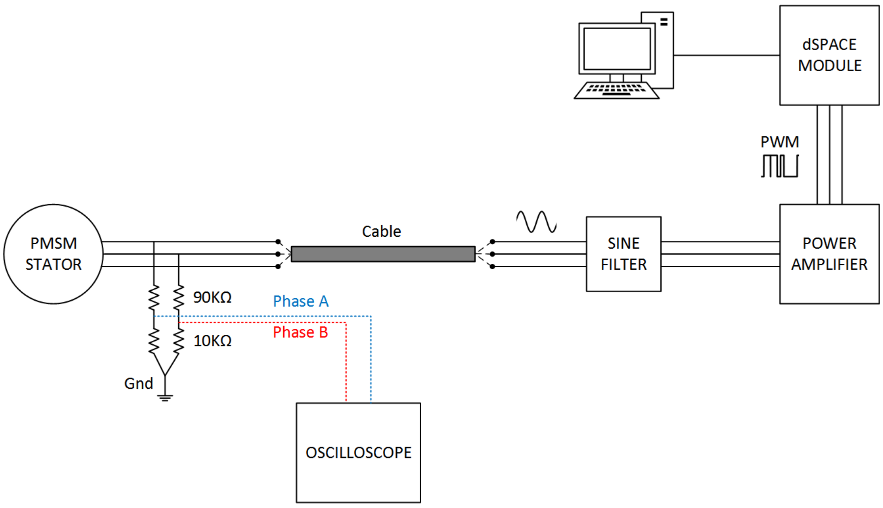
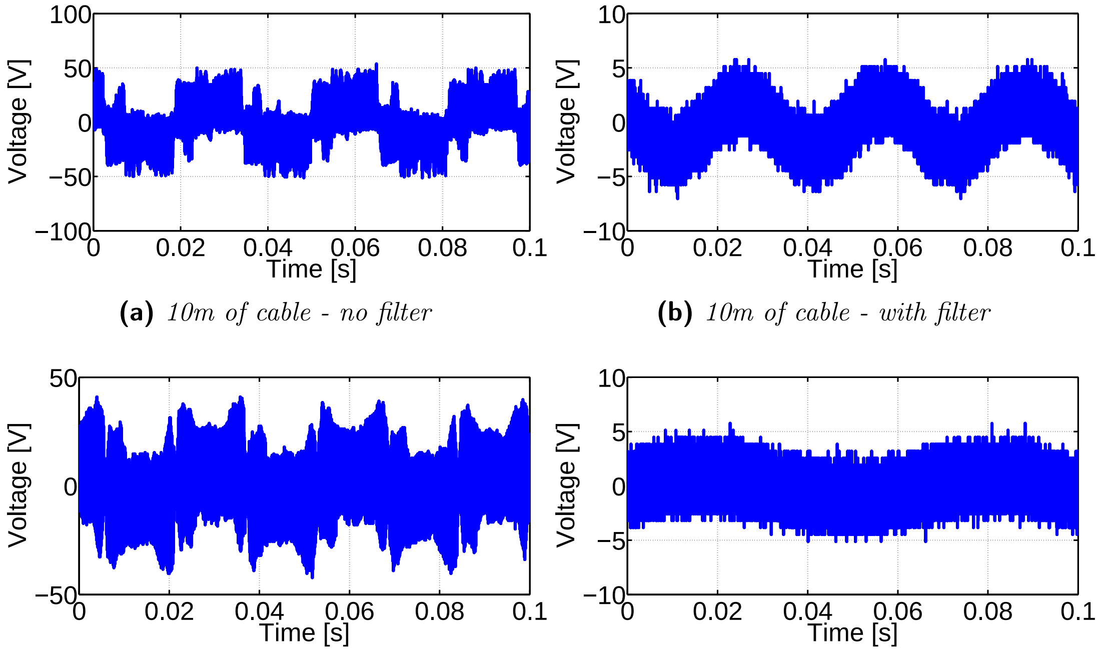


Figure 3.30 Schematic of the circuit used to qualify the effects of the filter

Besides selecting a different filter with higher performance, which will probably be necessary as a ultimate solution to the problem, some control methods can be included to mitigate the lower stability bound and vibrations introduced by the high-frequency contents present of the power line. These complications will be as described in the next chapter, dedicated to the control design.



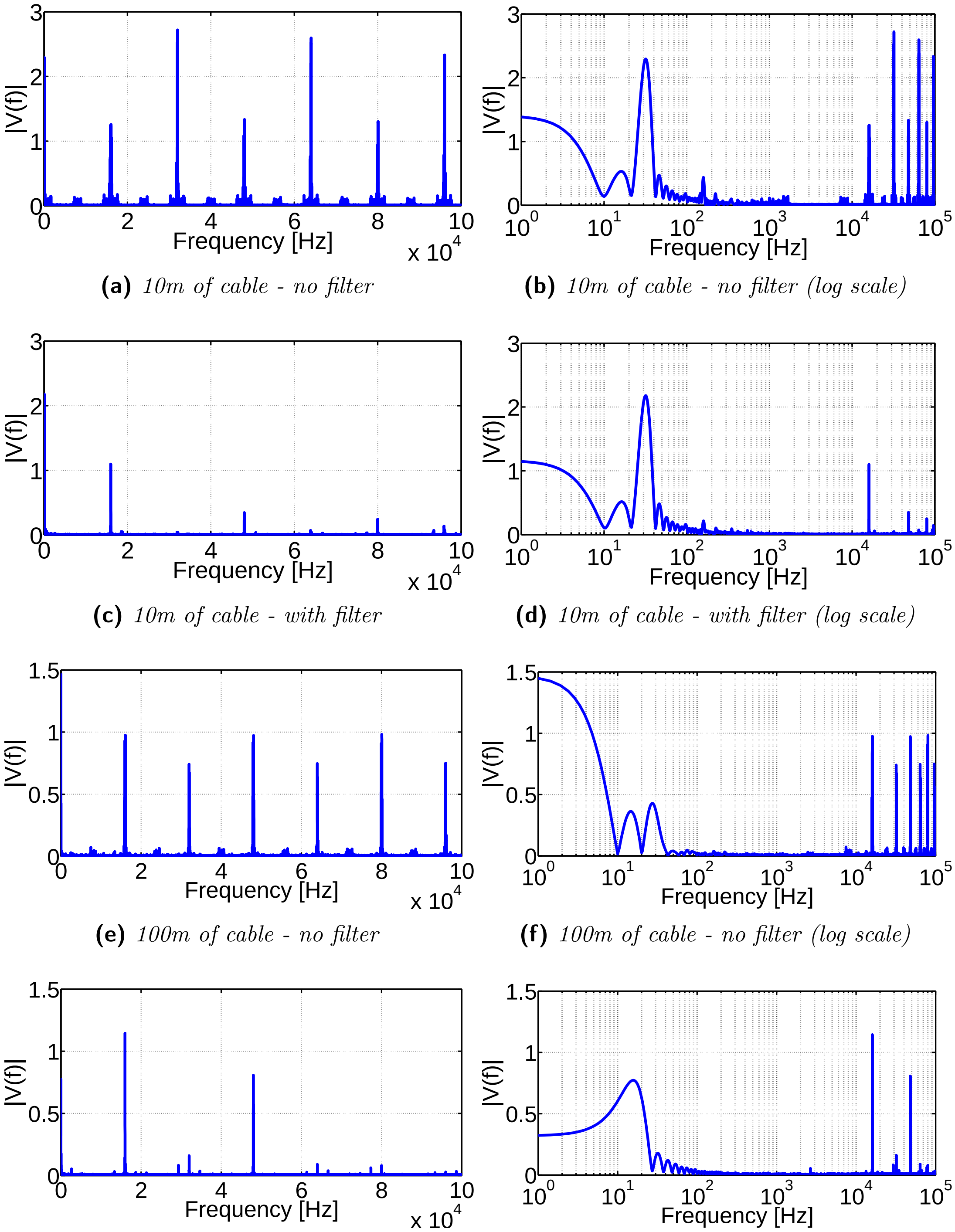
(a) *10m of cable - no filter*

(b) *10m of cable - with filter*

(c) *100m of cable - no filter*

(d) *100m of cable - with filter*

Figure 3.31 Time-domain plot of a pure sine wave transmitted on the power line for different cable length and configurations



(a) *10m of cable - no filter*

(b) *10m of cable - no filter (log scale)*

(c) *10m of cable - with filter*

(d) *10m of cable - with filter (log scale)*

(e) *100m of cable - no filter*

(f) *100m of cable - no filter (log scale)*

(g) *100m of cable - with filter*

(h) *100m of cable - with filter (log scale*

Figure 3.32 FFT of the signal measured on the power line for different cable length and configurations

4. Controller design

In this chapter, the control system design will be treated. For the basic architecture, a field oriented controller has been chosen. The standard regulator has been improved into several of its parts in order to increase the motion stability and performance, to deal with environmental effects and minimise vibrations. A steady-state Kalman filter was implemented to increase speed sensing accuracy and compared with standard and filtered discrete derivatives. For the tuning of the parameters, several approaches have been investigated and implemented in order to obtain the best possible scanning performances and to provide a repeatable tuning methodology for the next users. Finally, a compensation for the cable dynamics has been proposed and tested in simulation. All the simulations, based on the dynamic model obtained in the previous chapter, have been run on Matlab and Simulink.

4.1 Motion control techniques

Motion control area is a result of applying control theory to power electronics. Currently it is a fairly matured field, which is almost three decades old. The research on application of microprocessors for electrical drive control in the 1970's laid the foundation stone for the Motion Control area [41]. Nowadays, power electronics and motion control can be treated as two distinct fields that emerge as the technology advances.

In the next pages, the design of a motion controller dedicated to the beam wire scanner at CERN will be analysed and explained in its architecture and functions.

4.1.1 Field oriented control (FOC)

Because brushless motors are not self-commutating, they are more complicated to control. Brushless motors have three windings, rather than two. The currents and voltages applied to the motor windings must be controlled independently and correctly as a function of rotor position in order to produce useful torque. The electronics required to drive brushless motors are therefore substantially more complex than those required by brushed motors.

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. One component defines the magnetic flux of the motor, the other the torque. The control system of the drive calculates from the flux and torque references given by the drive's speed control the corresponding current component references. A simple PID controller can be used to keep the measured current components at their reference values. The pulse-width modulation of the variable-frequency drive defines the transistor switching according to the stator voltage references that are the output of the PID current controllers. It was originally developed for high-performance motor applications that are required to operate smoothly over the full speed range, generate full torque at zero speed, and have high dynamic performance including fast acceleration and deceleration. However, it is becoming increasingly attractive for lower performance applications as well due to FOC's motor size, cost and power consumption reduction superiority.

The working principle of the FOC is therefore based on the control of the direct and quadrature currents i_d and i_q and on the corresponding Clarke-Park transformations, as seen in *section 3.2.3.* It is represented in *Fig. 4.1.* The inner feedback loop controls the current to follow the given reference, expressed in the d-q frame. The middle and outer loops respectively control speed and position, so that the reference given to the next regulator is represented by the output of the previous one. This constitutes a pure cascade architecture where the absolute dynamic response of the inner mechanisms is partly affected by the characteristics of the outer ones and vice-versa. This means that if, for instance, the speed loop presents some delay, the current controller will be activated after a latency time, slowing down its response as well. On the other hand, if the current loop is delayed, the speed regulation produced in the outer loop will act on the system after a certain amount of time even if the speed loop reacts right away.

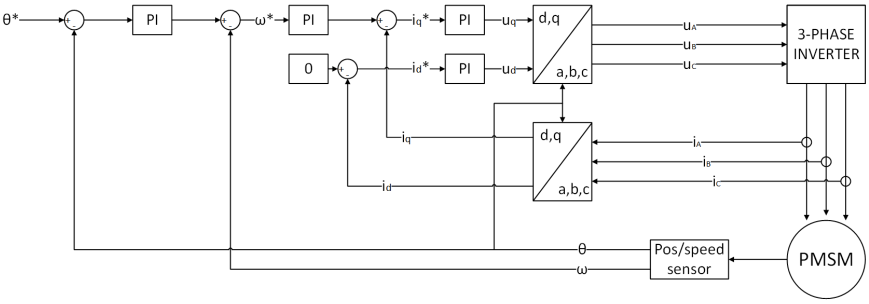


Figure 4.1 Field-oriented control schematic

The main advantages in the use of Field Oriented Control consists in the decoupling of the control problem into three levels: position, speed and current. The controllers can be tuned independently, providing design freedom in terms of performance and robustness. Furthermore, the transformation to the $d - q$ frame is useful to isolate the PID controllers from the time-varying currents and voltages, avoiding typical problems like saturations and making the position and speed controls possible despise an eventual inner loop low bandwidth.

4.2 Field Oriented Control applied to the BWS actuator

The classic scheme of Field Oriented Control maintains its fundamental architecture for most of the applications, except for some modifications. Servo-drives can be realised using different power supplies, different sensors and actuators and still adopt the control architecture shown in *Fig. 4.1*.

In the particular case of the Beam Wire Scanner, several details in the FOC structure have to be modified in order to properly describe the hardware. Once a reliable model has been built on a Simulink environment, the controller can be designed to fit the specifications and verified on the model, before being implemented on the actual system. As an example, the PMSM was described like in *Fig. 4.2*

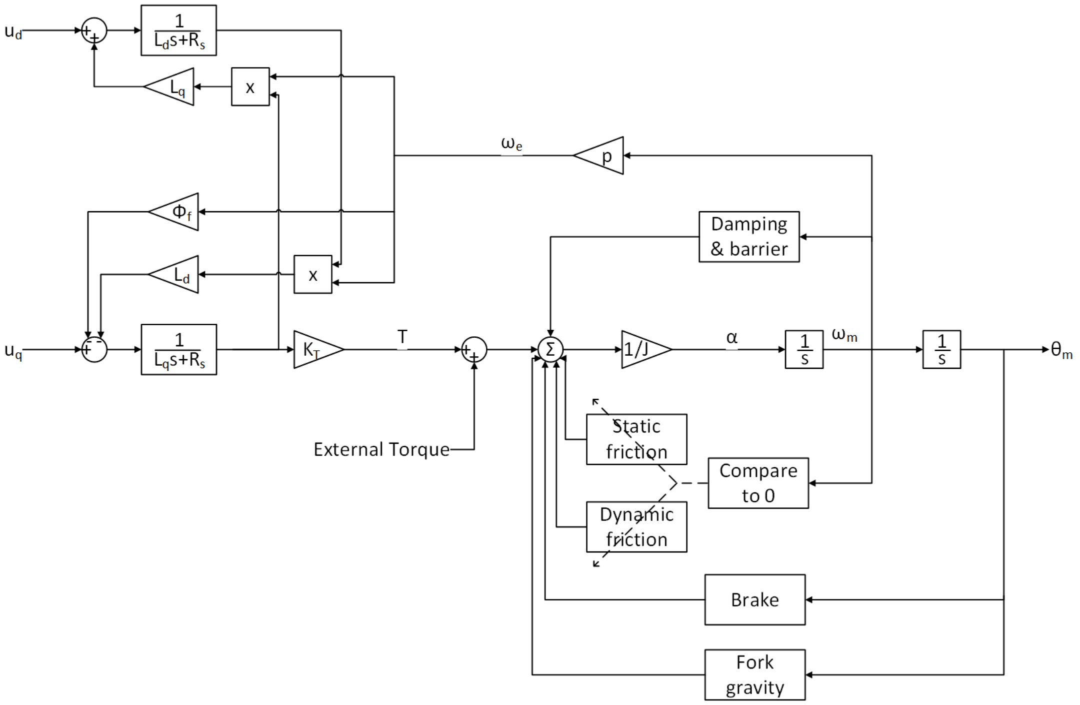


Figure 4.2 Simulink model of the PSMS in the $d - q$ frame

For the simulations, the blocks corresponding to real devices have been substituted by their models, obtained in the previous chapters, and finally compared to the experimental results. We will therefore use the complete model obtained in the previous chapter to describe the process, and interface the controller to its input and output connections to simulate the behaviour of the overall system.

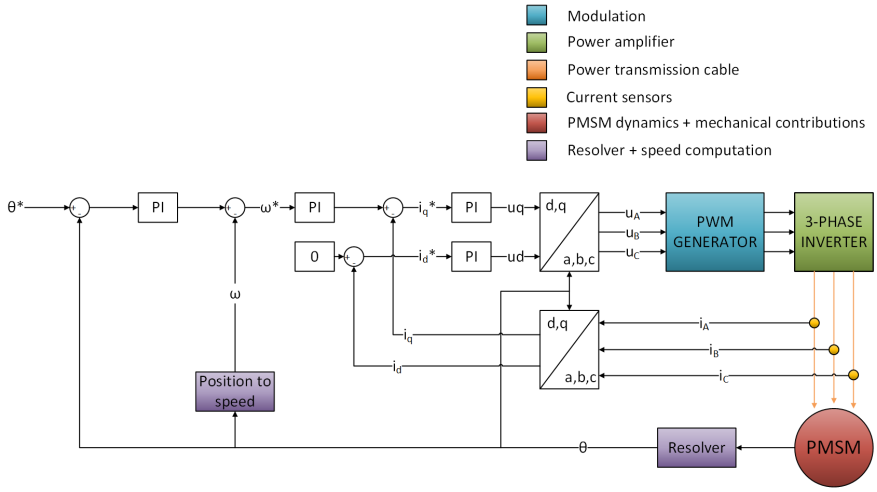


Figure 4.3 Field Oriented Controller schematic for the considered system

4.3 Motor dynamics decoupling

The first feature to be considered concerns the coupling between the two currents in the $d - q$ reference frame. It is described in current equations of the PMSM, here recalled in the Laplace domain:

$$
\begin{aligned}
sI_d &= \frac{1}{L_d}U_d - \frac{R_s}{L_d}I_d + \frac{1}{L_d}\Omega_e(L_q I_q) \\
sI_q &= \frac{1}{L_q}U_q - \frac{R_s}{L_q}I_q - \frac{1}{L_q}\Omega_e(L_d I_d + \Phi_f)
\end{aligned}
$$

A coupling exists between the two variables, and it is represented by the last contribution to each equation. It is partly due to the interaction between the two currents, and partly produced by the back-EMF generated by the motor itself.

This can be eliminated by a feedforward compensation. Considering the new inputs:

$$U'_d = U_d - \Omega_e(L_q I_q) \tag{4.2a}$$

$$U'_q = U_q + \Omega_e(L_d I_d + \Phi_f) \tag{4.2b}$$

this yields:

$$sI_d = \frac{1}{L_d}U'_d - \frac{R_s}{L_d}I_d \tag{4.3a}$$

$$sI_q = \frac{1}{L_q}U'_q - \frac{R_s}{L_q}I_q \tag{4.3b}$$

that are linear in their input and state variables. The concept of this transformation is shown in *Fig. 4.4*.

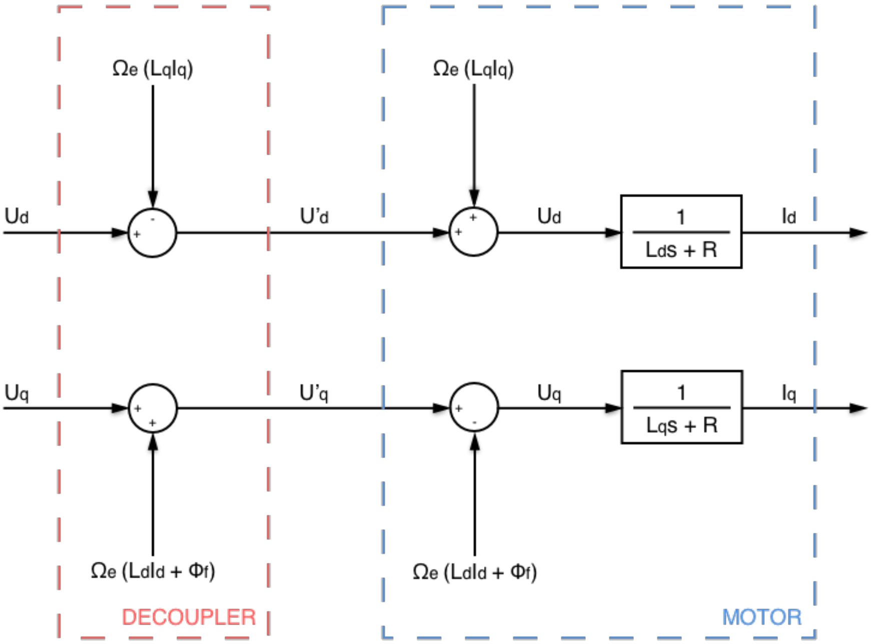


Figure 4.4 Decoupling action concept

After the decoupling block is inserted, the current loops assume the following convenient first-order form:

$$I_d = \frac{1}{L_d s + R_s}U_d \tag{4.4a}$$

$$I_q = \frac{1}{L_q s + R_s}U_q \tag{4.4b}$$

the advantage of having LTI equations describing the current loops is the possibility to perform the compensation with a PID controller using the typical design tools of the linear control. Furthermore, decoupling the $d-q$ currents leads to a sensitive increase of performance in terms of vibrations, since the control is more precise and torque fluctuations are avoided.

4.4 Anti-Windup solutions comparison and design

Every physical processes presents saturation problems due to limitations in the action of the actuators. They can be geometrical limits (typical of mechanical actuators such as motors, robotic arms, vehicles..) or electric ones (output voltage/current of a power supply, for example). In the control field, some saturations are generated by the process itself, while others are precisely designed and included in the system in order to avoid stressing the controller process.

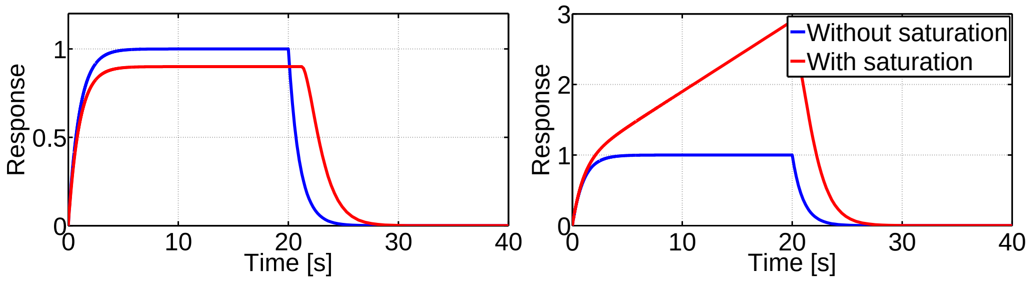
In the controller architecture of the BWS, all the PID regulators are saturated by design. The current and speed loop would have physical limits anyway, due to the power supply and the motor dynamics. The saturation limits, used to avoid peaks of current into the motor windings, are shown in *Table 4.1.* They also prevent the mechanics from being stressed.

Loop	Min value	Max value
Current	-1.5	1.5
Speed	-100	100
Position	-200	200

Table 4.1 Saturation values in the three PID controllers

The effects of such nonlinearities are well known, and they can be sometimes exploited to perform experimental tuning of the regulators (ATV technique, [30]). When a compensation through a PID controller is considered, the only problem rose by the presence of a saturation concerns the integral action: this is because the steady-state error, impossible to compensate due to the limits of the actuator, will keep being integrated. This effect takes the name of "windup" and its effects are shown in *Fig. 4.5.* They normally consist in a delayed response of the controller, that needs to "unwind" the accumulated value before getting back into the unsaturated zone.

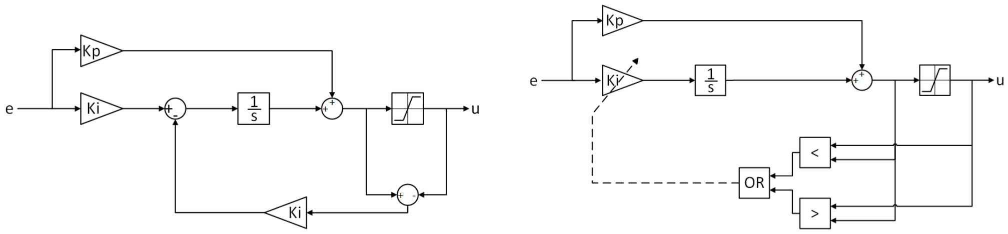
Anti-windup systems are included in controller architectures to solve saturation problems resulting from natural and technological limits in the actuators. There exist several different methods to implement this compensation. In this work, two of the most popular ones have been investigated: the *tracking* and the *conditioned* anti-windup architectures *Fig. 4.6.* The first one consists in feeding back the difference between the non saturated and the saturated outputs to



(a) *Saturation effect on output* **(b)** *Saturation effect on integrator*

Figure 4.5 Response of a system with saturated actuator

eliminate the integral effect. The second one checks the equality between the two values and, if they don't correspond to the same number, cuts off the integral block by giving it a constant zero input. The conditioned solution is more convenient for hardware implementation (and, in general, for digital controllers).



(a) *Tracking anti-windup* **(b)** *Conditioned anti-windup*

Figure 4.6 Block diagram of the two investigated anti-windup solutions

The performances of the two solutions are compared in *Fig. 4.7*. Due to the ease of implementation and the slightly better performances, the conditioned architecture was chosen and implemented in the motion controller.

4.5 Speed measurement limitations

The FOC architecture includes feedback loops for position, speed and current control, but only position and current sensors are available. That means that the instantaneous value of the speed has to be computed by exploiting the knowledge of the other two measures. Surely, the easiest and most immediate procedure is suggested by simple physics: perform a derivative of the position value, a discrete one in the case of a digital controller.

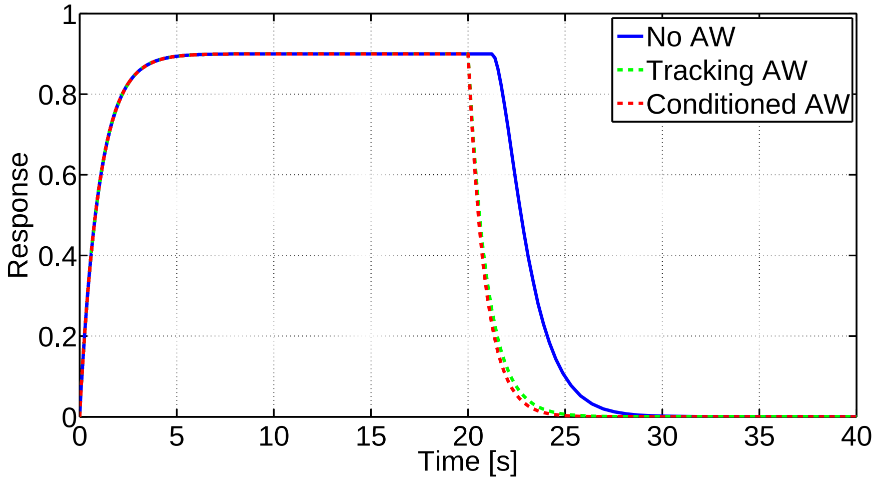


Figure 4.7 Compared performances of the two AW architectures

Unfortunately, this solution leads to the problem already addressed in *section 3.3*: the quantisation of the measured position. With the choice of 14 *bits* for the resolver resolution, the acquisition system was simulated on Matlab for a constant acceleration of 1000 rad/s^2. The discrete derivative of the the position, corresponding to the quantised speed, is plotted in *Fig. 4.8*.

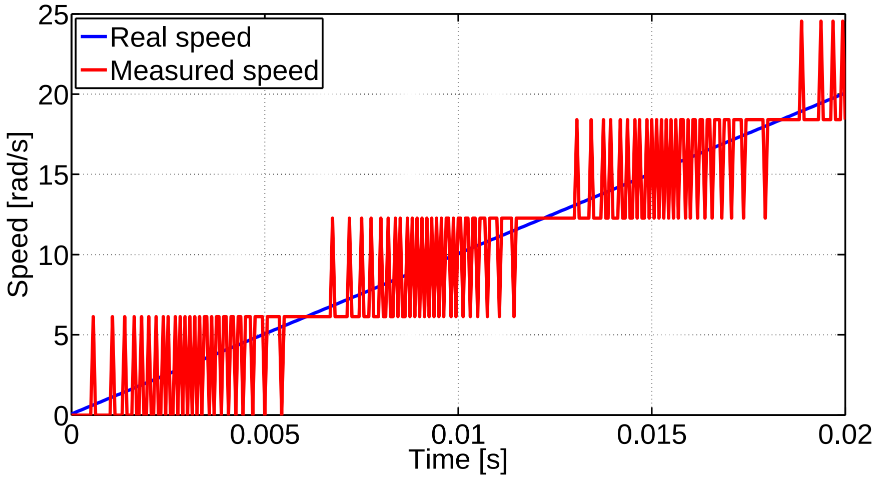


Figure 4.8 Effect of quantisation on speed computation

The main problems produced by this effect are:

- A reduced possible bandwidth for the speed loop: the ripply slope of the speed seen in *Fig. 4.8* has the same behaviour of noise. Therefore, the loop has to be shaped to withstand a bigger amount of noise, and a reduced bandwidth is a consequence.

- A lower performance in following speed references, especially in low-speed operations in which the quantisation noise largely overcomes the mechanical vibrations of the system.
- Oscillating behaviours.
- Audible noise during the motion.

Some of these effects are not a serious drawback to the system performance, while others affect crucial design requirements. Although the whole control system can also be implemented using a pure discrete derivative, in this section the typical smoothing methods and a more complex and performant solution based on Kalman filtering will be examined. After comparing the results - both simulated and experimental - we chose the best solution in terms of performance and reliability and implemented it in the final version of the motion controller.

4.5.1 Typical approaches

When we have to deal with a noisy signal the first idea is always the same: filtering. Analog and digital hardware allows to use a wide variety of filters to smooth signals perturbed by noise, like for the case of the speed signal coming from the discrete derivative of the position sensed by a resolver. In this case, what we need is a low-pass filter.

The two solutions adopted in DSP are *Finite Impulse Response* (FIR) and *Infinite Impluse Response* (IIR) filters. The difference between the two is, as the names suggest, the length of the response of the filter to an impulsive input. The FIR filters, being basically a weighted average of the last input values, stop reacting to an impulse after a certain number of samples. On the other hand, IIR filters are more complicated, including feedback actions, and they can keep ringing indefinitely after receiving an impulsive input. In these pages the performance of the two digital filter types will be compared, substituting the IIR with an analog filter to better specify its frequency response.

In *Fig. 4.9* the response of analog filters to the previously described problem is plotted.

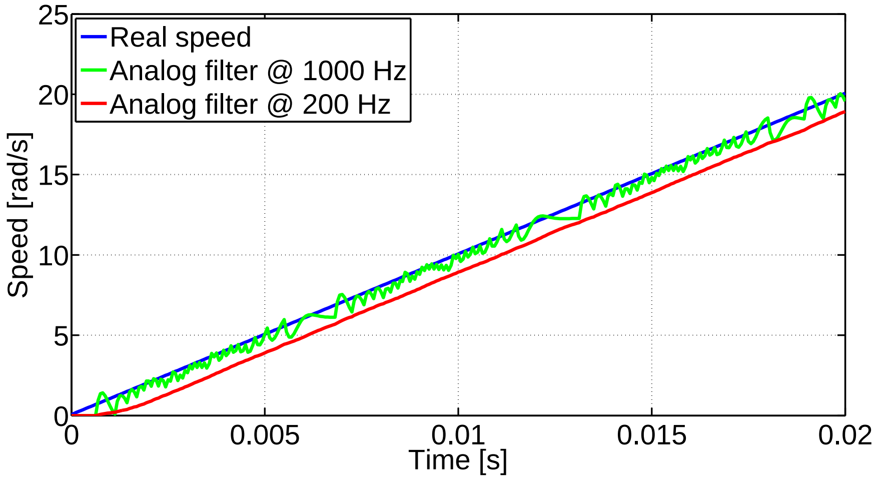


Figure 4.9 Speed filter - analog solution

The obtained results are quite clear:

- Both of the filters have a lag: this is normal for low-pass action, especially for digital filters.
- The solution with a higher bandwidth (1000 Hz) presents a lower lag but a large oscillating behaviour.

If the ripple in the red plot can generate problems in the feedback loop due to noise peaks, the delay in the green one is even worse from the control point of view.

In *Fig. 4.10* the performance of the FIR filter is shown.

Also in the case of digital filters, both of the two implementations present a small delay. We notice that, by increasing the width of the averaging window, the quantisation problem is mitigated while the delay increases.

Moreover, a typical approach of the control environment was implemented and simulated. It consists in computing the derivative of a signal by generating a control loop having as an output the actual measured value (the position, in this case) and extracting the signal before the integral action. This method is known as *dirty derivator*, and its equation can be written as:

$$\dot{\theta}_d = \frac{s}{s+a}\theta \tag{4.5}$$

where a is a design parameter used to limit the derivative action and so avoid peaks in the output where a step disturbance occurs. The value of a was tuned

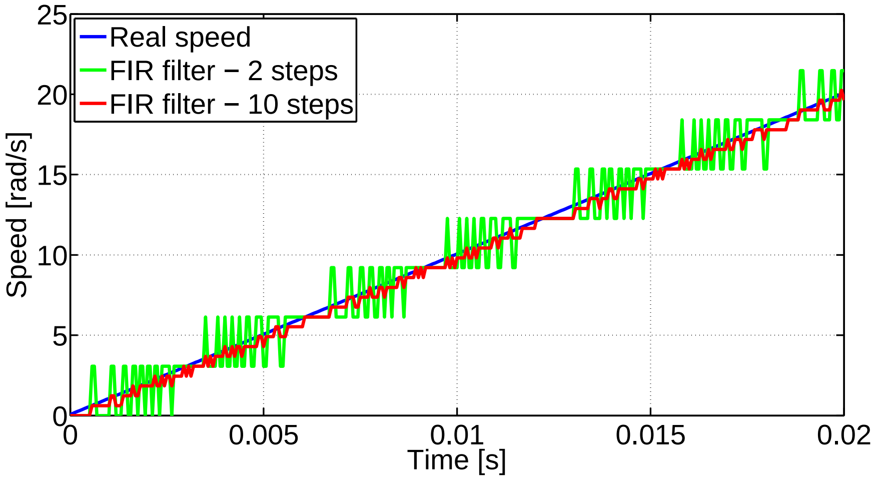


Figure 4.10 Speed filter - FIR solution

through an iterative method to minimize the *Integral Absolute Error* (IAE) of the output compared to the actual speed slope. It still presented the same problems as the other filters (*Fig. 4.11*).

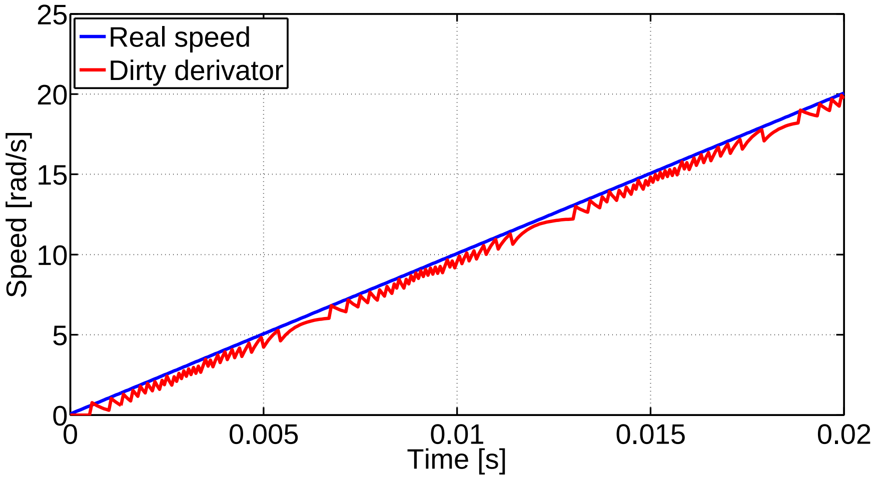


Figure 4.11 Speed filter - dirty derivator solution

4.5.2 Increased performance using the Steady-State Kalman Filter (SSKF)

Since the previously evaluated methods were not entirely satisfactory in terms of smoothness and delay, an enhanced system for speed filtering was considered.

A different solution, which introduces a smaller delay maintaining a high level of performance, consists in employing a state observer that provides the speed estimation, starting from the measured position. This observer can be of deterministic type [31] or of stochastic type, based on a Kalman filter [32]. The last solution allows a greater flexibility, and was preferred.

Kalman filtering allows the obtainment of both the advantages of a precise estimation performed following the principles of sensorless control [33] and the robustness of exploiting an actual measure through a position sensor (resolver).

The Kalman filter is based on the following stochastic discrete-time model of the mechanical system:

$$\begin{aligned} x_n &= Ax_{n-1} + b\tilde{a}_n \\ y_n &= c^T x_n + \eta_n \end{aligned} \tag{4.6}$$

where

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

Since the dynamic system given by (4.16) 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 $n \to \infty$. In this way a *Steady-State Kalman Filter* (SSKF) is utilised, with the advantage of the reduction in the number of mathematical operations [34].

Once the gain vector g is determined, the operations to be computed at every sampling interval are only those related to prediction and correction:

$$\begin{aligned} \tilde{x}_n &= A\hat{x}_{n-1} + b\tilde{a}_n \\ \hat{x}_n &= \tilde{x}_n + g(\theta_n - c^T \tilde{x}_n) \end{aligned} \tag{4.7}$$

The value of the estimated acceleration $\tilde{a}$ can be obtained from the measure of the currents: in fact, the torque generated into the motor is proportional to the quadrature current, as in (3.16). The conceptual scheme is like in *Fig. 4.12.*

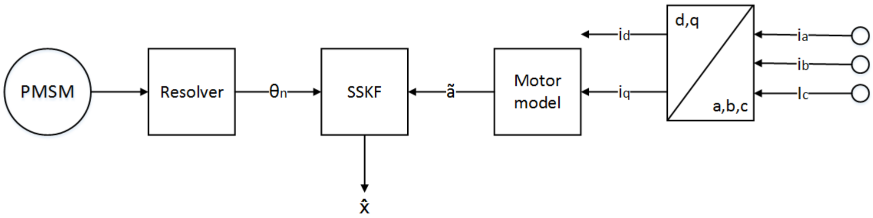


Figure 4.12 Conceptual schematic of the Stedy-State Kalman Filter

4.5.3 SSKF tuning

The dynamic characteristics of the speed control loop are strongly influenced by the properties of the SSKF algorithm and so by the gain vector. To avoid this issue, an approach based on the allocation of the filter transfer function poles can be adopted. As shown in [35], the poles of the discrete time transfer function of the filter are equal to the eigenvalues of the following dynamic matrix:

$$\tilde{A} = A(I - gc^T) \tag{4.8}$$

Defining the following vector:

$$l = Ag = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \tag{4.9}$$

dynamic matrix $\tilde{A}$ can be expressed as:

$$\tilde{A} = A - lc^T = \begin{bmatrix} 1 - l_1 & T_s \\ -l_2 & 1 \end{bmatrix} \tag{4.10}$$

Imposing the determinant of matrix $\lambda I - \tilde{A}$ equal to zero, the following characteristic equation is obtained:

$$\lambda^2 + c_1\lambda + c_0 = 0 \tag{4.11}$$

where

$$\begin{aligned} c_1 &= -2 - l_1 \\ c_0 &= 1 + l_1 + T_s l_2 \end{aligned} \tag{4.12}$$

Choosing (4.11) to have two real eigenvalues $\lambda_1 = \rho_1$ and $\lambda_2 = \rho_2$, we can write:

$$\begin{aligned} c_1 &= -\rho_1 - \rho_2 \\ c_0 &= \rho_1 \rho_2 \end{aligned} \tag{4.13}$$

The eigenvalues of $\tilde{A}$ can be imposed by discretization of a continuous filter characterised by two real negative poles of modulus p_0 and p_1:

$$\begin{aligned} \rho_0 &= e^{-p_0 T_s} \\ \rho_1 &= e^{-p_1 T_s} \end{aligned} \tag{4.14}$$

Different simulations have shown that it is convenient to select for p_0 and p_1 values three, five time as high as the bandwidth foreseen for the drive. Then, the coefficients in (4.12) can be derived as:

$$\begin{aligned} l_1 &= c_1 + 2 \\ l_2 &= \frac{c_0 - c_1 - 3}{T_s} \end{aligned} \tag{4.15}$$

Finally, the elements of the gain vector g can be computed as:

$$\begin{aligned} g_2 &= l_2 \\ g_1 &= l_1 - T_s g_2 \end{aligned} \tag{4.16}$$

The described tuning method, followed by experimental verification, allowed the final gain value to be fixed as $g = [1\ 2000]'$.

The performance of the SSKF are shown in *Fig. 4.13*.

Apart from the low level of quantisation noise, the main achievement is represented by the almost total absence of lag. This could be obtained because of the estimation of the acceleration derived by the current vector.

In *Fig. 4.14* a comparative of the filters performances based on *Integrated Absolute Error* (IAE) is shown.

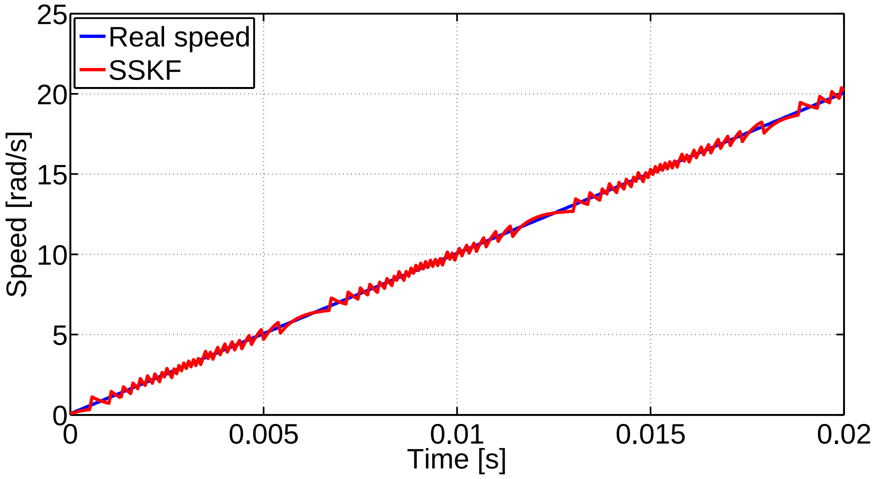


Figure 4.13 Speed filter - SSKF

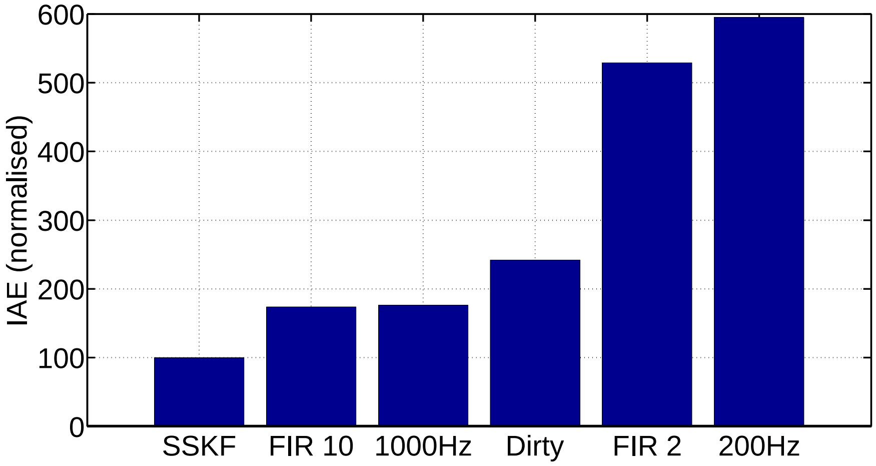


Figure 4.14 Speed acquisition methods comparison

4.6 PID controllers tuning

The controller components which are responsible for the dynamic response of the three control loops are the actual PI regulators. Thus, particular care has to be taken in their parametric tuning, in order to obtain a good precision and fast performances. Since the transfer function of the two $d-q$ currents is the same, the number of parameters to be tuned is 6: proportional and integral actions for the position, speed and current loops.

This section will give an exhaustive description of the used tuning methods, in increasing order of complexity, performance and experimental reliability.

4.6.1 Theoretical approach

As a first method, a theoretical approach to PID tuning based on the knowledge of the various dynamic responses of the beam wire scanner actuator system will be described. This will be useful to generate a set of "basic" parameters, theoretically stable and reasonably efficient, to initialise the experimental setup. Furthermore, theoretical methods have the great advantage of repeatability, which is crucial in situations where the system parameters are not themselves ultimate and a future re-tuning could be necessary.

The theoretical tuning method is based on first-order approximation and zero-pole cancellation [36] the zero inserted by the combination of proportional and integral action of the PID controller is positioned in correspondence of the pole at the lowest frequency present in the first-order approximation of the process.

4.6.1.1 Current loop design

The inner loop is the first to be shaped, since its dynamic response will influence the outer ones as well.

To perform the tuning, we will refer to the i_q variable (i_d is equivalent). The two main contributions to be considered are:

- The actual current dynamics, representing the most important component of the current loop response.
- The contribution of the control algorithm T_{alg} and the switching T_{switch} to the total delay.

Recalling the transfer function for the i_q, and including the gain factor due to the amplifier:

$$G_{I_q\_el} = \frac{U_{DC\_n}}{2} \frac{1}{L_q s + R_s}$$

Where $U_{DC\_n} = 300\ V$ is the nominal voltage provided by the power supply.

Considering the total delay to be constituted by one sampling period to perform the control algorithm routine, and another half sampling period to model the switch, like in [28], we obtain:

$$T_{tot} = T_{alg} + T_{switch} = T_s + \frac{T_s}{2} = \frac{3}{2}T_s \tag{4.17}$$

and substituting this delay with its first-order approximation:

$$G_{I_q\_del} = \frac{1}{T_{tot}s + 1} \tag{4.18}$$

the complete dynamic in Bode form becomes:

$$G_{I_q} = G_{I_q\_el}\ G_{I_q\_del} = \frac{U_{DC\_n}}{2} \frac{1}{L_q} \frac{1}{s + \frac{R_s}{L_q}} \frac{1}{T_{tot}} \frac{1}{s + \frac{1}{T_{tot}}} \tag{4.19}$$

Considering now the generic expression of a PI controller:

$$G_{PI} = K_a\ \frac{(K_b + s)}{s} \tag{4.20}$$

the open-loop tf of the inner loop can be written as:

$$G_{I_q\_OL} = \frac{U_{DC\_n} K_{a\_I_q} R_s}{2 L_q T_{tot}} \frac{(K_{b\_I_q} + s)}{s} \frac{1}{s + \frac{R_s}{L_q}} \frac{1}{s + \frac{1}{T_{tot}}} \tag{4.21}$$

Imposing now:

$$K'_{a\_I_q} = \frac{U_{DC\_n} K_{a\_I_q} R_s}{2 L_q T_{tot}} \tag{4.22}$$

and the cancellation of the lowest-frequency pole by means of the PID zero:

$$K_{b\_I_q} = \frac{R_s}{L_q} = 182.8358 \tag{4.23}$$

we obtain the final open-loop expression:

$$G_{I_q\_OL} = K'_{a\_I_q}\ \frac{1}{s + \frac{1}{T_{tot}}} \tag{4.24}$$

from which it is possible to compute the corresponding closed-loop expression, recalling the usual equation $G_{CL} = G_{OL}/(1 + G_{OL})$.

I can be written as:

$$G_{I_q\_CL} = \frac{K'_{a\_I_q}}{s^2 + \frac{1}{T_{tot}}s + K'_{a\_I_q}} \tag{4.25}$$

that, compared to the canonic form of the second-order transfer function:

$$G_{2^{nd}} = \frac{\omega^2}{s^2 + \xi\omega s + \omega^2} \tag{4.26}$$

suggests $K'_{a\_I_q} = \omega^2$.

This relation can be exploited in the design by imposing the bandwidth of the closed-loop function to derive the corresponding value of $K_{a\_I_q}$. Imposing $\omega_{I_q} = 10^4$, we obtain:

$$K_{a\_I_q} = \frac{2\omega_{I_q}^2 L_q T_{tot}}{U_{DC\_n}} = 0.0838 \tag{4.27}$$

The designed PI regulator, which step response is plotted in *Fig. 4.15*, assumes the form:

$$G_{PI\_I_q} = 0.0838\,\frac{(182.8358 + s)}{s} \tag{4.28}$$

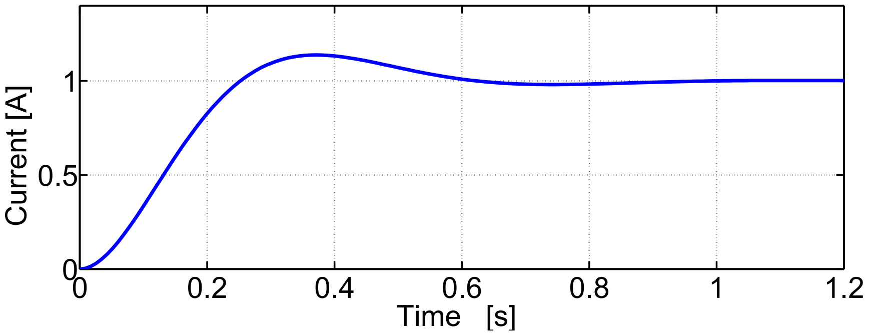


Figure 4.15 Step response of the current loop

4.6.1.2 Speed loop design

To shape the speed loop response, the same procedure used in the last section will be adopted.

The speed dynamic response is a product of the mechanical speed response as in (3.16), of the motor and the current dynamics, as in *Fig. 4.1.* For these computations, the current loop is approximated to its first-order representation:

$$G_{I_q\_1^{st}} = \frac{1}{1+\frac{s}{\omega_{I_q}}} \tag{4.29}$$

$$G_\omega = G_{\omega\_mech}\, G_{I_q\_1^{st}} = \frac{K_T}{J}\, \frac{1}{s+\frac{B}{J}}\, \omega_{I_q}\, \frac{1}{s+\omega_{I_q}} \tag{4.30}$$

The open-loop form of the speed dynamic can be written as:

$$G_{\omega\_OL} = \frac{K_T K_{a\_\omega}\omega_{I_q}}{J}\, \frac{(K_{b\_\omega}+s)}{s}\, \frac{1}{s+\omega_{I_q}}\, \frac{1}{s+\frac{B}{J}} \tag{4.31}$$

Similarly as above, the K_{b_ω} is chosen to cancel the lowest-frequency pole as:

$$K_{b\_\omega} = \frac{B}{J} = 375 \tag{4.32}$$

and renaming:

$$K'_{a\_\omega} = \frac{K_T K_{a\_\omega}\omega_{I_q}}{J} \tag{4.33}$$

we obtain the closed-loop transfer function as:

$$G_{I_\omega\_CL} = \frac{K'_{a\_\omega}}{s^2+\omega_{I_q}s+K'_{a\_\omega}} \tag{4.34}$$

again, imposing a lower bandwidth to the mechanical speed , like $\omega_\omega = 10^3$, it is possible to choose the K_{a_ω} parameter. The lower bandwidth is necessary for stability, since a speed sensor is not ideal both for time response and for oscillations due to the quantisation. It is still capable of providing a rise time $T_{rise\_w} = 0.01\ s$, which is enough of many control application, like the BWS motion.

It yields:

$$K_{a\_\omega} = \frac{\omega_\omega^2 J_{total}}{K_T\omega_{i_q}} = 0.1025 \tag{4.35}$$

Finally, the speed PI regulator assumes the form:

$$G_{PI\_\omega} = 0.1025\, \frac{(375+s)}{s} \tag{4.36}$$

4.6.1.3 Position loop design

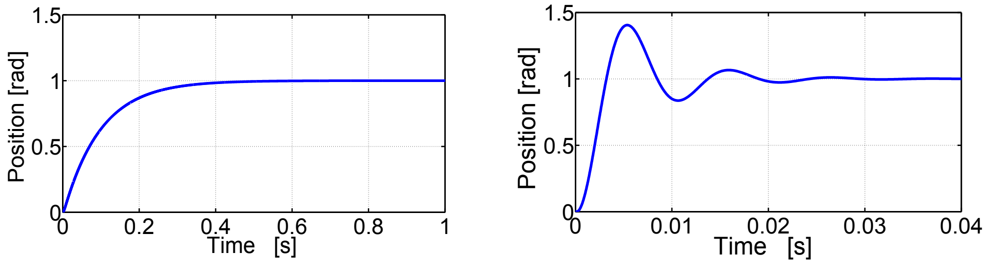
Recalling (3.18), the position transfer function is:

$$G_\theta = \frac{1}{s}\frac{1}{1+\frac{s}{\omega_\omega}} \tag{4.37}$$

A useful observation is that the position loop presents an integrator itself, so in principle it does not need the integral effect of the PID, while a simple proportional regulator should be enough to obtain a zero steady-state error. Thus, imposing $G_{P\_\theta} = K_{a\_\theta}$, the closed-loop equation yields:

$$G_{\theta\_CL} = \frac{\omega_\omega K_{a\_\theta}}{s^2 + \omega_\omega + s\omega_\omega K_{a\_\theta}} \tag{4.38}$$

In the case of the angle feedback, the tuning really depends on the specific application requirements. in *Fig. 4.16* two regulation are proposed: the first one presents a bandwidth $\omega_\theta = 10^2$ for a robust, bumpless overshoot while the second, tuned for $\omega_\theta = 10^3$ has a more aggressive and ripply response, but a much lower rise time.



(a) $\omega_\theta = 10^2$ **(b)** $\omega_\theta = 10^3$

Figure 4.16 Step response of the position loop for different tuning solutions

For the BWS application, the first solution was preferred for its smoother action on the mechanics and the position proportional controller takes the form:

$$G_{P\_\theta} = 10 \tag{4.39}$$

4.6.2 Numerical approach

The theoretical tuning as above is an efficient solution to achieve good results in simulations for a generic model of the motor. It is simple and repeatable, but its reliability is not sufficient to implement the resulting parameters on a controller for an real system and expecting optimal results.

To properly dimension a regulator for real systems, the recommended procedure is normally experimental. If a very good model of the process is available, though, numerical methods can be exploited for the tuning or, at least, provide a reasonable and reliable start point to initialise the parameters for experimental optimisation. This was the case for the BWS control parameters.

As a second approach, the full model of the system was included in a Matlab transfer function and the corresponding dynamic model was regulated by means of the SISOTOOL suite. To show the difference between the first-order approximation and the actual model, the two different approximation methods for delays (first-order and Padè) are shown in *Fig. 4.17*

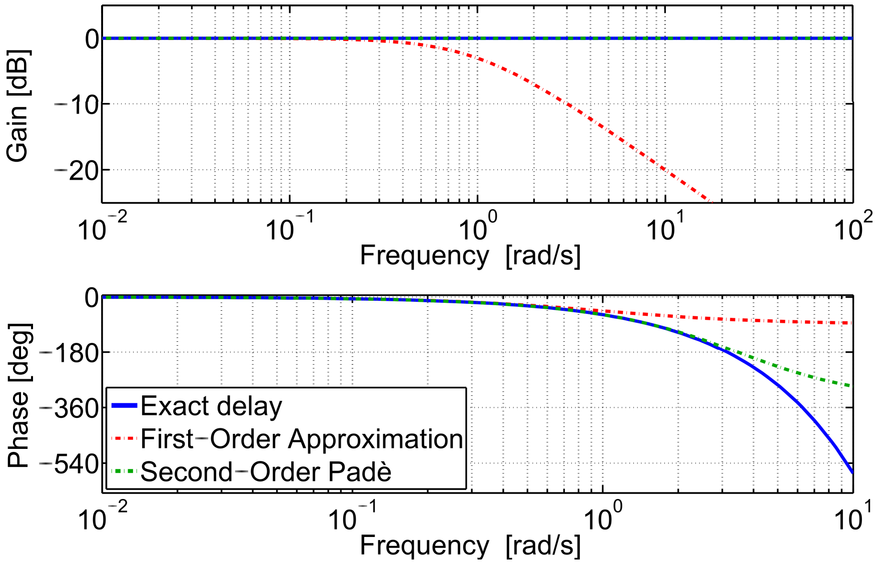


Figure 4.17 Comparison between delay approximation methods

This is just one of the improvements included in the actual model with respect to the previously seen approximation. The model used in SISOTOOL contains the entire model of the system seen in *chapter 3*.

In order to judge the quality of the dynamic response obtained through the use of a certain set of parameters, some preliminary specifications need to be given. For the current controller, the following specifications were considered:

- Steady-state error should be zero
- The rise time should be lower than 1 millisecond, i.e. 16 sampling periods.
- The overshoot should not be higher than 10% to prevent current peaks in the motor windings.
- To guarantee robustness to the system, a phase margin of 40° and a gain margin of 10 dB were introduced.

The rise time specification, in particular, is almost equivalent to the imposition of the given bandwidth in theoretical approach. The difference is that now, apart from having a much more reliable model, we can shape the control loop in deeper details, including performance and stability constraints.

For what concerns the speed loop, different requirements were set:

- Steady-state error should be zero.
- The rise time should be lower than 10 millisecond, i.e. 160 sampling periods.
- The overshoot should not be higher than 20%.
- To guarantee robustness to the system, the same phase margin of 40° and gain margin of 10 dB were considered, as for the previous case.

Finally, the specifications for the position loop were:

- Steady-state error should be zero (it was experimentally verified that this requirement could not be accomplished without an integrator).
- The rise time should be lower than 1 second.
- No overshoot is admitted.
- Same phase and gain margin as for the other two systems.

After the constrained tuning was performed, the parameters were fixed as in *Table 4.2*.

Loop	Proportional gain	Integral gain
Current	0.07	180
Speed	0.05	6
Position	15	8

Table 4.2 Parameters for the PI controllers obtained by numerical tuning

4.6.3 Experimental approach - PSO method

The last optimisation step for the parameters tuning was experimental. Several experimental algorithms were developed for PID tuning, and they can basically be split into two categories:

- Open-loop methods, like the Ziegler-Nichols techniques
- Close-loop methods, like the ATV technique

These solutions have both perks and disadvantages: open-loop techniques are normally more performant, while closed-loop ones can be run without stopping the process from its usual working routine. However, all of the typical methods have the common defect of bringing the system to its stability limit, which is not ideal in the case of elecro-mechanical systems, since the risk of damaging some parts is not admissible.

In order to avoid these issues, an iterative optimising method called *Particle Swarm Optimisation* (PSO) was considered [37]. PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided towards the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

The working principle of the PSO algorithm is described in *Fig. 4.18*.

For this application, the particles are represented by the parameters that are being tuned. This means that the optimisation is running a parametric sweep

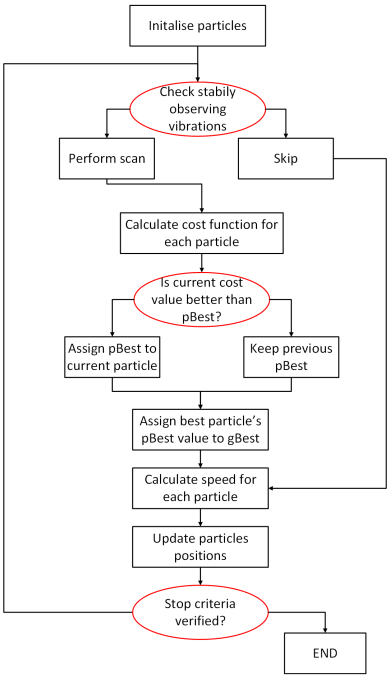

Initalise particles
Check stabily observing vibrations
Perform scan
Skip
Calculate cost function for each particle
Is current cost value better than pBest?
Assign pBest to current particle
Keep previous pBest
Assign best particle's pBest value to gBest
Calculate speed for each particle
Update particles positions
Stop criteria verified?
END

Figure 4.18 PSO method working principle

on a 6-dimensional space, evaluating a cost function which is represented by the IAE computed between the desired motion pattern and the obtained one:

$$IAE = \int |e|dt \tag{4.40}$$

Since the motion pattern consists in three different profiles to be followed, the resulting cost function was chosen a weighted sum of the three values of the costs. Also, as the three profiles contain weights themselves, in order to lead the optimisation algorithm to privilege those parameters that result in a better tracking performance in some zones (*Fig. 4.19*). This complication was introduced in order to minimise the vibrations in the wire during the beam crossing period, since they are amplified by acceleration and speed ripple. The final cost function assumes the form:

$$Cost = \alpha_1 IAE_\theta + \alpha_2 IAE_\omega + \alpha_3 IAE_{Iq} + \beta_1 IAE_{\omega\_1} + \beta_2 IAE_{Iq\_1} + \beta_3 IAE_{Iq\_2} \tag{4.41}$$

where ω_1, Iq_1 and Iq_2 indicate the critic zones of the speed and position profiles, and $\alpha_{1,2,3}$ and $\beta_{1,2,3}$ are weight parameters.

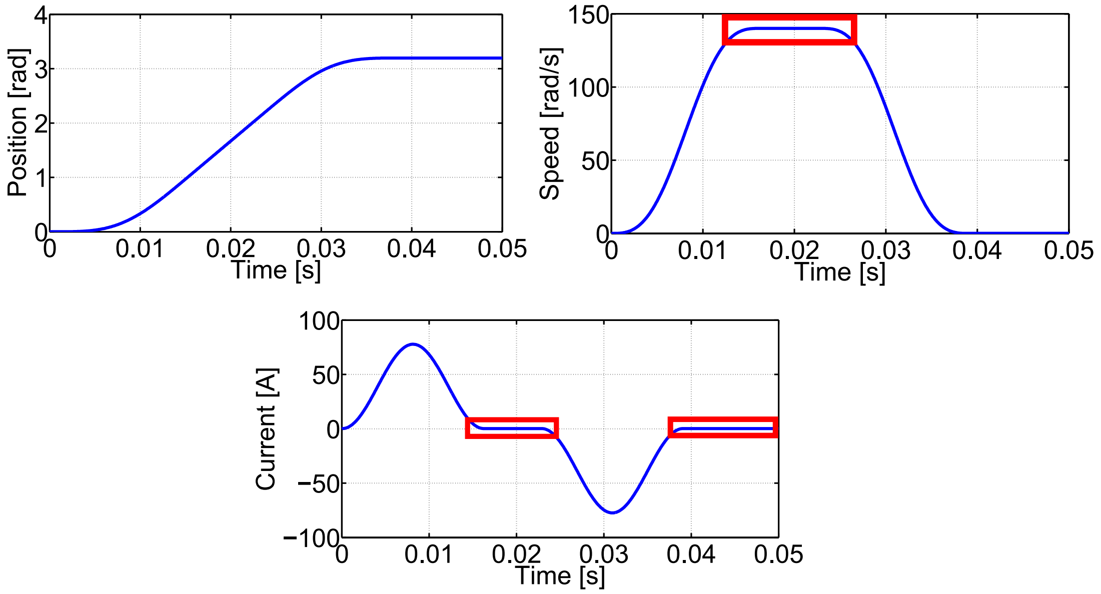


Figure 4.19 Desired profile used for cost function computation with critical zones

The velocity and position of the particle i are updated with the following:

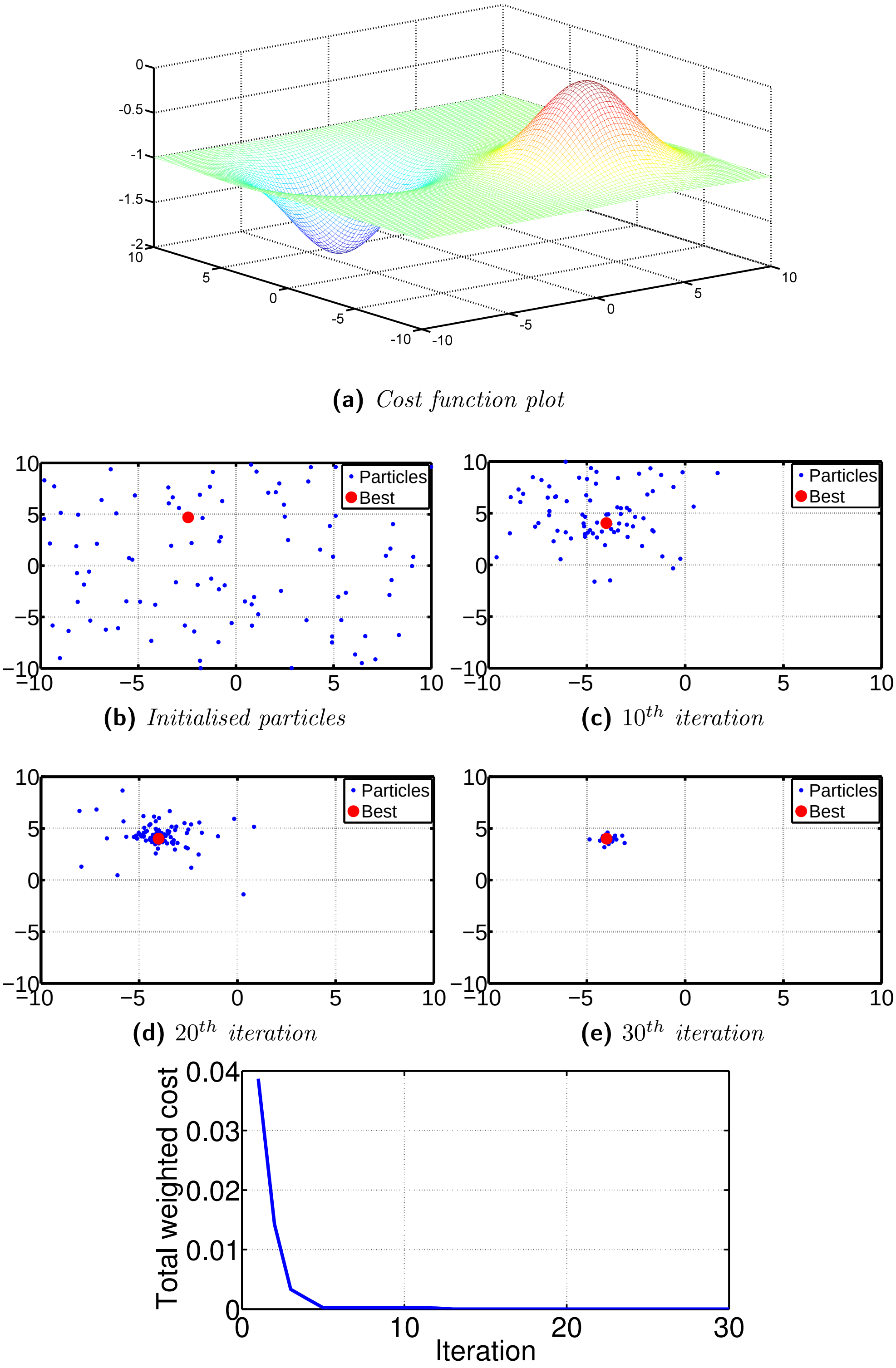
$$v_i^{k+1} = w_i v_i^k + c_1 r_1 (l_{best\_i} - p_i^k) + c_2 r_2 (g_{best} - p_i^k) \tag{4.42}$$

$$p_i^{k+1} = p_i^k + v_i^{k+1} \tag{4.43}$$

where p and v indicate the particle position and speed, w_i is a parameter that, decreasing with the number of iterations, tends to privilege local close solutions over farther ones, c_1 and c_2 are used to set how "explorative" the algorithm will be relating to the initial parameters, r_1 and r_2 are random numbers and l_{best_i} and g_{best} are, respectively, the local and global best solutions so far.

The performances of the particle swarm optimisation method are represented in *Fig. 4.20.*

After the simulation part, the optimisation algorithm was implemented on a dSpace system (*section* 5.2). The particles were initialised with a random variation up to 10% from the nominal parameter value obtained with SISOTOOL. The dSSPACE system uses Python language for control desk automatisation, thus a Python script was implemented to run the optimisation procedure as in *Fig. 4.18*. *Fig. 4.21* shows the user interface of the script. At the end of the tuning routine, a set of optimal parameters is available for the user and can be included in the controller design.



(a) *Cost function plot*

(b) *Initialised particles*

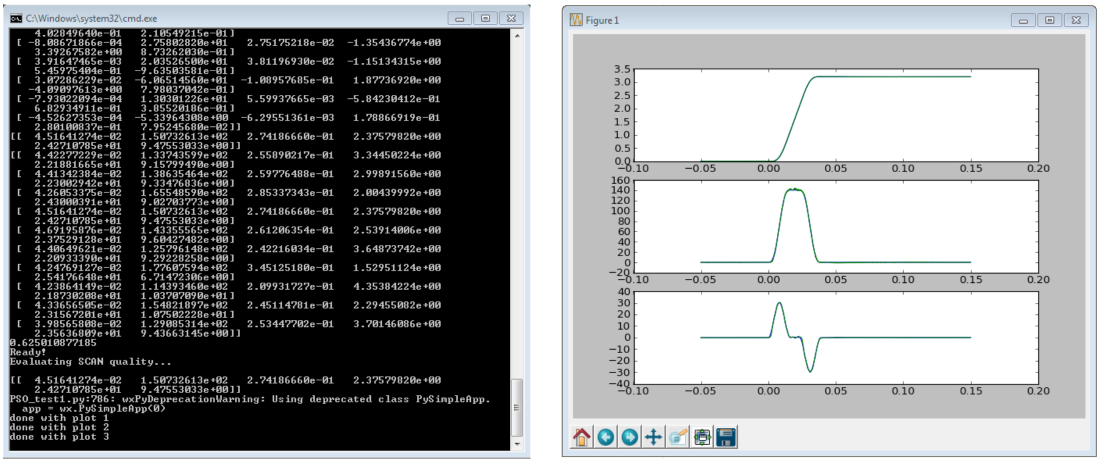
(c) 10^{th} *iteration*

(d) 20^{th} *iteration*

(e) 30^{th} *iteration*

(f) *Cost function evolution vs iterations*

Figure 4.20 PSO method working example in simulations



```
    4.02849640e-01   2.10549215e-01]
 [ -8.08671866e-04   2.75802820e+01   2.75175218e-02  -1.35436774e+00
    3.39267582e+00   8.73262030e-01]
 [  3.91647465e-03   2.03526500e+01   3.81196930e-02  -1.15134315e+00
    5.45975404e-01  -9.63503581e-01]
 [  3.07286229e-02  -6.06514560e+01  -1.08957685e-01   1.87736920e+00
   -4.09097613e+00   7.98037042e-01]
 [ -7.93022094e-04   1.30301226e+01   5.59937665e-03  -5.84230412e-01
    6.82934911e-01   3.85520186e-01]
 [ -4.52627353e-04  -5.33964308e+00  -6.29551361e-03   1.78866919e-01
    2.80100837e-01   7.95245680e-02]]
[[  4.51641274e-02   1.50732613e+02   2.74186660e-01   2.37579820e+00
    2.42710785e+01   9.47553033e+00]]
[[  4.42277229e-02   1.33743599e+02   2.55890217e-01   3.34450224e+00
    2.21881665e+01   9.15799490e+00]
 [  4.41342384e-02   1.38635464e+02   2.59776488e-01   2.99891560e+00
    2.23002942e+01   9.33476836e+00]
 [  4.26053375e-02   1.65548590e+02   2.85337343e-01   2.00439992e+00
    2.43000391e+01   9.02703773e+00]
 [  4.51641274e-02   1.50732613e+02   2.74186660e-01   2.37579820e+00
    2.42710785e+01   9.47553033e+00]
 [  4.69195876e-02   1.43355565e+02   2.61206354e-01   2.53914006e+00
    2.37529128e+01   9.60427482e+00]
 [  4.40649621e-02   1.25796148e+02   2.42216034e-01   3.64873742e+00
    2.20933390e+01   9.29228258e+00]
 [  4.24769127e-02   1.77607594e+02   3.45125180e-01   1.52951124e+00
    2.54176648e+01   6.71472306e+00]
 [  4.23864149e-02   1.14393460e+02   2.09931727e-01   4.35384224e+00
    2.18730208e+01   1.03707090e+01]
 [  4.33656505e-02   1.54821897e+02   2.45114781e-01   2.29455082e+00
    2.31567201e+01   1.07502228e+01]
 [  3.98565808e-02   1.29085314e+02   2.53447702e-01   3.70146086e+00
    2.35636809e+01   9.43663145e+00]]
0.625010877185
Ready!
Evaluating SCAN quality...

[[  4.51641274e-02   1.50732613e+02   2.74186660e-01   2.37579820e+00
    2.42710785e+01   9.47553033e+00]]
PSO_test1.py:786: wxPyDeprecationWarning: Using deprecated class PySimpleApp.
  app = wx.PySimpleApp(0)
done with plot 1
done with plot 2
done with plot 3
```

(a) *Python interactive shell output*

(b) *Graphical output*

Figure 4.21 PSO Python implementation for the dSPACE platform - outputs

An example of output from the PSO algorithm, consisting in the six optimal parameters, is shown in *Table 4.3*. The optimization was then run for different cable lengths and for the different available prototypes of the motor (*chapter 5*).

Loop	Proportional gain	Integral gain
Current	0.045	150.72
Speed	0.27	2.38
Position	24.31	9.47

Table 4.3 Parameters for the PI controllers obtained by numerical tuning

4.7 Feedforward action

The proposed architecture includes 3 cascaded control loops. As already seen in *section 4.1.1*, the dynamic response of one loop influences the behaviour of the other two and vice versa.

Especially, the problem is generated by the position and speed control loops that slow down the inner system, consisting in the current control scheme. This is definitely the major negative effect coming from the use of the resolver: even with the implementation of the SSKF, the bandwidth of the mechanical part is reduced due to the internal compensation of the resolver-to-digital converter.

The current loop, oh the other hand, has a quicker response due to the loop shaping described above, which is possible to achieve because the electrical part is physically faster. This slowdown effect can produce a drastic reduction in the performance of the system: the current will follow a delayed reference producing a wrong amount of torque, resulting finally in a bad tracking of the pre-generated scanning pattern.

A possible observation on this aspect of the regulator is that, being the model of the system already defined and accurately sized, the speed and current slopes are known *a priori* once the position pattern is defined. Even more: the position reference itself was generated starting from the others, thus referring to the model.

A good model of the process to be controlled allows the implementation of a *feedforward* system.

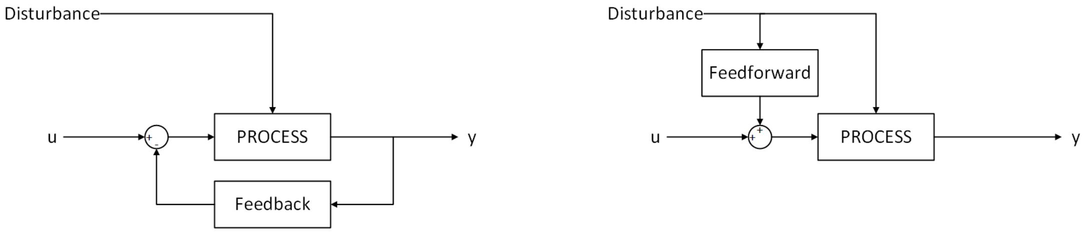


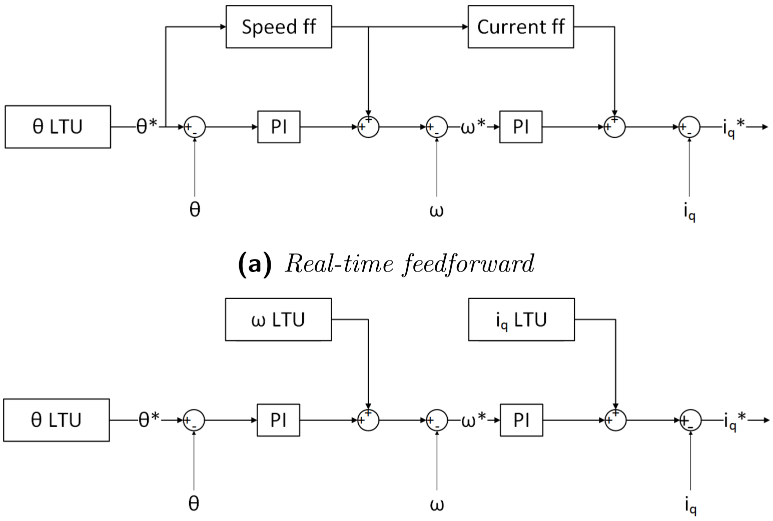
Figure 4.22 Principle of feedback and feedforward action

Feedforward is a popular control technique, and it is complementary to feedback. While feedback is necessary to grant robustness towards disturbances in the control, feedforward is mostly used to improve performance. 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.

There are two ways to include a predictive action in the considered motion controller:

- **Real-time computation:** The profile of speed is generated in real time by deriving the position pattern. Then, the speed is fed into a reversed version of the motor model to compute the current reference. These operations, especially the derivatives, are feasible because the original position profile is well known and obtained from a look up table with arbitrary precision. This solution is more flexible and it is recommended in case of a variable position reference (online generation, adaptive...)

- **Offline computation:** All of the motion profiles are generated in simulation and are already available at the moment of the scan. In this case, the feedforward system consists in three LUTs that are read at the same time.



(a) *Real-time feedforward*

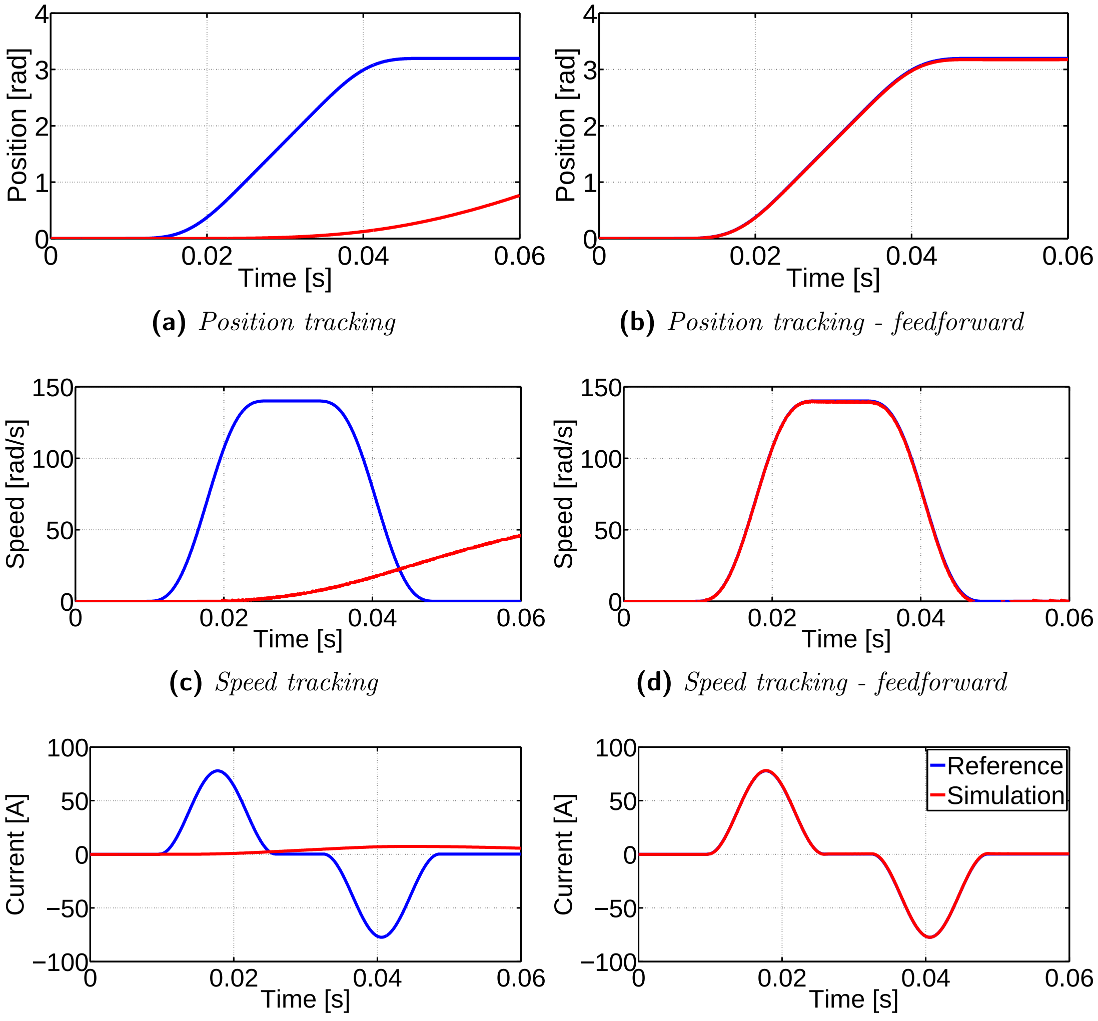
(b) *Offline feedforward*

Figure 4.23 Block diagram of the two investigated anti-windup solutions

The implementation in the regulator of the feedforward loops is actually one of the most important and efficient complications included in the motion controller. The performance increment is evident and can be seen in *Fig. 4.24*. Such a fast and accurate tracking couldn't be accomplished using just a feedback structure.

4.8 Dead time and DC line drops compensation

The power supply system is not ideal. Linear power amplifiers have the defect of introducing slew-rate issues, saturations, and lower efficiency. PWM amplifiers, on the other hand, exploit the fast dynamics of power transistors to reduce time response, but introduce a bigger amount of distortions. Furthermore, a certain quantity of power is routed into harmonics on the switching frequency and its multiples. For these reasons, even using a power filter, the behaviour of a pulse width modulation inverter cannot be approximated with a simple gain.



(a) *Position tracking*

(b) *Position tracking - feedforward*

(c) *Speed tracking*

(d) *Speed tracking - feedforward*

(e) *Current tracking*

(f) *Current tracking - feedforward*

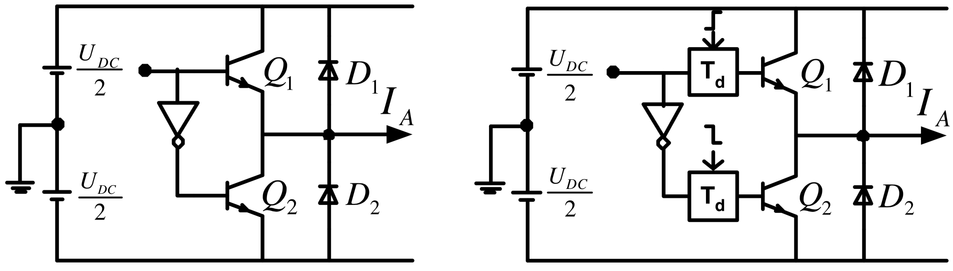
Figure 4.24 Performance comparison of the system with/without feedforward action

The two main effect of non-ideality of the power converter foreseen for the BWS prototype are the following:

- Nonlinearities introduced by the dead-time needed to properly implement the PWM.
- Distortions generated by voltage drops on the power DC line when, during the scanning phase, the capacitors provide extra current on the three phases.

4.8.1 Impact of dead time

In an ideal inverter, the upper and lower switching elements (considered to be ideal switches) are assumed to be changing their state at the same instant. In practice, when power electronic components like thyristors or IGBTs are used as the switching elements, one must always take into consideration the finite time taken by such elements for switching on (turn-on time) and switching off (turn-off time) [38].



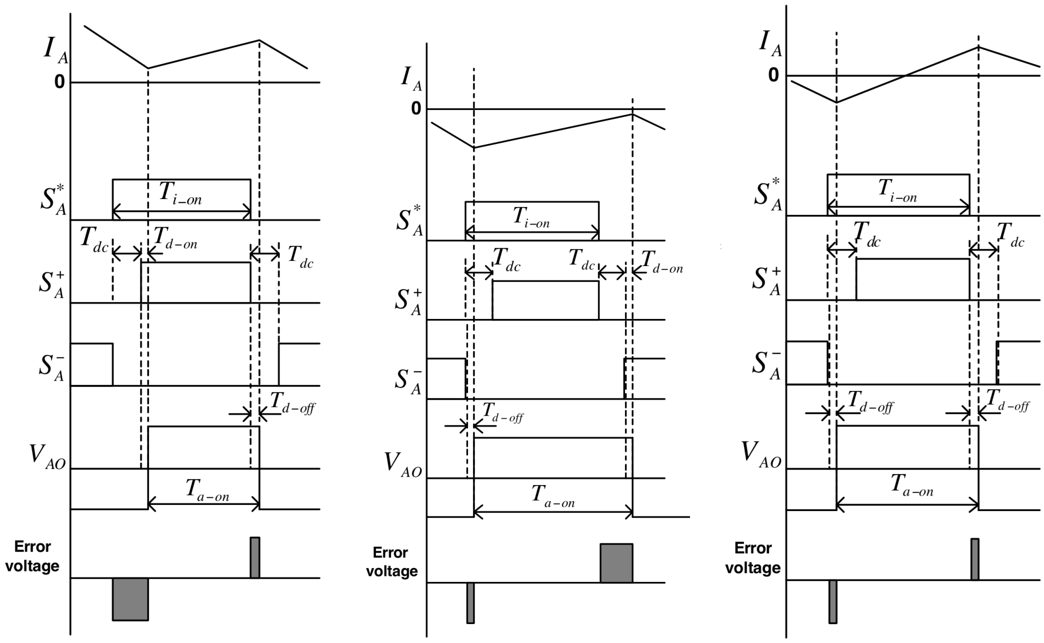
(a) *Without feedforward* **(b)** *With feedforward*

Figure 4.25 1-phase IGBT inverter

At this stage, it is interesting to see the mathematical impact of dead-time on the expected average sinusoidal fundamental voltage from an inverter. When supplying inductive loads, the presence of freewheeling diodes, bypassing the switches as shown in *Fig. 4.25 (b)* makes this analysis a complex task.

(a) $I_A > 0$

This explanation refers to *Fig. 4.26 (a)*. During effective dead-time, Q_1 and Q_2 are non-conducting. A positive I_A can only flow through D_2. Therefore, the node A will continue to be tied to $-U_{DC}/2$ until Q_1 starts conducting. When Q_1 starts conducting, since the node A is connected to $+U_{DC}/2$, the positive current will rise. At the end of the positive pulse, it is now required to switch off Q_1.



(a) *Positive current* **(b)** *Negative current* **(c)** *Switching current*

Figure 4.26 Conducting periods of switches with dead-time

After the switch-off signal is applied to Q_1, the positive current will still continue to increase until it is completely off. Once Q_1 becomes non-conducting, positive current will start flowing through D_2 again and the node A will be effectively connected to $-U_{DC}/2$. Now the actual conducting time due to the applied ideal switch-on signal is given by:

$$T_{a\_on} = T_{i\_on} - T_{dc} - T_{d\_on} + T_{d\_off} = T_{i\_on} - T\_d \tag{4.44}$$

(b) $I_A < 0$

Now, referring to *Fig. 4.26 (b)*, when Q_1 and Q_2 are non-conducting, a negative I_A can only flow through D_1. Therefore, $-U_{DC}/2$ will only appear at node A until Q_2 is effectively switched off. Current will start rising towards zero just as Q1 starts conducting. When the switch-off signal to Q_1 is applied, D_1 will still keep conducting even after the switch off of Q_1. The potential at A will change from $+U_{DC}/2$ to $-U_{DC}/2$, soon after Q_2 starts conducting. The actual conducting time due to the applied ideal switch-on signal in this case is given by:

$$T_{a\_on} = T_{i\_on} + T_{dc} + T_{d\_on} - T_{d\_off} = T_{i\_on} + T_d \tag{4.45}$$

(c) I_A changes sign during pulse

From *Fig. 4.26 (c)* it can be seen that apart from a time shift of T_{d_off} of the positive pulse, the dead-time does not change the pulse width in this case. The following equation can be derived for the actual on time in this case:

$$T_{a\_on} = T_{i\_on} \tag{4.46}$$

This shows that applying dead-time compensation in actual practice has to be totally dependent on the sign of the current in each phase. Since the dead-time has no bad effect, when the phase current is in the vicinity of zero, special care has to be taken on deciding when to apply the particular compensation scheme. With this background knowledge on the behaviour of switching dead-time, the focus will now move on to mathematical description of dead-time.

The same information can be presented in duty cycle form general to the three phases with the following approach [40].

4.8.2 Mathematical analysis of dead time

The aim here is to mathematically describe the effect of dead-time explained above. This will give an idea of how the output voltage in each phase is affected by the presence of switching dead-time. (4.44) and (4.45) can now be rewritten in compact form to give the actual on time of the upper switch of the inverter leg as:

$$T_{a\_on} = T_{i\_on} - T_d\ sgn(I_A) \tag{4.47}$$

According to *Fig. 4.27*, the expression of duty ratio d is:

$$d = \frac{T_{a\_on}}{T_{a\_on} + T_{a\_off}} = \frac{T_{i\_on}}{T_s} \tag{4.48}$$

Dividing (4.47) by $T_{a\_on} + T_{a\_off}$ will lead to a general expression for duty cycle of all three-phases given by:

$$d_k = d_{i,k} - \frac{T_d}{T_s} sgn(I_A) = d_{i,k} - \Delta d_k, \quad k = A, B, C \tag{4.49}$$

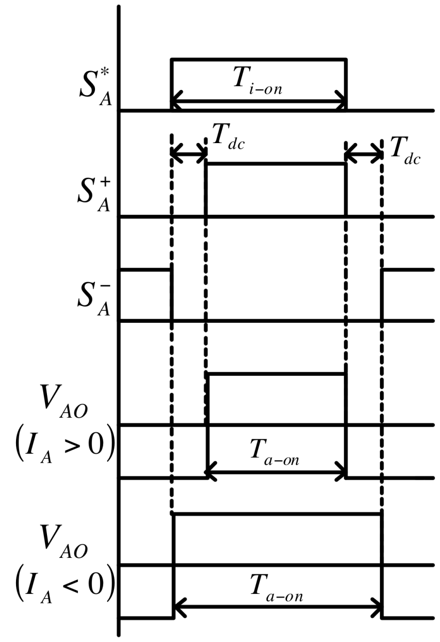


Figure 4.27 Duty cycle form of dead-time effect

The duty cycle error due to dead-time in each phase is given by the term Δd_k, $(k = A, B, C)$ in (4.49). Average phase voltage in each phase (V_{K0}, $k = A, B, C$), over a switching period can now be derived as:

$$V_{K0} = \frac{T_{a\_on} - T_{a\_off}}{T_{a\_on} + T_{a\_off}} \frac{U_{DC}}{2} = (2d_k - 1)\frac{U_{DC}}{2} \tag{4.50}$$

Substituting for d_k from (4.49), the average error voltage in each phase due to the effect of dead-time can be computed as:

$$\Delta V_{K0} = \frac{T_d}{T_s} U_{DC,n}\ sgn(I_k) \quad k = A, B, C \tag{4.51}$$

4.8.3 Impact of DC oscillations

All equations that were considered in this chapter so far assumed the fact that the DC-link voltage is constant (constant U_{DC}). For this to be true, the DC-link voltage has to be infinitely stiff. However, in practice this requires a large DC-link capacitance, which can be bulky and expensive. On the other hand, a large DC-link capacitance may cause much higher harmonic rejection [39]. This means that the DC-link voltage ripple is inevitable in practice.

The ripple frequency and amplitude may depend on the particular used rectifier configuration and the loading condition. Voltage fluctuations in the supply three-phase voltage can also contribute to DC-link voltage fluctuations. As a consequence of DC-link voltage ripple, input voltage of the machine might be influenced causing some low frequency distortions in phase currents. The ripple is most pronounced during large torque reversals of the machine and it can ultimately cause instabilities of the whole drive system [38].

Since the ripple occurs about a certain nominal DC level $U_{DC,n}$ of the bus voltage, the corresponding effect of the fluctuation on each phase voltage can be expressed as a scaling of the expected average voltage over one modulation cycle. This can be given as:

$$V_{K0} = \frac{U_{DC}}{U_{DC,n}} V_{i,K0}, \quad k = A, B, C \tag{4.52}$$

4.8.4 Compensation system

Compensation methods reported so far for the compensation of switching dead-time can basically be divided into two categories:

- Feedback compensation compensates actively the distortions acting on the disturbed signals. A voltage sensor on the various phases is necessary.
- Feedforward compensation is based on the knowledge of the behaviour of the disturb.

The second solution was chosen since no dedicated hardware was included in the electronic part to perform this kind of compensation.

In a PWM generation scheme, where the digital control system generates analog reference levels for PWM patterns, the user always has the ability to make small changes in the amount of modulation in each phase by adding a desired quantity to the reference levels. It is also possible to have prior knowledge of the PWM switching period T_s, effective dead-time T_d, and DC-link voltage U_{DC}. As the sampled line current information is available to the control algorithm, the sign of the current can also be decided. This makes it possible to work out the feed-forward term that must be added to the reference levels of each phase as the compensation term for dead-time distortion. This is a phase-wise compensation

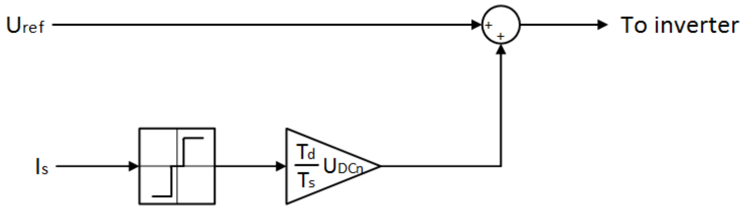


Figure 4.28 Dead time compensation scheme

scheme as it is done in the stator frame for three-phases. The scheme is graphically outlined in *Fig. 4.28*.

Also, being the nominal and the actual measure of the DC voltage ($U_{DC,n}$ and U_{DC}) available, it is possible to perform an analog compensation on the DC line to void the effects of fluctuations and drops. The scheme of this system is available in *Fig. 4.29*.

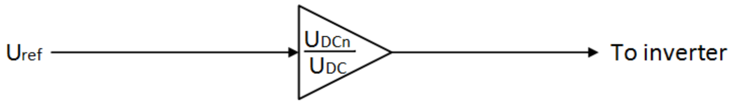


Figure 4.29 DC line fluctuation compensation scheme

4.9 Cable compensation methodology

The system performance and robustness drops related to a variation in the cable length were exhaustively investigated and modelled in *section 3.7.3*. After setting up a reliable model and verifying it, the following two main contributions were identified:

- A low-frequency effect, due to the resistive attenuation behaviour of the cable, that will increase the level of requested voltage to provide the same current to the motor.

- High-frequency nonlinear effects, due to harmonic contributions of the PWM which are not properly attenuated by the power filter.

In the next pages, the compensation methods related to these 2 issue will be described.

4.9.1 Dedicated controller per cable length

The first consequence of a long power line on the voltage supply to the PMSM is the low-frequency attenuation effect due to the non-ideality of the cable itself. It is composed by the sum of two contributions, and it has a high-pass behaviour as in *Fig. 4.30*. To compensate this effect, a model of the cable was extracted for several cable lengths and included in the complete model of the system. It was decided to limit the investigation to 10 different lengths, spanning the worst foreseen scenario (300 m) with steps of 30 m. The bode plots of the derived models are shown in *Fig. 4.30*.

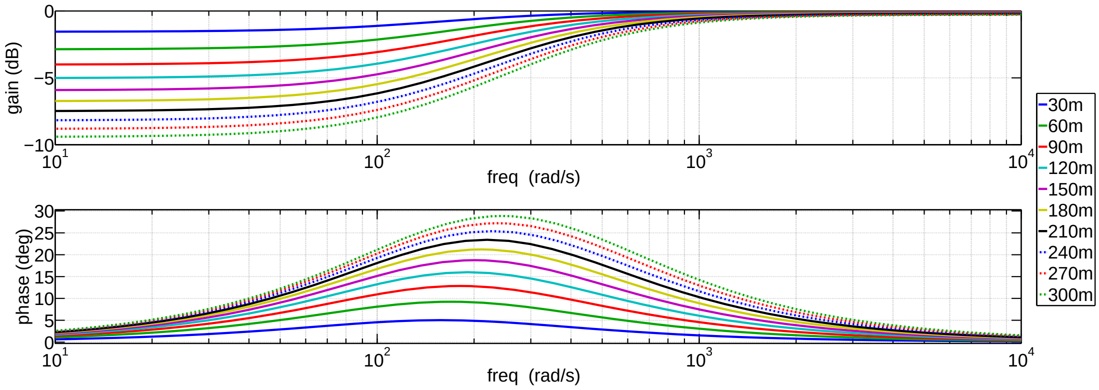


Figure 4.30 Bode plot of cable response for different cable lengths

These contributions were then used to simulate the behaviour of the entire system, including the power line.

The addition of two pairs of singularities in the low frequency spectrum, especially if close to each other, does not influence the dynamic response of the current loop in a relevant way in this case. SISOTOOL shows that the phase margin drop is in the order of 3° for the 300 m cable, which is totally acceptable since the margin was chosen to be reasonably conservative. Thus, a dedicated design for each cable length should not be necessary according to simulations. This result will be verified experimentally testing the system with different cable lengths.

The main effect of the attenuation is not on the transfer function itself: it will influence the system mostly due to saturation. Because of the lower gain at the low frequencies, a higher reference voltage will be requested by the drive to generate the same amount of current. This will ideally increase until the saturation level for the power line is reached, consisting in a bad performance in tracking and other effects on the electronics, including big voltage drops on the DC line.

According to simulations, the SPS wire scanner should not suffer from saturation problems, being driven at a nominal speed of 110 rad/s. This is especially true if the power supply voltage is set to 400 V instead of the nominal 300 V. In the LHC, on the other hand, the simulations show a borderline level of the requested voltage, thus the avoidance of the negative effects of saturation as above cannot be guaranteed. This issue should be solved with the design of a new motor, which will have a higher value of the motor constant K_T, and so will need less current to generate the requested amount of torque.

4.9.2 Variable structure PID

The other effect deriving by the cable concerns the high frequencies and it is more difficult to model and compensate. It is related to the imperfect action of the filter, that will let into the line some power on the PWM harmonics and its multiples. These power peaks at high frequencies affect negatively the stability of the controller, that presented oscillating behaviours when connected to the power line with the 100 m long cable. These effects were mostly observed during the steady-state, rest position of the actuator and less during the scanning action. The problem can be seen in two different ways:

- Due to the amplification of the 16 kHz signal (*Fig. 3.32*), the ringing effect on the switch output increases. Thus, the system will behave as if it was powered with a higher voltage with respect to the nominal tension of the power line. Recalling (4.21), it is clear that the open-loop steady-state gain of the I_q dynamics increases proportionally with the power supply voltage. To maintain a constant bandwidth, the gain of the PI should be reduced (as (4.27) suggests).

- More power on high frequency harmonics can be compared with a higher noise into the control loop. This voltage noise is fed into the motor, producing a noisy current signal. The current signal is then acquired and used to generate the error value which is the input of the PID regulator. A PID regulator in PI configuration is made of a proportional gain plus an integral gain. The integral part naturally filters high frequencies, giving a smooth output, while the proportional one is more sensitive to input noise.

Based on these observations, the solution was found in reducing the proportional gain of the PI controller when the system is in a steady-state situation. This was achieved by complicating the basic scheme of the PI relative to the current loop with a variable structured proportional gain.

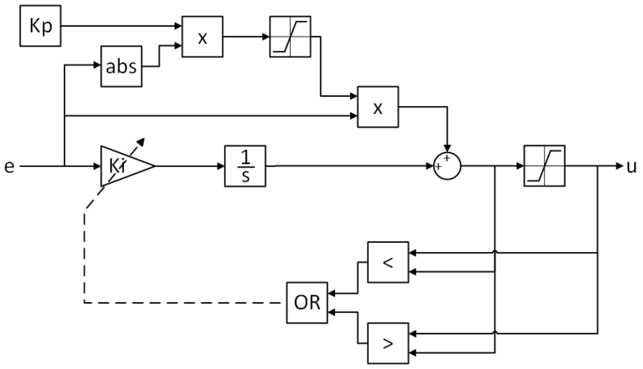


Figure 4.31 Architecture of the variable structure, anti-windup PI controller

The architecture of such a controller is shown in *Fig. 4.31.* This particular solution reduces the K_p to a minimum of 0.3 times its nominal value for errors smaller than 0.3. For bigger errors the gain increases up to its nominal value, which is maintained until the error becomes smaller than 1. Tuning the saturation limits and introducing a gain, one can decide how the gain will vary as a function of the current absolute error.

4.10 Final architecture of the controller

The controller design started from the basic architecture of the FOC, represented in *Fig. 4.1.* After the inclusion of the listed improvements, the final version of the motion control is as in *Fig. 4.32.*

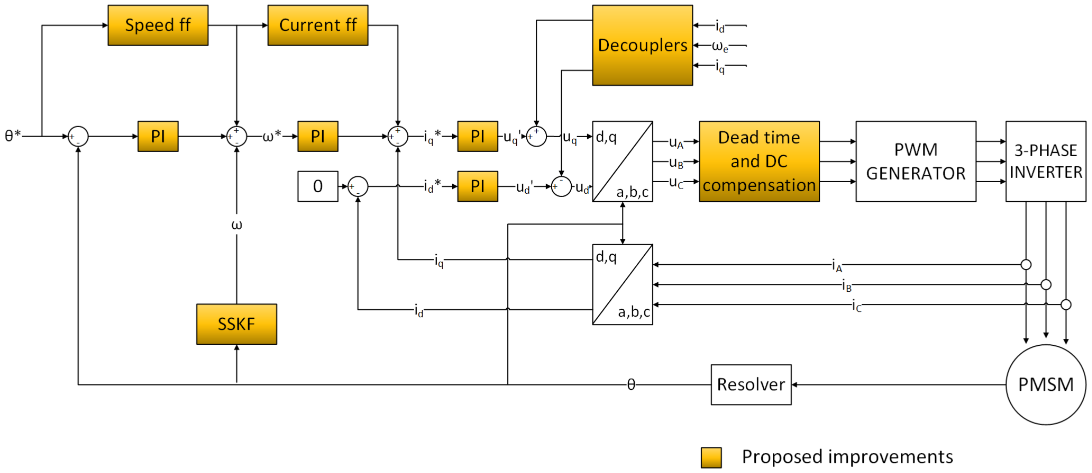


Figure 4.32 Final architecture of the controller, the improved parts are shown in yellow

5. Prototyping controller implementation

This chapter concerns the testbench setup and the experiments run to verify the considered model and the designed controller. To serve this purpose, two different prototypes of the system were realised and installed in the laboratory: a lighter, simplified version and a proper reproduction of the actual Beam Wire Scanner actuator. The other elements used throughout the experiments, like the power supply and the control system, will also be described in their specifications in the following pages. Then, particular attention will be taken in the specification of the dSPACE platform, on which the motion controller was implemented for a preliminary testing before the final implementation on FPGA. Finally, the experimental procedures for verification and tuning will be described in a systematic way, and the obtained results will be compared with the simulations to qualify the final system.

5.1 Testbench description

During the BWS actuator design phase, a whole testbench was realised in the CERN laboratory in order to allow the testing of the proposed system [20]. A schematic drawing of it is shown in *Fig. 5.1*.

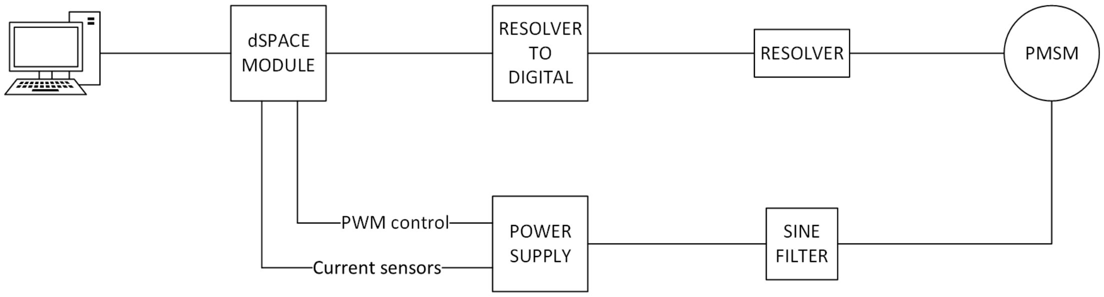


Figure 5.1 Schematic of the lab testbench

Once the design of the regulator was completed, this testbench was verified in its components and used to test it when interfaced with the real system.

The testbench consists in:

- The drive installation, including a mechanical support and the PMSM, fixed on the same shaft together with the position sensor. On the setup are also installed the plugs for signal and power cabling.
- The power supply module, in which the three-phase inverter is installed together with the voltage source, the current and voltage sensors and the power storage capacitors. The power filter is connected between this element and the drive.
- The resolver-to-digital converter, configured as in *section 3.3.4* and used to acquire the analog signal from the resolver.
- The fast prototyping control system, represented by the dSPACE 1103 interfaced with the power supply and to the sensors. It is connected to a PC by means of an optical interface.

The basic prototype of the motor is depicted in *Fig. 5.2*, where the connectors to the resolver and the HEIDENHAIN RON225 high accuracy incremental encoder are shown, as well as power supply interface.

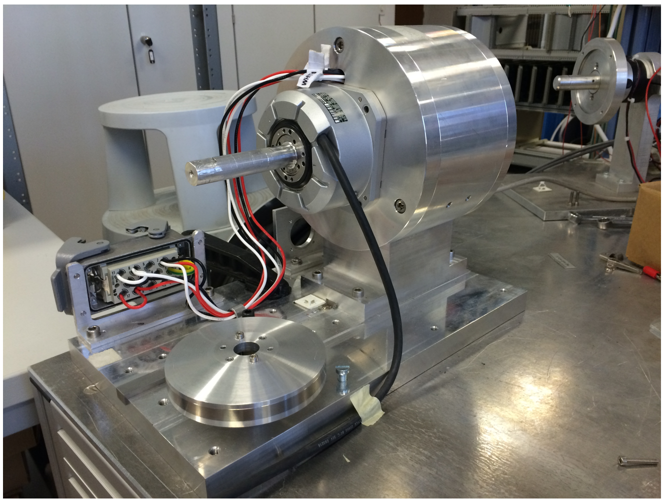


Figure 5.2 Beam Wire Scanner motor bench

Figure 5.3 HEIDENHAIN RON225 high accuracy incremental encoder

The encoder was used to compare the performances of the two different sensors and for the verification of the SSKF behaviour.

Its specifications are listed in *Table 5.1*.

HEIDENHAIN RON225 Encoder parameter	Value
Incremental signals	TTL x 2
Line count	9000
Integrated interpolation	2-fold
Output signals/revolution	18000
Recommended measuring step for position measurement	0.005°
System accuracy	5 arcseconds
Maximum mechanical speed	3000 rpm

Table 5.1 Construction parameters of the HEIDENHAIN RON225 encoder

Fig. 5.4 shows the reproduction of the actual Beam Wire Scanner actuator. The front case contains the resolver, the brake and the PMSM rotors. The relative stators are installed on its external surface. On the other side, the optical encoder disk and its support are mounted.

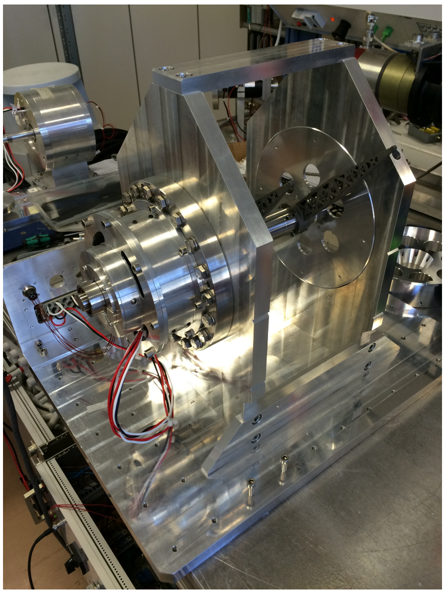


Figure 5.4 Prototype of the Beam Wire Scanner actuator

The power supply case (*Fig. 5.5*) is situated under the laboratory desk along with the power filter, the resolver-to-digital converter board and the dSPACE platform.

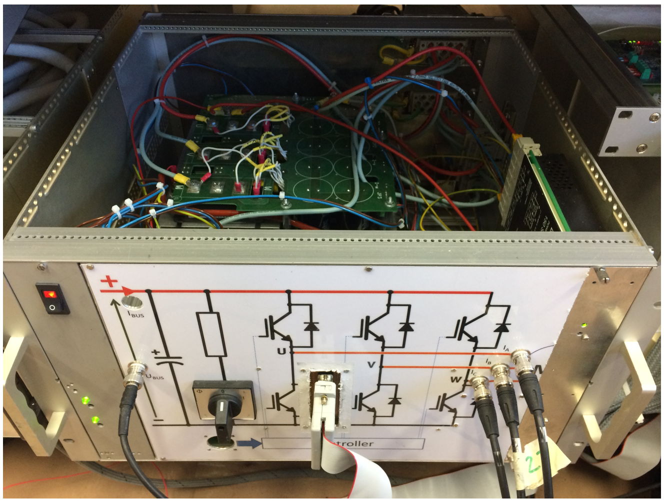


Figure 5.5 Power supply module

5.2 dSPACE platform overview

The first step of implementation of the motion controller was carried out on a prototyping dSPACE DS1103 PPC Controller system. This solution was chosen because the dSPACE DS1103 has a simple and fast programming method through a dedicated Simulink blockset, which allows real-time constraint verification and online diagnostic is handled by a GUI. It owns the following features:

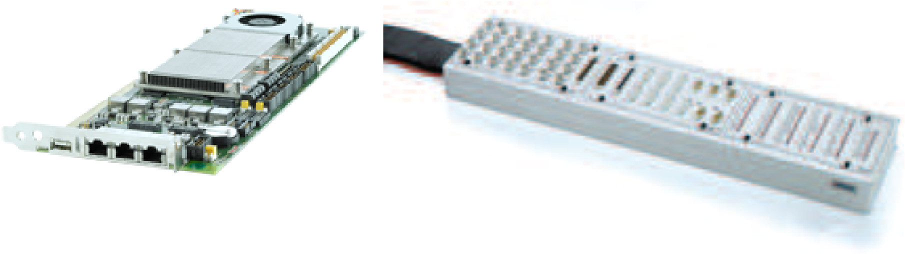
- Designed to be simple and quick to program through a Simulink dedicated blockset.

- Allows the online modification of the parameters.

- Includes a *Real Time Interface* (RTI) to implement control systems in a safe way due to the online check of real-time constraints.

- Has a practical graphic user interface to monitor the design variables, called *ControlDesk*.

- Presents the possibility of performing user-made triggered and continuous data acquisition to be stored directly in several formats, including a Matlab structure.
- The controller actions can be automatised my means of Python scripts.



(a) *DS1103 PPC Controller Board picture* **(b)** *CP1103 Connector and LED panel*

Figure 5.6 dSPACE DS1103 hardware components

5.2.1 Hardware and features

The choice of the DS1103 device was based on some requirements in terms of performance and interface. The internal architecture is shown in *Fig. 5.7*, while its main features are:

- Processor PowerPC (PPC) cadenced at 750 MHz
  - 96 MB Communication SDRAM;
  - ADC, 20 channels, 16-bit;
  - DAC, 8 channels, 16-bit;
  - Incremental Encoder, 7 channels;
  - Digital I/O, 32 channels;
  - Serial Interface RS232/RS422;
  - CAN Interface;
- Slave TMS320F204 DSP

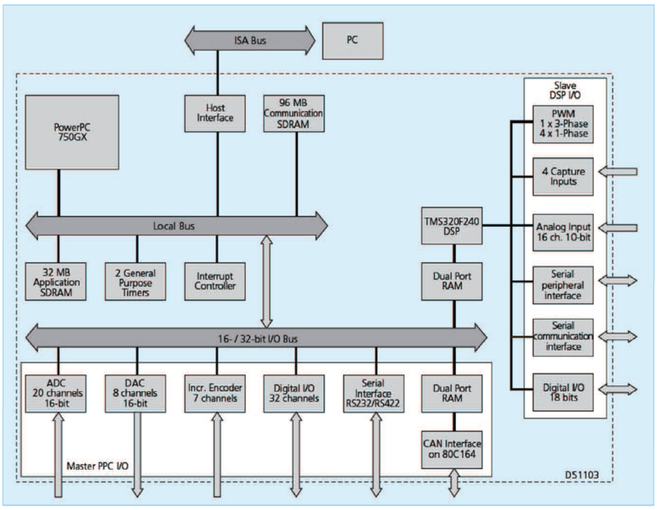


Figure 5.7 Internal architecture of the dSPACE 1103

The hardware-in-the-loop implementation for prototyping included real-time computation of the control variables at a sampling frequency of $16/kHz$. The dSPACE system was able to run it without errors, even being close to its limitations.

5.3 Hardware-in-the-loop improvements

The implementation of a real-time physical controller differs from a simulated one under many practical aspects. There are several complications related to the real world implementation of a regulator of which an engineer has to take care in order to produce a reliable system that will behave, as long as the model is well structured, like the simulated one.

These complications include the real-time constraints, saturations, data format, acquired data preprocessing and more. In the next pages the main problems emerged during the realisation of the motion control system on the dSPACE environment will be listed and the modifications implemented to solve them will be described.

5.3.1 Correction for multiple rotations

The resolver used to sense the shaft angular position is able to provide an absolute measure of the angle, but it gives a periodic measure. In other words, it cannot distinguish if a full turn was travelled and it will start over giving a 0° output after a displacement of 360°. The BWS has two main functions: scanning and approaching. During the scan, the total angular displacement of the shaft will be (ideally) exactly $\pi\ rad$. After the crossing point the motor will start decelerating, tracking the given profile and stopping in the second "home" position, after half a turn. Due to physical constraints, the beam wire scanner will in principle not be able to perform a full turn, at least avoiding damaging the wire. According to these specifications, it would not be necessary to correct the position readout system to distinguish when a full rotation was performed.

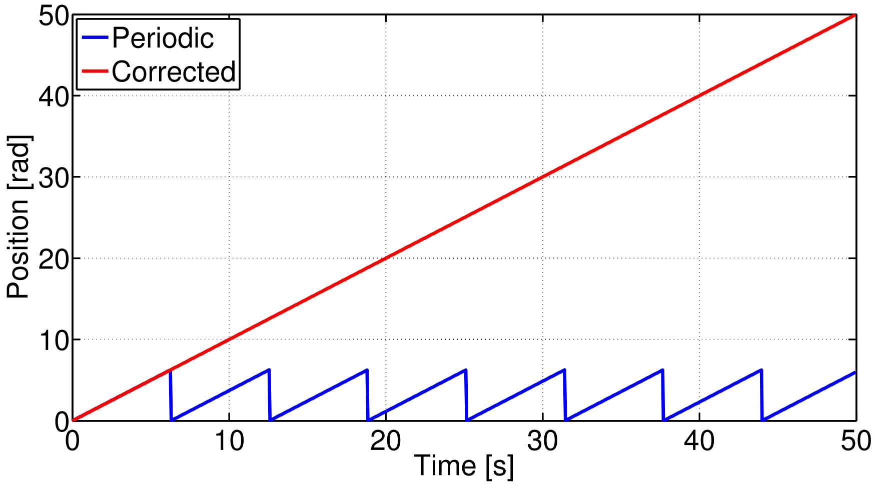


Figure 5.8 Response of a periodic position sensor during constant speed turning

Nevertheless, for the testing phase and to guarantee robustness to the system, it is always a good use to allow the motor to travel more than one full turn around its axis. This is especially recommended taking into account the fact that, in a closed-loop system, an periodic angular step of 2π in the measure can drive the system to diverge even if all the internal responses are stable.

This issue can be easily solved: it is sufficient to identify the moment when the 0° crossing occurs, and correct the measure adding or subtracting 2π, depending on the turning direction. To recognise when to correct the measure, the discrete derivative of the system was used. Since the sampling frequency is given, we can write:

$$\omega_{cross} = \frac{\Delta\theta}{\Delta t} = \frac{2\pi}{T_s} = 1.03 \times 10^5 \frac{rad}{s}$$

which is far higher than the mechanical speed achievable by the considered

PMSM, rated around 670 rad/s. Choosing a reasonable threshold, settled on $Th = 2000\ rad/s$, a system was implemented in order to identify when the measured speed corresponds to a 0° crossing and accordingly correct the position output. A schematic representation of this system is given in *Fig. 5.9*.

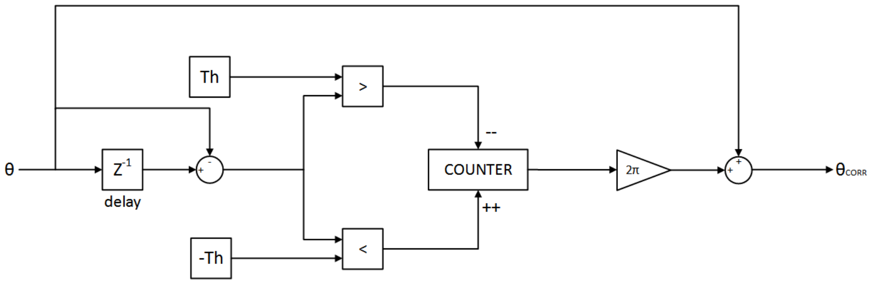


Figure 5.9 Schematic of the full rotation compensation method

5.3.2 PID integral action reset

The motion controller was implemented to behave correctly for several modes of operation, including the scanning and approaching functions, a lock state in which it maintains a given position, a constant speed state used for robustness test, and a turn-off state in which the PWM signals are all kept in a constant low level. Also, this idle state is the one used for the controller initialisation. An issue could rise when the system is turned on activating the PWM generator: since the PID regulators are active even when the motion controller is in its off state, they will keep integrating the errors present in their inputs. These errors can have a high value, depending on the situation, which will obviously become higher and higher as time passes, due to integral action. When the system is switched from its idle state to a working state, the inner control chain will already be in a saturation level, or will anyway request a big amount of torque to be produced in the motor. This will cause an impulsive current to flow into the stator windings. Such a "power kick" could damage the motor in both its electrical al mechanical components, and need to be prevented. Furthermore, recovering from this effect could take a long time before a useful working routine can be achieved again.

Since the problem is basically caused by the integral action of the PID controllers, the proposed solution consists in a complication of the PID architecture including an externally resettable integrator block. The reset is synchronised with the PWM activation signal, so that when the motion controller is switched on, the integral registers are set to zero and the current pulse is avoided. This architecture appeared to solve the problem entirely, and its implementation schematics is shown in *Fig. 5.10*.

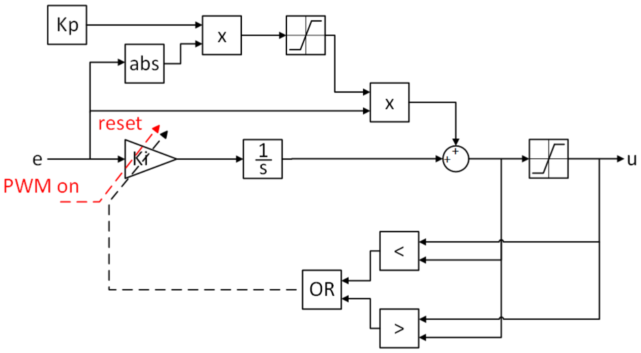


Figure 5.10 Schematic of the conditioned Anti-Windup, variable structure PID controller with external reset

5.3.3 Safety measures

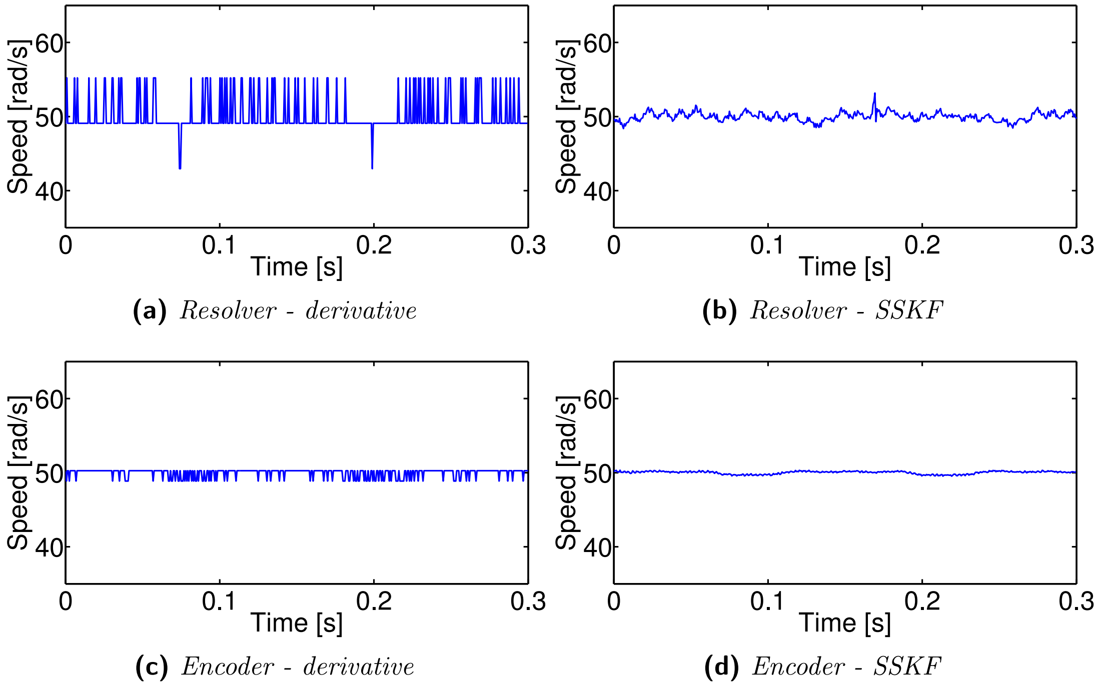
Some security measures were included in the implementation of the regulator for the BWS actuator prototype. This electro-mechanical device need a high reliability level, for it is supposed to keep working with nominal performance for several hundreds of thousands of scans. In the previous systems, a safety function was repositioning the motor at "home" position when the error became bigger than a certain threshold, but didn't include countermeasures about loss of control. In order to avoid issues coming from a possible system instability, a maintenance block was realised to stop the PWM generation from working in case the shaft rotates too far from its home position. The bounds for its proper rotation were limited to the set $[-2\pi, 2\pi]$. This function is can be turned off when multiple rotations are needed.

5.4 Experimental results

The final step of verification for the developed model and the control system is the comparison between the results obtained by the simulations and the corresponding experimental data.

5.5 Kalman filter test

Before starting with the actual controller qualification, the SSKF was tested for a constant speed control at 50 rad/s. The experiment consisted in a comparison between the performances of the encoder and the resolver, using both a simple discrete derivative and the Kalman filter proposed in *section 4.5.2*. The results, plotted in *Fig. 5.11*, show a good performance of the SSKF, capable to overperform the discrete derivative approach. When implemented on the resolver, the experiment shows results comparable with the encoder output when this improved architecture is not used.



(a) *Resolver - derivative*

(b) *Resolver - SSKF*

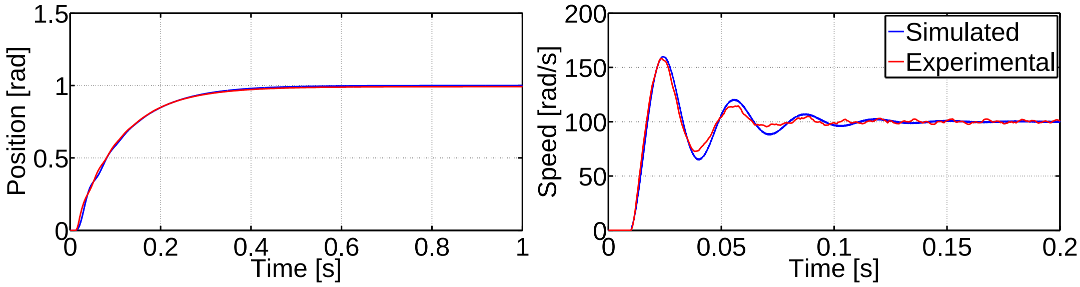
(c) *Encoder - derivative*

(d) *Encoder - SSKF*

Figure 5.11 Performance evaluation of the SSKF

5.6 Validation of the model for lab motor bench

As a first experiment on the controller, mostly useful for the model validation, a step reference signal was given to the motor in both the position and the speed variables. For this test the parameters derived with the theoretical approach were used. The compared results are plotted in *Fig. 5.12*, where a good fit between the simulated and the real system is shown.



(a) *Position response* **(b)** *Speed response*

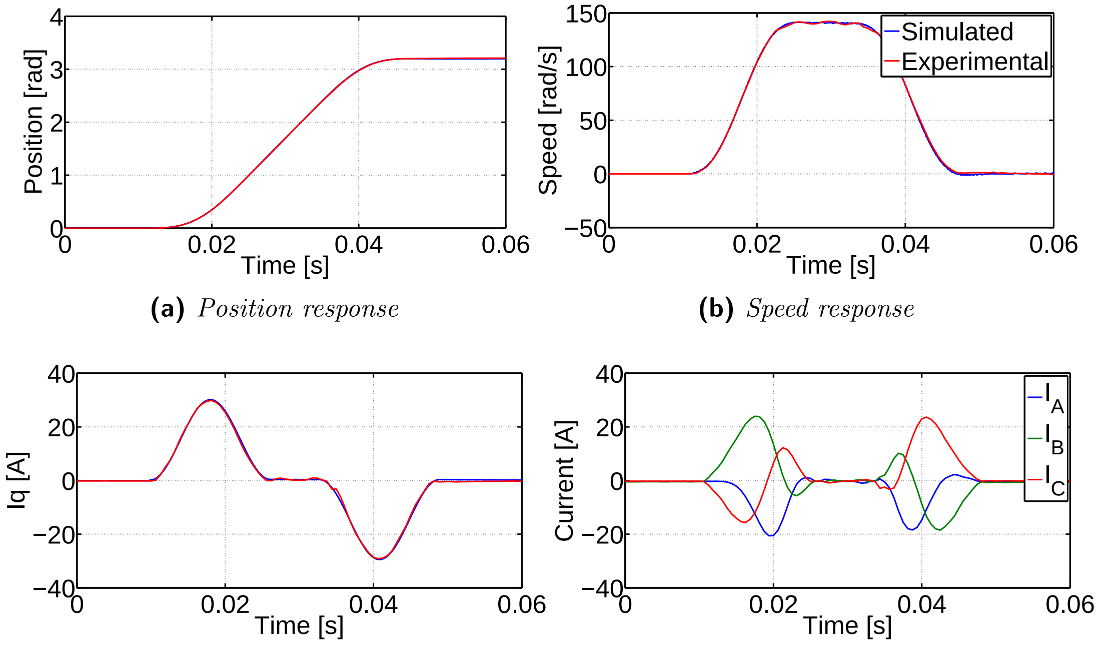
Figure 5.12 Validation of the motor bench model - step response

5.7 Motor bench - scan performance

Once the model for the system has been validated, the goal of the experiments is to qualify the controller performances during the actual scanning operation, which is the main function provided by the BWS. Some plots of scans using the final version of the controller can be seen in *Fig. 5.13* for 140 rad/s and *Fig. 5.14* for a maximums speed of 200 rad/s. Tests were also performed for a speed of 110 rad/s, but are not shown because they are are equivalent to the 140 rad/s ones.

For the profile using low speed, we can see that the performances are quite good: the system is stable and has low oscillations, the various profiles are well tracked and they are consistent to each other. In other words, the computed current profile generates exactly the desired speed profile, so that finally both of them have the same slope as in the simulations, as well as the position plot.

When a higher-speed scan is performed, as in *Fig. 5.14*, some differences are appearing between the simulated and the experimentally achieved profiles: even though the current plots look the same, the actual speed is lower than expected, especially at the end of the acceleration. It then tends to flatten on the desired scanning speed, before starting decreasing again due to the negative reference current.



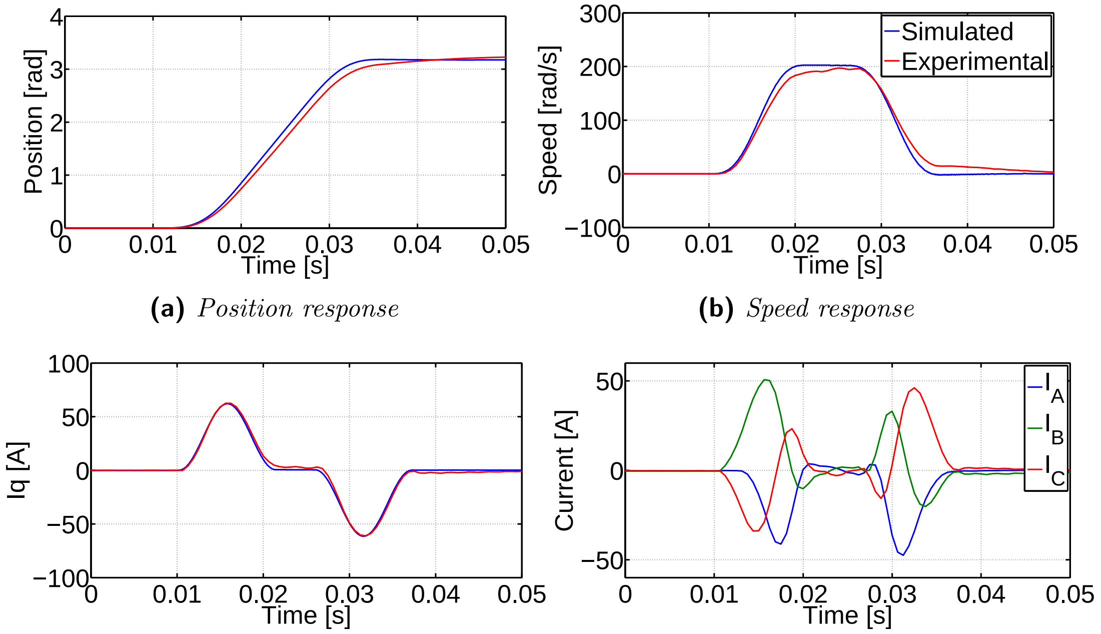
(a) *Position response*

(b) *Speed response*

(c) *Current response*

(d) *Three-phase current response*

Figure 5.13 Experiment of scan for a top speed of 140 rad/s - motor bench



(a) *Position response*

(b) *Speed response*

(c) *Current response*

(d) *Three-phase current response*

Figure 5.14 Experiment of scan for a top speed of 200 rad/s - motor bench

5.7.1 Nonlinearities in the torque constant

Recalling (3.12), we can write:

$$T(i_q) = \frac{3}{2} p \Phi_f i_q = K_T i_q$$

where the produced torque is - ideally - proportional to the vector sum of the currents into the statoric windings in the direction normal to the flux generated by the rotor, previously called i_q. This linear proportionality is in fact just an approximation: while it is reliable for a low amount of current or torque, it was experimentally verified to be untrue when a higher value of current is applied to the motor.

To better show this nonlinear effect, we will refer to the experimental results shown in *Fig. 5.14*. Here, it is clear that the provided current is not sufficient to achieve the desired final speed, that is, the produced acceleration was smaller than expected. In this situation, considering that the used model is the same, the hypothesis is that something might have changed in the parameters of the system. There could be two posible causes to such an incoherence:

- An increase of the inertia of the system, J.
- A decrease of the torque constant, K_T.

Since the inertia is a physical property of the device and since it is not supposed to change during a scan, the cause of the observed effect can be most likely imputed to nonlinear effects in the torque constant. The described effect could be due to saturation in the ferromagnetic materials, but this performance loss starts before the limits specified by the supplier and it is not described by dedicated graphs or equations. Thus, an experimental approach was used in order to quantify it.

The experiment consisted in performing a current control following the standard scheme of the *Direct Torque Control* (DTC, [42] and [43]). In this configuration, the system is driven to follow a certain current value. Using a step function as a reference and varying its value, the motor will theoretically receive a constant acceleration proportional to the current. Unfortunately, the acceleration cannot be kept constant for a long time due to damping effects and speed saturation due to back-EMF. The measure needs to be performed in a short time interval, and this was achieved by using the incremental encoder with and improved efficiency granted by the SSKF. The speed signal was then derived performing a discrete derivative and smoothed by means of a FIR digital filter. Finally, the value of the torque was obtained by scaling the measured signal and correcting it with the friction and damping contributions:

$$T = J\alpha_m - k_f sgn(\omega_m) - B\omega_m$$

This method is surely not the ideal way to measure the torque produced by an electric motor since it is not very precise, but it gave relevant and feasible results, at least until a certain amount of current. By correlating the value of the i_q with the achieved torque in a plot, we obtain *Fig. 5.15*.

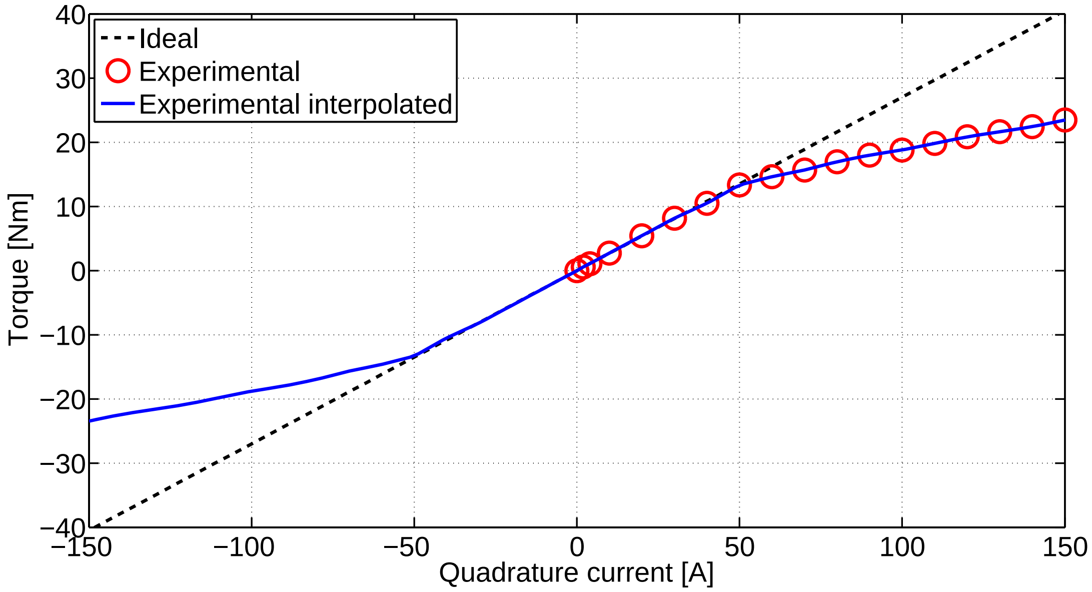
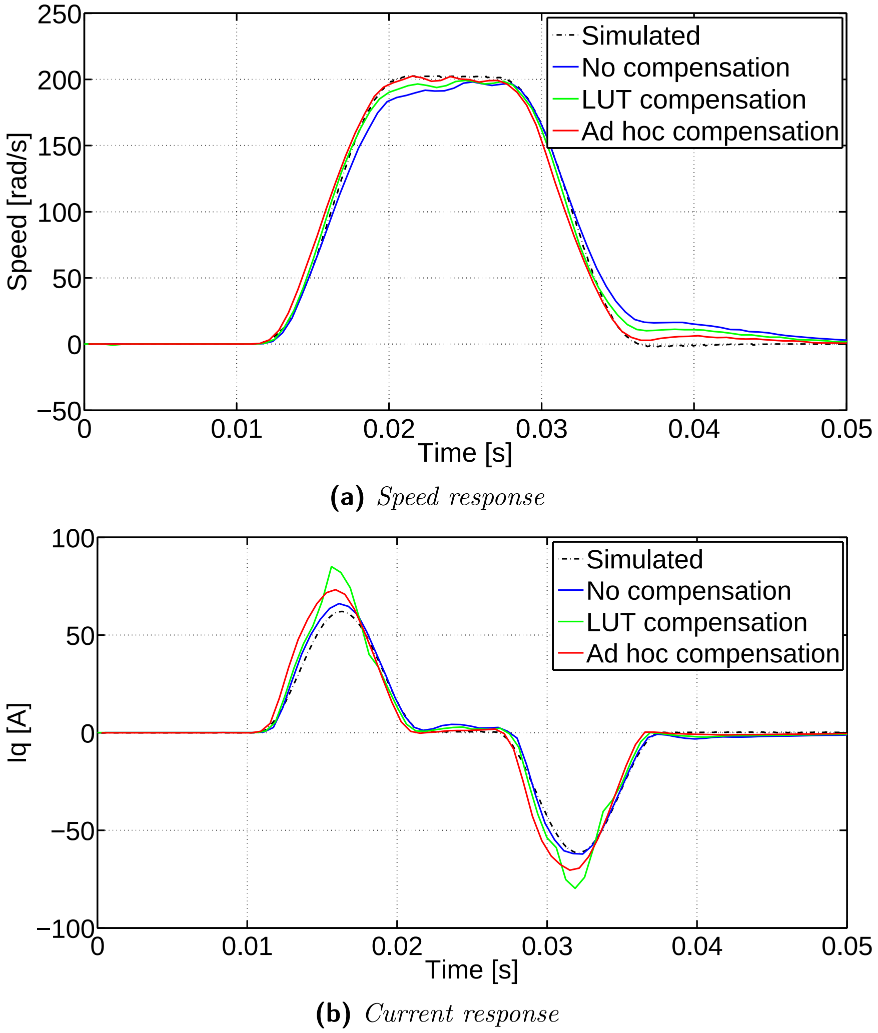


Figure 5.15 Experimental current vs torque plot

The goal of the experiment was to show that the relationship between current and torque is not linear over a certain value, which is in fact demonstrated by *Fig. 5.15*. The obtained correlation was then put into a look-up table and used to correct on-line the current reference given by the feedforward block.

A different proposed compensation method was an "ad hoc" correction of the K_T parameter to input to the controller. If a lower value for the torque constant is taken into account, the motion controller will behave as if it was driving a weaker motor, thus giving more current to the PMSM. In our case, the correction was tuned such that the scanning speed was equal to the requested one.

Fig. 5.16 shows compared results for the two proposed methods. It is important to remark that the signal that should be properly followed is the speed response. The current response, being subject to nonlinearities in this case, will necessarily differ from the simulations. The "ad hoc" compensation seems to be more efficient, both for the tracking of the final speeds and for the lower peak current produced. A drawback is the difference in the speed profiles: if a constant K_T is considered, than the acceleration will be higher than expected for the first period, then lower. This is not a particular issue in our case. Furthermore, since only a limited number of scan profiles is considered, a dedicated correction for each of them can be computed to cover all the possible configurations.



(a) *Speed response*

(b) *Current response*

Figure 5.16 Comparison between the proposed correcting methods for torque nonlinearities

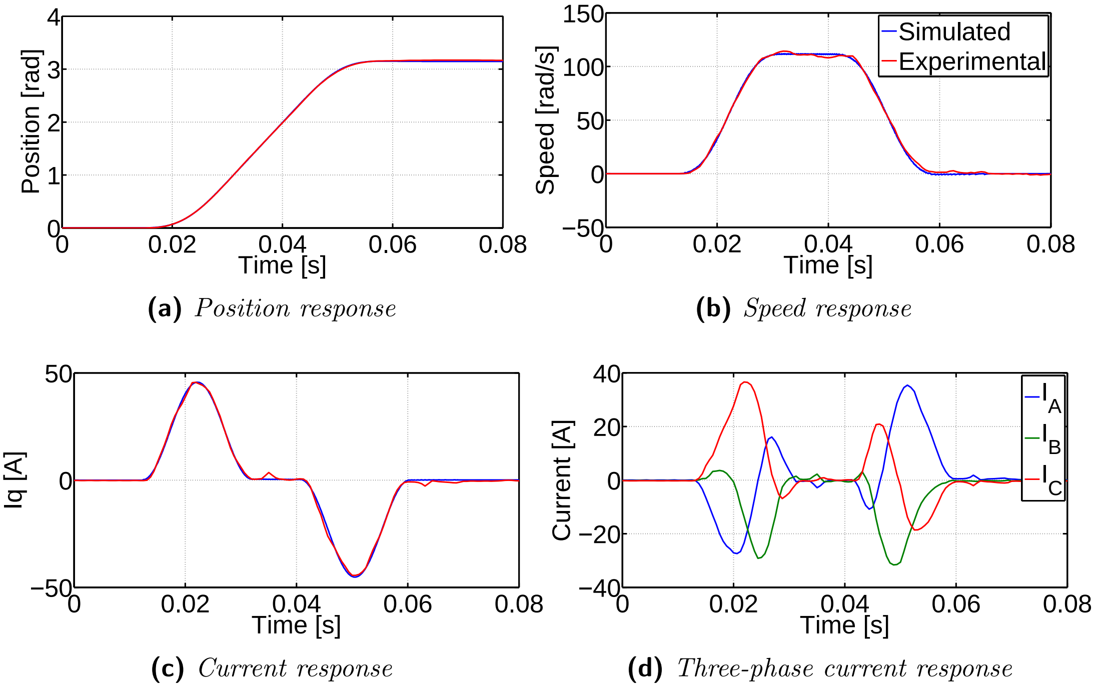
5.8 BWS prototype - scan performance

The last verification step for the motion controller is the qualification of the actual system performances in scanning mode. For the testing, the beam wire scanner actuator prototype was used. The differences of this controller with respect to the one used with the motor bench can be summarised in the following points:

- Due to the different placement of the resolver, the sign of the acquired position is reversed.
- Different values for the inertia and for the torque constant are taken into account, as well as different friction and damping coefficients.
- Feedforward corrections for the effects of brake and gravity (only for vertical scanners) are included.

Fig. 5.17 and *Fig. 5.18* show the comparison between the simulations and the actual performances of the device for scanning speeds equals to 110 rad/s and 140 rad/s. Nominal peak speed of 200 rad/s, foreseen for the machines mounted in the LHC, could not be achieved due to the insufficient torque of the provided motor. It is useful to notice that the configuration of the actuator for the first session of this experiment (BWS actuator reproduction, scanning speed of 110 rad/s) represents the nominal working conditions of the beam wire scanner in the SPS accelerator.

The shown experimental data were collected with the system working in its nominal configuration, except for the optical disk which was not installed because not available. The performances are not as smooth as for the previous experiments due to the imperfect correction of the nonlinear effect, such that the brake. An experimental verification of the brake model is foreseen as a future improvement. All the plots represent the BWS actuator powered through a 10 m cable. Analogue tests for the 100 m version were performed but not shown since they entirely overlap on the presented ones.



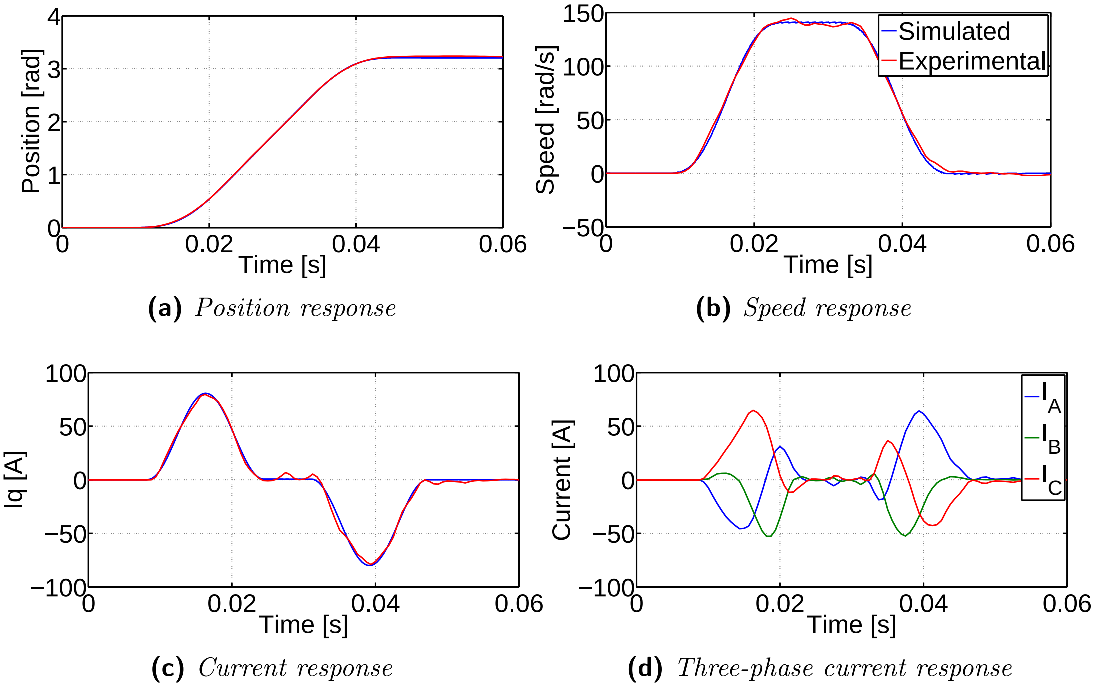
(a) *Position response*

(b) *Speed response*

(c) *Current response*

(d) *Three-phase current response*

Figure 5.17 Experiment of scan for a top speed of 110 rad/s - BWS prototype



(a) *Position response*

(b) *Speed response*

(c) *Current response*

(d) *Three-phase current response*

Figure 5.18 Experiment of scan for a top speed of 140 rad/s - BWS prototype

6. Implementation on FPGA

The last part of this thesis work concerns the realisation of the proposed controller on a ALTERA Arria V FPGA board. The implementation on programmable hardware was chosen for the high speed performance due to parallel architecture, for its reliability and to fit with the CERN standards. Nevertheless, it implies a certain number of disadvantages, like a more complicated firmware development and debug phase, or the necessity of adopting a fixed-point arithmetic to have a full hardware implementation, compromising from the precision point of view. Floating-point libraries and IPs are available, but much more demanding in hardware resources. In this chapter, the structure of the controller realised on FPGA will be explored in its main components and hardware blocks. Together with the logic and arithmetic part, ported on a fixed-point version, the interface and diagnostic blocks will be described to provide a wider understanding of the overall system. The interface part was necessary to properly connect the Arria V with the rest of the hardware - sensors, ADCs... - and exchange data with it. Diagnostic blocks were implemented for data acquisition and monitoring during the scanning action, and were also necessary for the debug. To validate the design, all of the subcomponents were simulated independently, before being put together and perform the final simulations and hardware test.

6.1 Field-Programmable Gate Arrays (FPGA)

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of *configurable logic blocks* (CLBs) connected via programmable interconnections. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from *Application Specific Integrated Circuits* (ASICs), which are custom manufactured for specific design tasks. The main advantage of the FPGA solution over the design of a dedicated hardware is the possibility of changing its configuration and improve/extend its functionalities when a bug is detected or when, with time, they become obsolete.

The FPGA configuration is generally specified using a *hardware description language* (HDL). The most diffused are VHDL and Verilog. For this work - and, generally, in the CERN environment - VHDL was preferred, so the whole firmware is based on this specific language.

6.2 Arria V DevKit overview

The *ALTERA Arria V SoC Development Kit* (*Fig. 6.1*) is a development kit provided by ALTERA embedding its midrange FPGA board, the Arria V.

Apart for the FPGA, it contains several high-speed external I/O and memory interfaces, some devices useful for control and debug such as LEDs and buttons, USB ports for JTAG connection and programming, and a dedicated *Hard Processor System* (HPS), consisting in a Highly Integrated ARM Cortex A9. Some of its specifications are listed in *Table 6.1.*

The choice of this device could appear oversized for the control application described in this thesis work, but its firmware will not contain only that. The motion controller for the BWS will take just a part of the available resources in the FPGA board.

The rest will be used to implement several complementary operations (shown in *Fig. 6.2*) such as:

- High speed connectivity
- Scanner monitoring and diagnostic
- Data acquisition

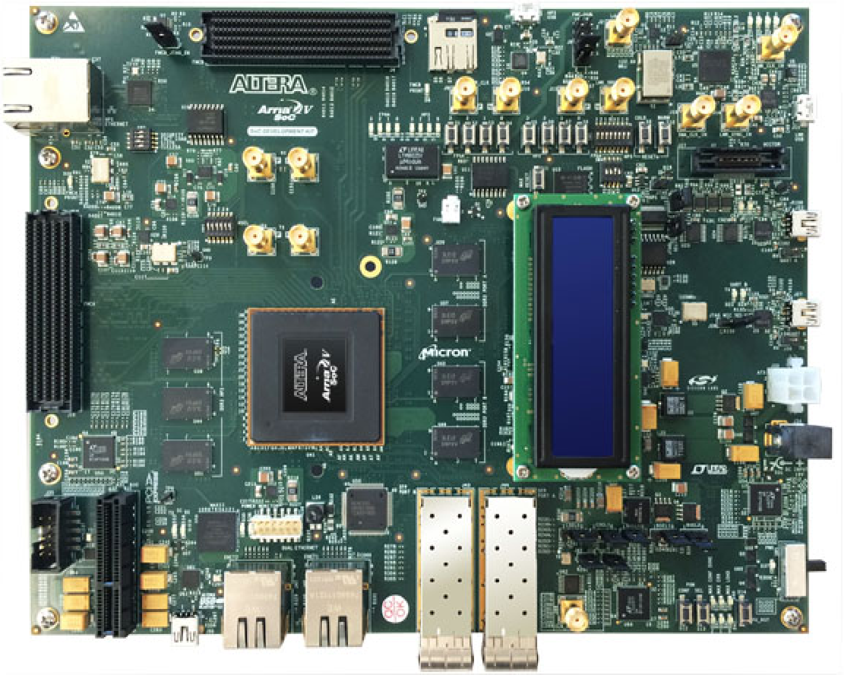


Figure 6.1 The ALTERA Arria V SoC Development Kit

Arria V 5ASTFD5K3F40I3
360K logic elements (LEs)
240 10Gbos tranceivers
C5 speed grade
Three I/O expansion slots
2x 1,024 MB 32-bit DDR3 SDRAM
SFP+ connections
Phase-locked loops (PLLs)

Table 6.1 ALTERA Arria V characteristics

In the following pages, this latter aspects of the FPGA firmware will not be described in details, while more importance will be given to the regulator realisation.

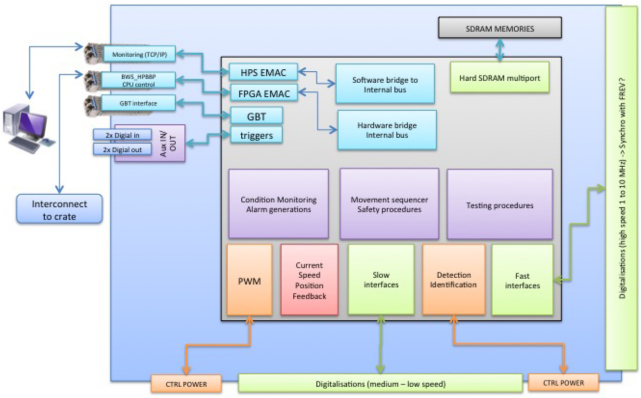


Figure 6.2 Full block diagram of the FPGA control board

6.2.1 Finite precision design

One of the main advantages of hardware programming is the possibility to design a logic system in its details using a representation which is as close as possible to the actual hardware, even routing the designed signals into the specific logic gates, if desired. On the other hand, this kind of signal representation is not as convenient when arithmetic operations need to be performed. Typical processors are more suitable to mathematics due to floating-point ALUs that "mask" the actual arithmetic to the user, performing even complicated math operations with a serial approach thanks to pipelined architecture. This architecture is hard to implement on a FPGA core, and not entirely useful in our case, since the mathematical operations performed by the motion controller are hardly much more complicated than divisions.

The typical solution in hardware programming is a dedicated architecture based on fixed-point arithmetic. Fixed-point arithmetic provides a lower accuracy in mathematical computation, but has the advantage of being much easier to implement since it is basically equivalent to integer arithmetic. The numeric representation can vary for every particular design, accordingly with the application requirements, and different fixed-point representations can also be implemented in the same architecture, if the precision requirements are mutable. The depths of the integer and fractional part of the used arithmetic are crucial design choices. They can generate relevant issues in the system behaviour, like unacceptable precision losses or saturation effects.

A simulation was set up on MATLAB to evaluate the performance drop due to the quantisation and the fixed-point representation of the variables. The performance of the quantised controller are compared to the nominal version using the same cost function used to tune the PID parameters, thus based on a miscellaneous contribution from the three profiles. The results are plotted in *Fig. 6.3*. It is shown how a lower-precision arithmetic would provoke a visible performance loss, while a the drop becomes less significant when using more than 30 bits.

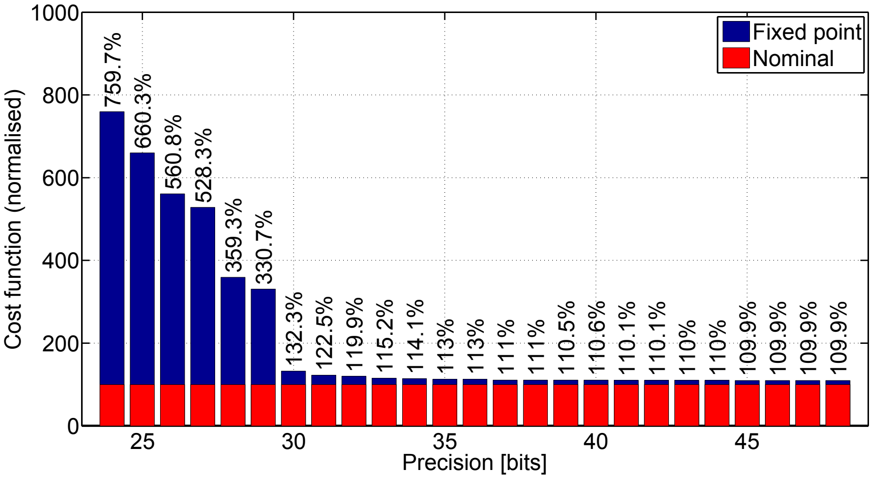
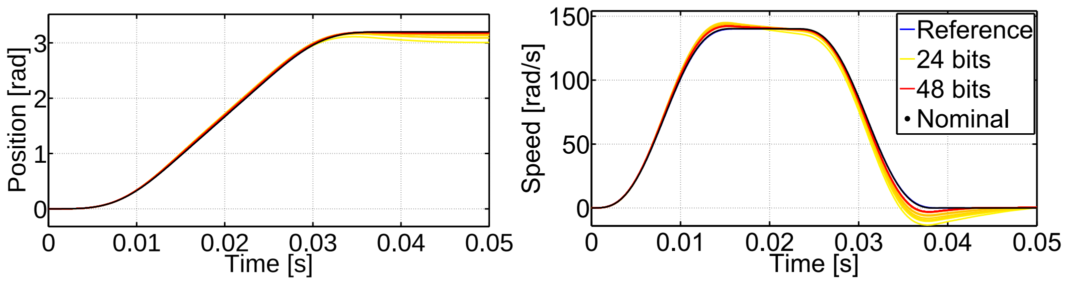


Figure 6.3 Evaluation of the performance drop using fixed-point arithmetic

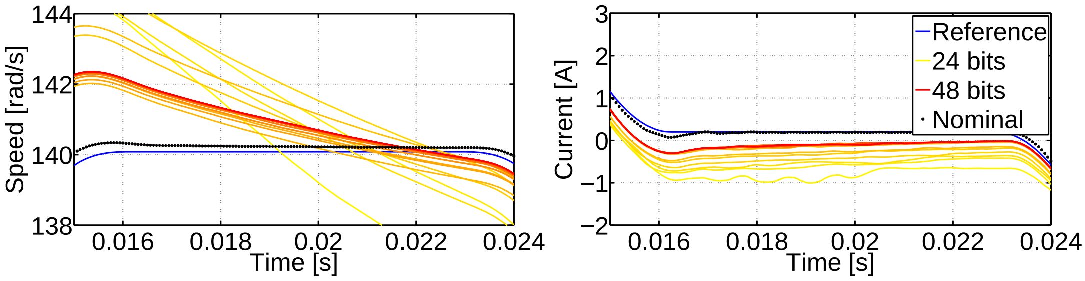
To better show the effects of finite precision implementation, the system response has been plotted and compared with nominal performances. The results are shown in *Fig. 6.4*. Particularly relevant is the fact that the system stability is not affected by the width of the word used to represent data, at least in the considered precision interval.

The central part of the scanning profile corresponds to the phase where the wire crosses the particle beam. This is the moment when the precision of the scan is affected by wire vibrations. Since the vibrations are mostly generated by torque and speed ripple, care must be take to this effects in the design. A detail of the profile in this sensitive phase is shown in *Fig. 6.5* for different fixed-point resolution.



(a) *Position response* **(b)** *Speed response*

Figure 6.4 Response of a system using finite precision



(a) *Speed response (detail)* **(b)** *Current response (detail)*

Figure 6.5 Response of the system in the critical part for vibrations using finite precision

Finally, a fixed-point representation with 32 bits was chosen for the realisation of the controller on the FPGA. This precision was selected accordingly to the plot in *Fig. 6.3*, and also to the type of operations allowed in the libraries used for the implementation. The used libraries are the IEEE standard fixed-point `fixed_float_types_c.vhdl` and `fixed_pkg_c.vhdl`. They are standard libraries in which signed and unsigned, floating and fixed point representation and arithmetic are defined together with a set of math operations and type conversion functions. The package allows full precision multiplication and division operations for inputs up to 32 bits.

The typical use of the fixed-point library in the implementation of a logic block is the following:

- Acquisition of the data in `slv(31 downto 0)` format, a vector of 32 logic elements.

- Conversion of the data to `sfixed(16 downto -15)`, a signed fixed-point number with 15 bits for the integer part and 16 bit for the fractional part. The sign is coded using two's complement.

- Mathematical operations are computed in fixed point using the arithmetic defined in `fixed_pkg_c.vhdl`. The 32-bit precision is maintained throughout the various performed operations.

- Conversion to `slv(31 downto 0)`

- Data are sent to the output port of the block in the standard logic vector format.

6.3 Realisation of the controller

The motion controller was implemented on the FPGA using the ALTERA development environment *Quartus II*, adopting the top-down approach. A top-down approach (also known as stepwise design) is essentially the breaking down of a system to gain insight into its subsystems. In a top-down approach, an overview of the system is formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in detail, using eventually further sublevels to complicate its structure and increasing its functionalities. Some advantages of top-down design are:

- An overview of the overall system is needed to give specifications to its components, but the design attention is focused on a smaller thus simpler subsystem.

- The functionalities of a single component can be improved in at a later stage of the design.

- Subsystems can be replicated and reused, with small modifications, in other parts of the design.

The final version of the firmware will then be organised in a hierarchical architecture, that is, it will be made of hardware blocks containing other simpler hardware blocks and so on. The higher levels of the controller were handled through a schematic interface, provided by the *Quartus II* suite.

The lower levels, on the other hand, were implemented in VHDL language. The use of automatic code generation was avoided to maximise reliability and optimise the resource usage within the board.

Five main components were implemented for the higher-level structure of the BWS motion controller: interface blocks, arithmetic blocks, control blocks, a synchronisation and diagnostic system and data acquisition system.

6.3.1 Interface blocks

The interface blocks are responsible for the exchange of data between the Arria V board and the rest of the system. They consist in memory, I^2C and SPI controllers, used for communication between the FPGA and other integrated circuits, like the current and voltage sensors ADCs. *Serial Peripheral Interface* (SPI) is used to collect data from the sensors. Four sensors need to be read for the control feedback: three for the currents (one for each phase) and one for the position acquisition (the resolver). The two different circuits used for the analog-to-digital conversion, namely *AD2S1210* and *ADC161S626*, both support SPI communication. Some of the characteristics of the *ADC161S626* are listed in *Table 6.2*. It is used for the current sensors and does not need a configuration, thus just a *read* mode is implemented.

Parameter	Value
Resolution (Bits)	16
Sample Rate (max) [SPS]	250 $kSPS$
Interface	Serial SPI
Architecture	SAR
Input Type	Differential
SNR (dB)	93.2

Table 6.2 Parameters of the *ADC161S626* integrated circuit

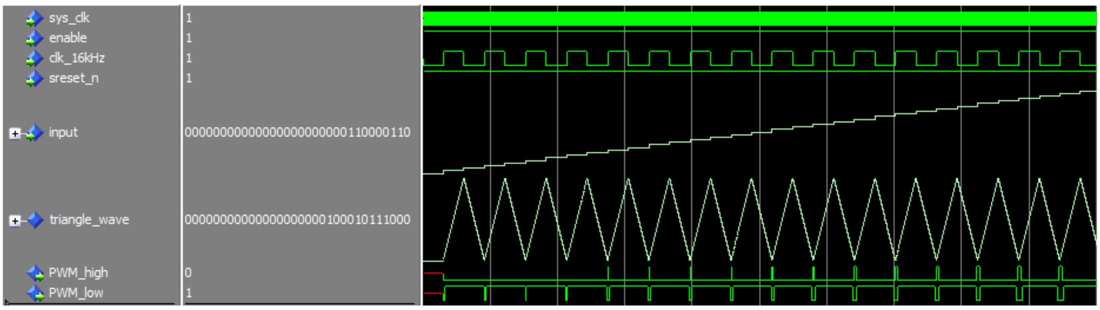
The *ADC161S626*, on the other hand, has several functions to be set that are initialised through a start-up routine.

This category also includes the PWM generator, since the pulse width modulation of the three desired voltages represents the output signal. It is implemented to include a triangle wave generator, a function to rescale the voltages for the comparison and the dead-time integration. A simulation for this block is shown in *Fig. 6.6*.

The dynamic resolution of the PWM generator corresponds to the number of steps in the triangle wave first ramp, which is given by:

$$RES_{PWM} = \frac{2f_{PWM}}{f_{CLK}} = \frac{2 \times 16000}{50000000} = 0.00032 \tag{6.1}$$

which corresponds to an accuracy of 13 bits.



```
sys_clk         1
enable          1
clk_16kHz       1
sreset_n        1
input           00000000000000000000000110000110
triangle_wave   00000000000000000001000101111000
PWM_high        0
PWM_low         1
```

Figure 6.6 Simulation of the PWM generator

6.3.2 Arithmetic blocks

Most of the mathematical operations performed within the FPGA firmware of the motion controller for the BWS actuator exploit a set of arithmetic blocks. Their working principle is based on the `fixed_pkg_c.vhdl` library. This library provides a series of dedicated functions for fixed-point logic, which include sum, subtraction, multiplication and division up to 32 bit for each operand.

Simple examples of arithmetic blocks are the sum and subtraction functions used for feedforward and feedback integration. They basically work following this logic:

- Waits until the two inputs have an updated and valid value (provided by synchronisation blocks).
- Computes sum/difference of the two inputs in fixed point with the given precision.
- Rescales the operation result to the same precision as the input. This step could produce a loss of information if the result is not representable with the chosen precision.

- If no errors occurred, it outputs the correct 32-bits representation of the result and generate a validity signal to trigger the following logic.

A more complicated arithmetic block is represented by the Clarke/Park transform. These transformations consist in not only one simple mathematical operation but several more complicated ones, including the computation of the sine and cosine of the angle given by the position sensor. If the geometric transformation can be expressed through a list of simple operations such as sums and multiplications, this does not apply for a trigonometric computation. Normally, there are two ways to carry out sine and cosine calculations on hardware. The first one includes a dedicated core (*CORDIC* - COordinate Rotation DIgital Computer) that refines the estimation of the desired value through iterations, providing an output value with the desired precision after a certain amount of clock cycles due to pipelined architecture. This solution is usually preferred when different levels of precision are requested, or if a very precise computation is needed. On the other hand, it requires a bigger amount of resources and its response becomes slower as the desired precision increases. A second approach consists in the generation of a look-up table, where the value corresponding to the result of the considered mathematical operation can be fetched. This solution was preferred in the design.

The LUT was generated with a Matlab script, and consists of a table of 16384 elements (coded on 14 *bits*) containing the values of $sin(x)$ for $0 < x < \pi/2$. It was then loaded into the *Quartus II* project by means of an IP block implementing a clocked ROM module. The 16-bit precision was chosen according to the fact that just the fractional part is useful for sine and cosine waves. Besides, the choice of including just a quarter of a full period in the LUT is due to the fact that the slope of a sinusoidal signal is the same in each $\pi/2$ interval, and only its orientation and sign change. This method was used to reduce the amount of memory needed for the implementation by a factor 4.

Using a decoder fed with the desired angle and operation to be computed, the correct location of the memory array can be addressed. A schematic of the *sin/cos* computation block is shown in *Fig. 6.7*, and its simulations in *Fig. 6.8.*

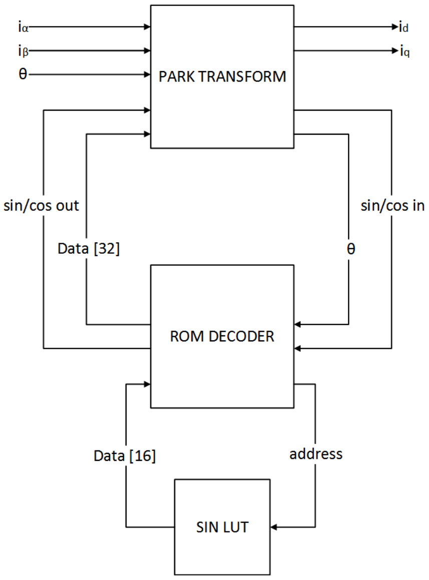
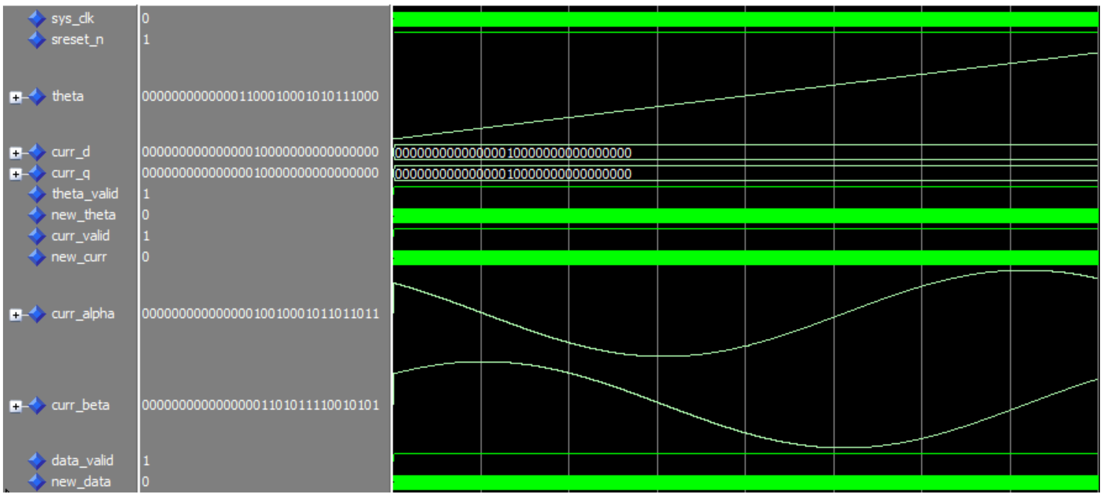


Figure 6.7 Schematic of the sin/cos computation architecture



```
sys_clk       0
sreset_n      1
theta         00000000000001100010001010111000
curr_d        00000000000000001000000000000000
curr_q        00000000000000001000000000000000
theta_valid   1
new_theta     0
curr_valid    1
new_curr      0
curr_alpha    00000000000000010010001011011011
curr_beta     00000000000000001101011110010101
data_valid    1
new_data      0
```

Figure 6.8 Simulation of sin/cos computation through a LUT within Park transform

6.3.3 Control blocks

The main functions of the FPGA controller are carried out by the control blocks. They carry out several of the operations explained in *Chapter 4*, including PID regulation, feedforward and feedback, decoupling and profile generation.

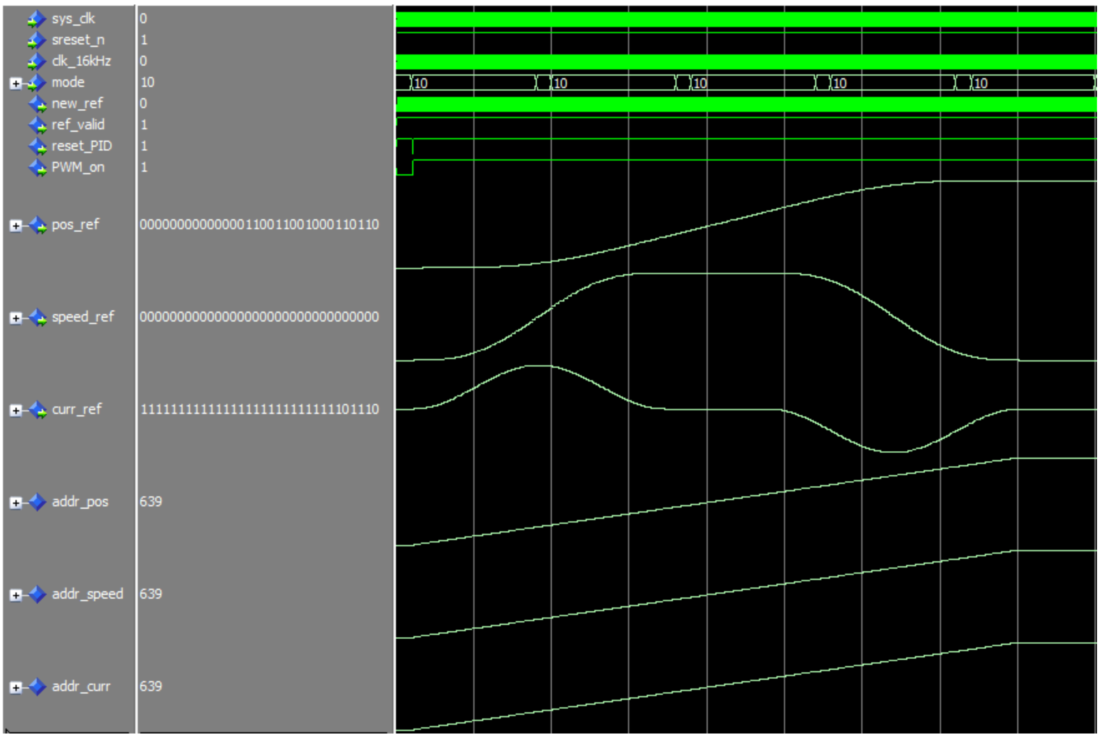
The profile generator is a primary functional hardware block, responsible of the user interface and of the generation of the position, speed and current profiles used as references and for the feedforward action. It implements a *Finite State Machine* (FSM) for the selection of the working mode:

- **Idle :** the motor receives a constant zero reference for position, speed and current. The PWM signals are driven low deactivating the IGBTs and leaving the corresponding pins floating.
- **Lock :** the shaft is locked in a certain position and it is aligned with the magnetic field of the brake, so that no torque is applied by the brake to the motor. It formally corresponds to the 'idle' state, but the PWM signals are activated.
- **Scan :** when this function is activated, the profile generator starts producing references for position, speed and current as outputs. These are subtracted to the measured values in order to activate the controller. Since the LUT for the profiles is read in the two directions, reverse scans are performed with a time interval of 1 s.

The three profiles are loaded in three ROM blocks containing the samples of the desired patterns at a frequency of $f_{PWM} = 16\ kHz$. The simulations of the profile generator are depicted in *Fig. 6.9.*

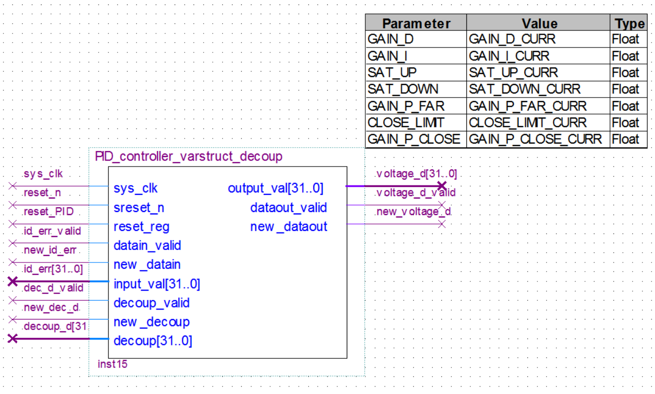
Two versions of PID controllers were realised: a standard PID regulator with auto-saturation and an anti-windup function, and the complicated version including the variable structure shown in *Fig. 4.31.*

A hardware module representation for this second version is outlined in *Fig. 6.10.* The derivative action is not used, but was still implemented to make the block more flexible for eventual later use.



sys_clk	0
sreset_n	1
clk_16kHz	0
mode	10
new_ref	0
ref_valid	1
reset_PID	1
PWM_on	1
pos_ref	00000000000000110011001000110110
speed_ref	00000000000000000000000000000000
curr_ref	11111111111111111111111111101110
addr_pos	639
addr_speed	639
addr_curr	639

Figure 6.9 Simulation of the profile generator



Parameter	Value	Type
GAIN_D	GAIN_D_CURR	Float
GAIN_I	GAIN_I_CURR	Float
SAT_UP	SAT_UP_CURR	Float
SAT_DOWN	SAT_DOWN_CURR	Float
GAIN_P_FAR	GAIN_P_FAR_CURR	Float
CLOSE_LIMIT	CLOSE_LIMIT_CURR	Float
GAIN_P_CLOSE	GAIN_P_CLOSE_CURR	Float

Figure 6.10 Anti-windup, variable structure PID implementation as a FPGA block

6.3.4 Lossless integral

When a finite precision architecture is adopted for mathematical calculations, it is normal to have approximated results within the signals in the dataflow. Approximations are acceptable if they don't affect the performance of the system over a given bound. These specifications were already discussed in *section 6.2.1*.

When an integrator is used, however, the problem is more complicated: since the current value of the integral needs to be stored in the memory and updated at each clock cycle, the error will be accumulating as well, becoming relevant after a certain amount of time. The issue comes from the rescaling function, that cuts off the lowest significant bits which still contain useful informations. Since multiplications and division can be implemented for numbers up to 32 × 32, a hardware complication is needed to increase the precision. A discrete integral is calculated as:

$$I_{new} = I_{old} + e\ T_s \tag{6.2}$$

In our case, the input of the PID block has a resolution of:

$$e_{RES} = 2^{-16} \tag{6.3}$$

and being $T_s = 2^{-14}$ it yields:

$$I_{RES} = e_{RES}\ T_s = 2^{-30} \tag{6.4}$$

The above equations show that an additional $14 - bit$ register is needed to keep in memory the error produced by the integral block for lossless computation. The enhanced architecture including this feature is shown in *Fig. 6.11*.

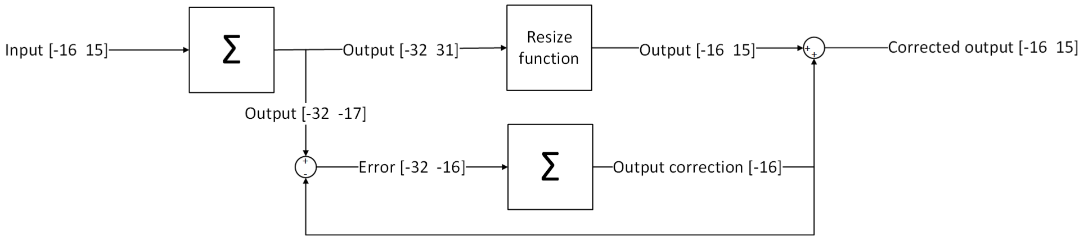


Figure 6.11 Lossless integral architecture

6.3.5 Synchronisation system

The whole BWS firmware is based on two different signals: a fast clock running at 50 MHz, and a slower trigger set to 16 kHz. The clock is generated by the internal 50 MHz single-ended oscillator embedded on the board, while the second one is generated by a counter. They are used to perform two different functions: the fast clock handles all of the arithmetic operations and the synchronous logic, the trigger is used to mark the sampling and the PWM frequency, f_{PWM}.

The timing system of all the implemented blocks exploits the two signals described before. To organise the dataflow, all of the arithmetical and logical operations need to be performed in the correct order: every single function needs to know when the previous one has finished its tasks and provided its outputs. When the inputs of the blocks are ready and stable, it can, in turn, initialise its own routine. For this purpose, the structure of the blocks was designed to be as in *Fig. 6.12*.

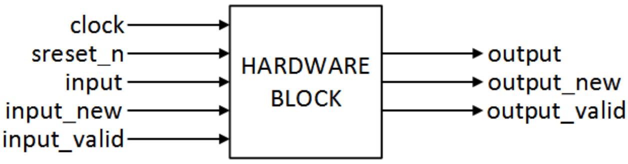


Figure 6.12 Typical architecture of a logic - arithmetic block

This architecture includes at least 5 inputs: the clock, a synchronous reset signal which is active low, an input value and two synchronisation flags, `input_valid` and `input_new`. `input_valid` has a high value when the input data is coherent: it is supposed to stay in that state for the whole time when the system is turned on, except during initialisation, or if a reset or an error occur. The `input_new` signal, on the other hand, generates a pulse when the variable is renewed. When the pulse is detected, the block routine starts. Once the result of the function is ready and stable the `output_valid` flag is raised, if it was previously low, and a pulse is generated on `output_new`. An example of this technique can be seen in *Fig. 6.13*.

Apart from this synchronisation mechanism, some blocks were implemented appositely for synchronisation purpose. Since several functions operate on more than one single input, they need to wait until all of them are available. When more than one signal is input to the block, a previous stage is included, where the multi-

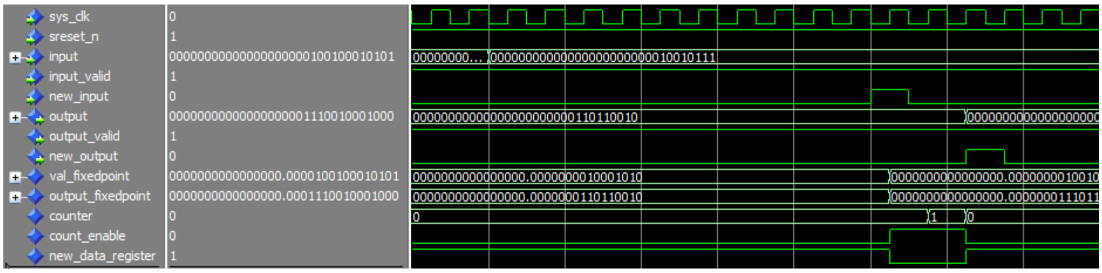


Figure 6.13 Evaluation of the performance drop using fixed-point arithmetic

ple signals are checked before the function is activated. A `Multiple_ready_check` block instance with three inputs on *Quartus II* is depicted in *Fig. 6.14*.

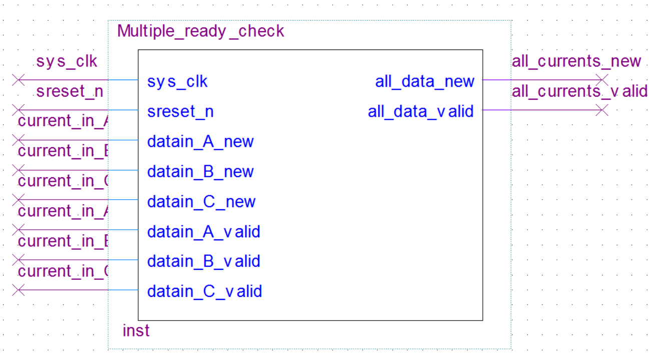


Figure 6.14 Synchronisation blocks check whether all the input data are new and stable

The `All_valid` signal is simply computed as the logic product of the inputs. To generate a pulse in `All_new`, the circuit waits until the three lines present a pulse first, using three registers to store these informations. When all of the registers have been set, the block produces a pulse itself to activate the following system, as in *Fig. 6.15*.

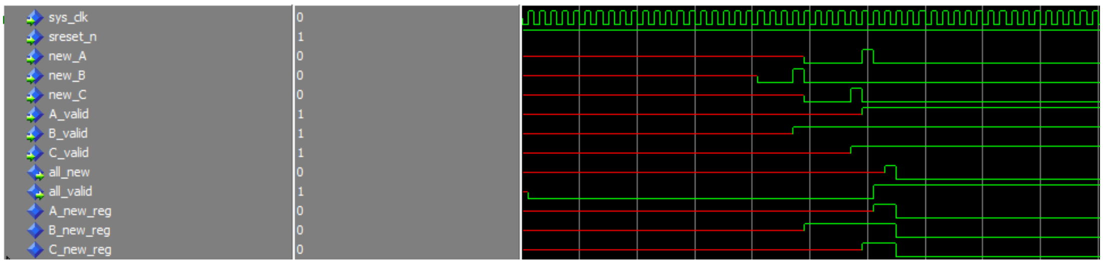


Figure 6.15 Simulation of the synchronisation hardware

Finally, an additional subsystem is used to synchronise the data DSP dataflow: the 16 kHz signal is used to trigger the acquisition from the sensors and the feedback system. The ADCs are anyway free-running between two trigger signals, so that this oversampling can be used in future for filtering.

6.3.6 Diagnostic and data acquisition

The main signals into the controller are acquired during each scan by a parallel system, running on the HPS. Data acquisition is an essential part of the design, since it will register the profile of the scan and all the control variables. It will also include important diagnostic and debug functions.

For this purpose, 35 register with a 32-bit depth were included in the motion controller firmware, to allow the real-time monitoring of the variables. These registers are accessible by means of a custom *Direct Memory Access* (DMA) hardware component for read/write operations. The DMA controller is interfaced with the HPS system, that receives the data using *Avalon Memory Mapped* (MM) *Interface* and stores them on the on-chip memory for later analysis. The acquisition is temporised by a trigger generator controlled by the profile generator. When a scan is requested, the DMA is activated and starts collecting data from the controller for diagnostic.

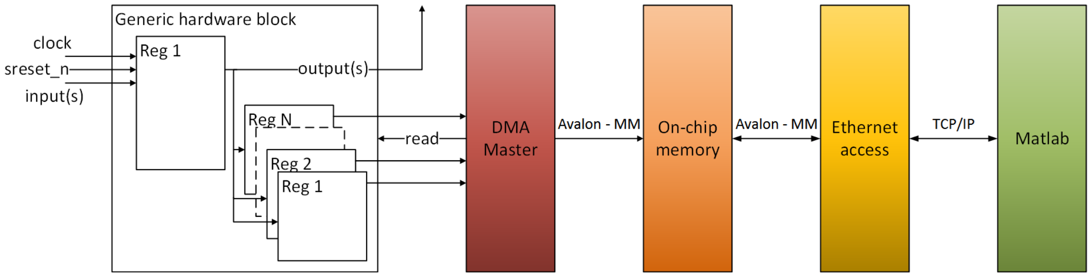


Figure 6.16 Structure of a generic hardware block embedding the Ethernet readout system

Once the data are stored into the memory, the HPS is used to send them to a PC via Ethernet through TCP/IP protocol [44]. This system was designed to be interfaced with Matlab fro real-time analysis and DSP.

To implement this kind of architecture, the blocks need to embed a set of additional registers where the informations are stored during execution, so that the DMA can access them. The structure of a data acquisition chain, then, becomes like in *Fig. 6.16*.

6.3.7 Overview of the control system

The main components of the motion controller FPGA firmware were described in the last chapters. In *Fig. 6.17*, a schematic overview of the system is shown.

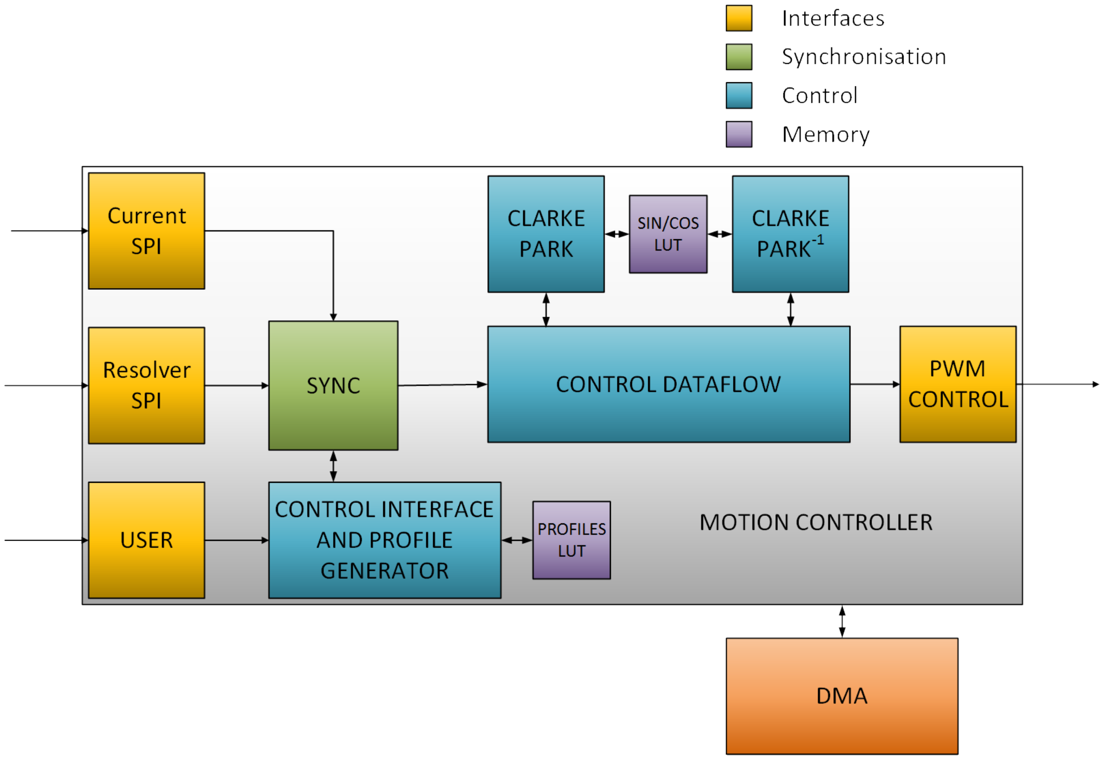


Figure 6.17 Block diagram of the architecture for the motion controller FPGA implementation

6.4 Implementation procedure

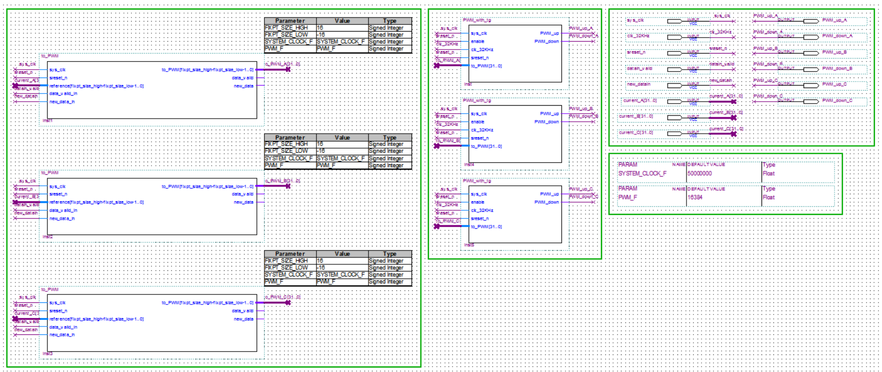
The FPGA controller is composed of several subsystems, all independently functional and all concurring to achieve the final result. To realise a system with a methodological approach, an implementation procedure needs to be defined and followed for each block of the design.

The basic entities of the firmware were realised according to the following steps:

- **Specifications :** as a first step, the specifications for a certain hardware component need to be set. Beside its function, the number of inputs, outputs, the used clock and the implementation principle (internal registers, state machines...) have to be decided. For arithmetic operations, the dataflow and the desired precision are designed and verified before implementation. For the blocks which present strict timing constraints, as the SPI interfaces, a reference waveform was considered and reproduced by means of a state machine.

- **Hardware implementation :** once the structure of the block is settled, the proper implementation part can start. As we said previously, a VHDL implementation was chosen for low-level entities, while the higher levels were realised in schematic. Normally, the architecture of the considered entity should contain just two processes: a synchronous one, including only the clock in its *sensitivity list*, and an asynchronous one, sensitive to all the signals contained in its body. All of the implemented hardware blocks contain a header, in which the main functions, the used constants, the implementation procedure and eventual comments on missing functionalities or possible improvements are included. An example of schematic implementation is shown in *Fig. 6.18*, where the PWM generator block is represented. Different zones are identified on the GUI and separated for subsystems of different kinds, and two additional portions of the sheet are dedicated to the I/O pins and the generics.

- **Simulation :** the simulation for the VHDL components was performed on *ModelSim* software. For every compiled block, a *testbench* file in VHDL was realised and used for the simulations. A *testbench* file is a portion of code, written typically in VHDL but also in c and others, that simulates



Parameter	Value	Type
FIXPT_SIZE_HIGH	16	Signed Integer
FIXPT_SIZE_LOW	-16	Signed Integer
SYSTEM_CLOCK_F	SYSTEM_CLOCK_F	Signed Integer
PWM_F	PWM_F	Signed Integer

Parameter	Value	Type
FIXPT_SIZE_HIGH	16	Signed Integer
FIXPT_SIZE_LOW	-16	Signed Integer
SYSTEM_CLOCK_F	SYSTEM_CLOCK_F	Signed Integer
PWM_F	PWM_F	Signed Integer

Parameter	Value	Type
FIXPT_SIZE_HIGH	16	Signed Integer
FIXPT_SIZE_LOW	-16	Signed Integer
SYSTEM_CLOCK_F	SYSTEM_CLOCK_F	Signed Integer
PWM_F	PWM_F	Signed Integer

PARAM NAME	DEFAULT VALUE	Type
SYSTEM_CLOCK_F	50000000	Float
PWM_F	16384	Float

Figure 6.18 Schematic implementation of the PWM generator

the behaviour of the logic which is external to the considered *Device Under Test* (DUT). ModelSim presents a graphic interface, already shown in some simulations, where the waveforms related to the system architecture can be plotted and explored. The various subsystems were tested for a exhaustive range of input values to verify their proper behaviour. This methodology was iterated for bigger design parts, up to the top level of the motion controller firmware.

- **Hardware verification :** the main functionalities were subject to two testing phases: a software simulation, as explained before, and a real preliminary hardware test, realised on smaller board used for firmware prototyping. The adopted board is a Terasic DE2-115 board based on ALTERA's Cyclone IV model FPGA (*Fig. 6.19*). Some of its characteristics are listed in *Table 6.3*.

  In this case, an actual testbench was realised on hardware to properly excite the blocks with coherent signals, in order to verify its behaviour. The testbenches basically consist in a proper interface architecture with the external hardware, an input generator and a debug system. To monitor the internal signal, the *SignalTap* GUI provided within the Quartus II suite was used. *SignalTap* allows the implementation of a logic analyser using the circuitry in the FPGA which is left free by the project. The signals of interest can be selected in a list of existing variable, corresponding to actual wires within the circuit. By comparing the results achieved in simulation with the ones obtained through hardware test, the proper operation of the component can be verified.

Hardware verification is more complex and time-consuming than simulations, but it is the only method to guarantee the efficiency of the system with certainty.

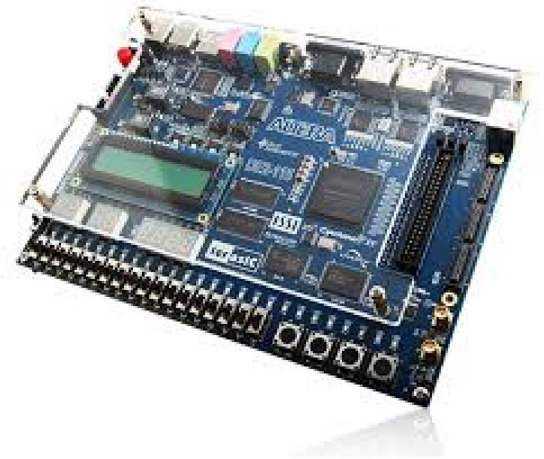


Figure 6.19 Terasic DE2-115 board

Cyclone IV EP4CE115
114,480 logic elements (LEs)
3,888 Embedded memory (Kbits)
266 Embedded 18 x 18 multipliers
4 General-purpose PLLs
528 User I/Os

Table 6.3 ALTERA Cyclone IV characteristics

6.5 Real-time constraints

Digital controllers are intrinsically real-time systems: their response must be available within a known amount of time to be usable. For the described application, the constraints are satisfied if dataflow between the inputs and the outputs of the architecture takes less than one sampling period to be entirely computed.

It was already pointed out how the various subsystems are activated by the previous ones once their task is accomplished, thus, calculating the longest possible duration to complete all the operations, a worst-case latency can be evaluated.

Recalling that the frequency of the two signals is 16 kHz and 50 MHz, the maximum duration of the algorithm is:

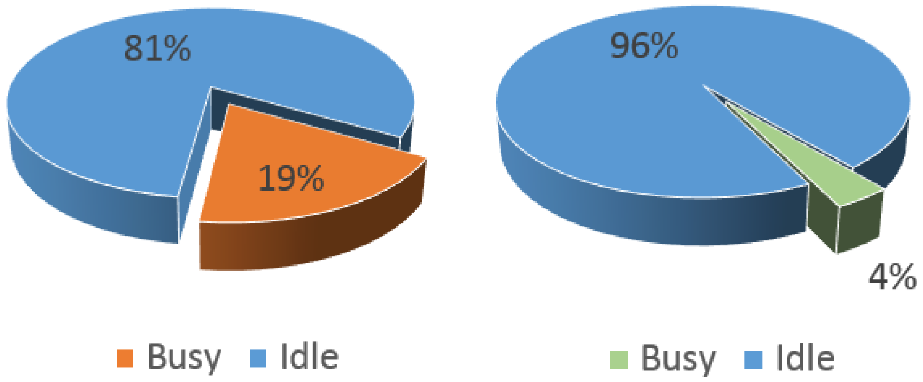
$$\frac{50 \times 10^6}{16 \times 10^3} = 3125 \; clock \; cycles$$

Function	Time consumption [clock cycles]
Current SPI	200
Resolver SPI	200
Synchronisation	8
Current acquisition	1
Position acquisition	8
Profile Generator	4
Multiple turn curreсtion	8
Speed computation	6
Clarke-Park	55
Clarke-Park inverse	56
Adder-subtractor	36
PID controller	76
Decouplers	12
PWM generator	4

Table 6.4 Time consumption for the different functions

Table 6.4 shows the impact of the different tasks on the total required time, multiplied by the number of blocks of the same kind. It is clear how the duration of the subtasks carried out at a frequency of 50 MHz is much smaller then the sampling period.

in *Fig. 6.20* the worst-case requested time, represented by the case in which all the tasks are accomplished sequentially, and the simulated computing time are shown.



(a) *Idle and busy time - worst case* **(b)** *Idle and busy time - simulated*

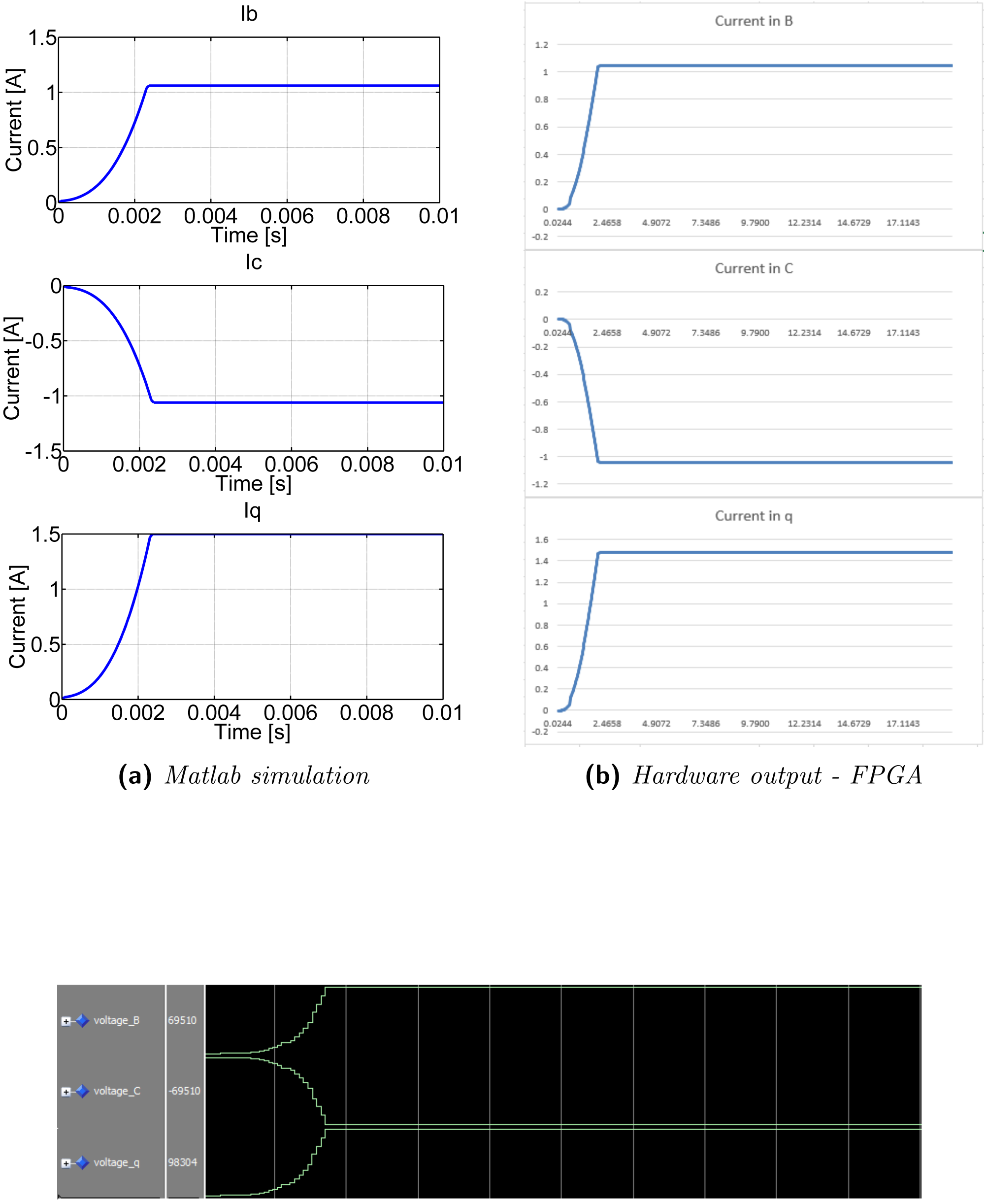
Figure 6.20 Load for the controller implemented on FPGA

6.6 Simulation vs hardware implementation

The final version of the motion controller firmware was tested on *ModelSim* and via hardware verification before the final implementation on the Arria V DevKit. Since an extremely complicated testbench would be needed to simulate the behaviour of the beam wire scanner actuator, the design was simulated for a set of constant inputs always equals to zero, in open-loop. Then, the results of the simulations were compared with the output of the actual hardware, and finally with the Matlab model of the controller. Only a set of output references were considered, the other ones were stuck at zero, thus not relevant for the evaluations The results are shown in *Fig. 6.21*

Unfortunately, the ultimate version of the hardware is currently not available at the time when this work is coming to an end, thus a real test of the system in closed loop was not possible.

It will be subject of future work.



(a) *Matlab simulation*

(b) *Hardware output - FPGA*

(c) *Modelsim simulation - FPGA*

Figure 6.21 Matlab simulation vs hardware simulation vs hardware output for the controller in open-loop action

7. Conclusions and future work

In particle accelerators, the strong dependency of the interaction rate on the beam sizes motivates the requirement of an accurate measurement of the beam profiles. The *Beam Wire Scanner* (BWS) belongs to the category of interceptive beam transverse profile measurement instruments, and it consists in an electro-mechanical device which measures the transversal beam density profile by moving a thin wire through the beam intermittently and measuring the quantity of secondary particles produced during the interaction.

The control of its actuator, consisting in a PMSM, requires special care due to the strong requirements in terms of performance and robustness in the three controlled quantities: position, speed and current. Several design choices, including nonlinearities introduced by mechanical installations, a closed-loop position sensor with limited accuracy which reduces the achievable bandwidth, the presence of a variable length power cable and others make the motion control design for this device a complicated engineering task.

In these pages, the design of a dedicated digital controller was presented and can besummarised as follows:

- A model of the motor was produced starting from mathematical models of the components. Some of them, such as the power cable, were also experimentally verified to conclude on their reliability.

- The proper controller was designed from the basic architecture of the field oriented control, sized to match the design specifications, and improved by means of several complications to increase its efficiency and its robustness.

- The proposed controller was interfaced with the BWS drive and its performance was tested on a testbench appositely built in the CERN laboratory, exploiting a dSPACE prototyping system for the hardware-in-the-loop implementation.

- The final version of the motion controller was finally implemented on a FPGA board in VHDL language.

The linear transfer functions and the nonlinear contributions were extrapolated by their physical equations or by means of simulation tools. The experimental sessions consisted in frequency-response evaluation for the two devices presenting uncertainties in their parameters estimation: the power cable and the sine filter. They were carried out using a function generator, or a the dSPACE system controlled by a PC to input a known signal in the cable, and an oscilloscope to gather the output data. The results show a good fit with the proposed model. Furthermore, the three-phase transfer function of the power cable was transferred to the $d - q$ frame as in [APPENDIX A].

The basic scheme of the field oriented control includes three hierarchical feedback loops regarding position, speed and current regulation. The flux-producing and the current-producing currents are computed by means of Clarke/Park transformations. This architecture was improved by adding current decouplers to linearise the currents transfer function and feedforward action to speed up the system response and contrast the bandwidth limitation due to the closed-loop position acquisition system. A Steady-state Kalman filter was implemented to solve the problem of the poor accuracy in the speed measure deriving by the use of a discrete derivative. The PID controllers present an enhanced architecture, including a conditioned anti-windup system and a variable structure on the proportional action. This last solution was necessary to eliminate the vibrations during the steady-state generated by the long cable. The tuning was performed in three phases: an analytical one, based on zero-pole cancellation, a numerical one carried out using the SISOTOOL software in Matlab, and a last, experimental one consisting in an iterative optimisation of the IAE during a scan by means of a particle filter. It was implemented in Python language on the dSPACE platform.

The experimental testbench, set up within the CERN laboratory, allowed a closed-loop test of the system for verification and tuning. Experimental results showed a good fit with respect to the modelled system. The observed performances were comparable for both the motor bencha and the prototype of the actual BWS actuator. During the testing phase, nonlinearities in the value of the motor constant, K_T, were observed, quantified and corrected.

As a final step, a hardware version of the motion controller was implemented on a FPGA board in VHDL language. The firmware included the interfaces with the external devices for sampling and actuation, and an internal dataflow design based on a 32-bit fixed-point pipelined architecture. The synchronisation was handled by dedicated hardware blocks and triggered with a sampling frequency of 16 kHz. The data acquisition was carried out by a DMA hardware component. The real time-constraints were set and are verified online. Finally, the firmware was implemented on the FPGA board and the acquired open-loop data were compared with the Simulink and Modelsim simulations showing the same behaviour. This step represented the final step of verification for the hardware implementation included in this work.

Despite the large amount of work carried out within this thesis and other people's work, the beam wire scanner system will still need some steps of improvements and verification before being installed into the accelerators.

From the control point of view, the improvement of the position acquisition by means of the optical encoder could increase the quality of the speed response and stability. The resolver-to-digital converter could be implemented into the FPGA firmware in order to obtain a more compact system and reduce the number of needed interfaces. Furthemore, an iterative optimisation of the controller for the minimisation of the vibrations in the wire based on its model is ongoing. Possible hardware improvements could allow a feedback compensation of the dead-time in the switching pattern. The possibility of implementing a faster PWM, allowed by a different kind of transistors in the amplifier could also be investigated. Moreover, the FPGA firmware needs to be tested in closed-loop for a final verification. In the near future, the actuator qualification and calibration procedure will take place. The testing phase should conclude by the end of the summer of 2015, while the installation of the new scanners within the CERN facility is foreseen for the second Long Shutdown, scheduled for 2018-2019.

8. Appendix A - Clarke/Park maths for cable model frame switch

In this appendix, the mathematical procedures regarding the computation of the effects of Clarke/Park transformations on the transfer function of the cable will be described in details. The necessary background for the following calculations was given in (3.2.3) and (3.7.3).

Our goal is to obtain the transfer function between the u_d and u_q currents from the input to the output of the power cable. We recall the equation describing this relationship, (3.46):

$$\begin{bmatrix} u_{d_{out}} \\ u_{q_{out}} \\ u_{0_{out}} \end{bmatrix} = \frac{2}{3} \begin{bmatrix} c_\theta & c_{\theta-\frac{2\pi}{3}} & c_{\theta+\frac{2\pi}{3}} \\ -s_\theta & -s_{\theta-\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix} \begin{bmatrix} c_\theta & -s_\theta & \frac{1}{2} \\ c_{\theta-\frac{2\pi}{3}} & -s_{\theta-\frac{2\pi}{3}} & \frac{1}{2} \\ c_{\theta+\frac{2\pi}{3}} & -s_{\theta+\frac{2\pi}{3}} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_{d_{in}} \\ u_{q_{in}} \\ u_{0_{in}} \end{bmatrix}$$

where t_D and t_C represent, respectively, the direct attenuation and the cross-talk transfer functions of the cable itself.

To simplify the notation, we can fix:

$$\frac{2\pi}{3} = \alpha$$

and recall:

$$sin(\alpha) = sin\left(\frac{2\pi}{3}\right) = \frac{\sqrt{3}}{2}$$
$$cos(\alpha) = cos\left(\frac{2\pi}{3}\right) = -\frac{1}{2}$$

First, we will compute the effect of $u_{d_{in}}$ on $u_{d_{out}}$:

$$u_{d_{out}}|_{u_{q_{in}}=0}\, f(t) = \frac{2}{3}\begin{bmatrix} c_\theta & c_{\theta-\alpha} & c_{\theta+\alpha}\end{bmatrix}\begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix}\begin{bmatrix} c_\theta \\ c_{\theta-\alpha} \\ c_{\theta+\alpha}\end{bmatrix} u_{d_{in}} \tag{8.2}$$

that yields:

$$\frac{u_{d_{out}}}{u_{d_{in}}} = \frac{2}{3}\left(t_D(c_\theta^2 + c_{\theta-\alpha}^2 + c_{\theta+\alpha}^2) + 2t_C(c_\theta c_{\theta+\alpha} + c_\theta c_{\theta-\alpha} + c_{\theta-\alpha}c_{\theta+\alpha})\right) \tag{8.3}$$

than, we can analyse the two parts of the (8.3) separately :

$$\begin{aligned} c_\theta^2 + c_{\theta-\alpha}^2 + c_{\theta+\alpha}^2 &= c_\theta^2 + (c_\theta c_\alpha - s_\theta s_\alpha)^2 + (c_\theta c_\alpha + s_\theta s_\alpha)^2 \\ &= c_\theta^2 + c_\theta^2 c_\alpha^2 + s_\theta^2 s_\alpha^2 - 2c_\theta c_\alpha s_\theta s_\alpha + c_\theta^2 c_\alpha^2 + s_\theta^2 s_\alpha^2 + 2c_\theta c_\alpha s_\theta s_\alpha \\ &= c_\theta^2 + 2c_\theta^2 c_\alpha^2 + 2s_\theta^2 s_\alpha^2 \\ &= c_\theta^2 + 2c_\theta^2\frac{1}{4} + 2s_\theta^2\frac{3}{4} \\ &= \frac{3}{2}c_\theta^2 + \frac{3}{2}s_\theta^2 \\ &= \frac{3}{2} \end{aligned} \tag{8.4}$$

$$\begin{aligned} c_\theta c_{\theta+\alpha} + c_\theta c_{\theta-\alpha} + c_{\theta-\alpha}c_{\theta+\alpha} &= c_\theta(c_\theta c_\alpha - s_\theta s_\alpha) + c_\theta(c_\theta c_\alpha + s_\theta s_\alpha) \\ &\quad + (c_\theta c_\alpha + s_\theta s_\alpha)(c_\theta c_\alpha - s_\theta s_\alpha) \\ &= 2c_\theta^2 c_\alpha + c_\theta^2 c_\alpha^2 - s_\theta^2 s_\alpha^2 \\ &= -c_\theta^2 + c_\theta^2\frac{1}{4} - s_\theta^2\frac{3}{4} \\ &= -\frac{3}{4} \end{aligned} \tag{8.5}$$

finally, (8.3) becomes:

$$\frac{u_{d_{out}}}{u_{d_{in}}} = \frac{2}{3}\left(t_D\frac{3}{2} + 2t_C\left(-\frac{3}{4}\right)\right) = t_D - t_C \tag{8.6}$$

Now, considering $u_{q_{in}}$ on $u_{q_{out}}$:

$$u_{q_{out}}|_{u_{d_{in}}=0} = \frac{2}{3}\begin{bmatrix}-s_\theta & -s_{\theta-\alpha} & -s_{\theta+\alpha}\end{bmatrix}\begin{bmatrix}t_D & t_C & t_C\\ t_C & t_D & t_C\\ t_C & t_C & t_D\end{bmatrix}\begin{bmatrix}-s_\theta\\ -s_{\theta-\alpha}\\ -s_{\theta+\alpha}\end{bmatrix}u_{q_{in}} \tag{8.7}$$

that yields:

$$\frac{u_{q_{out}}}{u_{q_{in}}} = \frac{2}{3}\left(t_D(s_\theta^2 + s_{\theta-\alpha}^2 + s_{\theta+\alpha}^2) + 2t_C(s_\theta s_{\theta+\alpha} + s_\theta s_{\theta-\alpha} + s_{\theta-\alpha}s_{\theta+\alpha})\right) \tag{8.8}$$

proceeding as before:

$$\begin{aligned}s_\theta^2 + s_{\theta-\alpha}^2 + s_{\theta+\alpha}^2 &= s_\theta^2 + (s_\theta c_\alpha - c_\theta s_\alpha)^2 + (s_\theta c_\alpha + c_\theta s_\alpha)^2\\ &= c_\theta^2 + s_\theta^2 c_\alpha^2 + c_\theta^2 s_\alpha^2 - 2c_\theta c_\alpha s_\theta s_\alpha + s_\theta^2 c_\alpha^2 + c_\theta^2 s_\alpha^2 + 2c_\theta c_\alpha s_\theta s_\alpha\\ &= s_\theta^2 + 2s_\theta^2 c_\alpha^2 + 2c_\theta^2 s_\alpha^2\\ &= s_\theta^2 + 2s_\theta^2\frac{1}{4} + 2c_\theta^2\frac{3}{4}\\ &= \frac{3}{2}s_\theta^2 + \frac{3}{2}c_\theta^2\\ &= \frac{3}{2}\end{aligned} \tag{8.9}$$

$$\begin{aligned}s_\theta s_{\theta+\alpha} + s_\theta s_{\theta-\alpha} + s_{\theta-\alpha}s_{\theta+\alpha} &= s_\theta(s_\theta c_\alpha + c_\theta s_\alpha) + c_\theta(s_\theta c_\alpha - c_\theta s_\alpha)\\ &\quad + (s_\theta c_\alpha - c_\theta s_\alpha)(s_\theta c_\alpha + c_\theta s_\alpha)\\ &= 2s_\theta^2 c_\alpha + s_\theta^2 c_\alpha^2 - c_\theta^2 s_\alpha^2\\ &= -s_\theta^2 + s_\theta^2\frac{1}{4} - c_\theta^2\frac{3}{4}\\ &= -\frac{3}{4}\end{aligned} \tag{8.10}$$

that gives again:

$$\frac{u_{q_{out}}}{u_{q_{in}}} = \frac{2}{3}\left(t_D\frac{3}{2} + 2t_C\left(-\frac{3}{4}\right)\right) = t_D - t_C \tag{8.11}$$

The obtained results show that the transfer function of the cable is the samefor both the direct and quadrature voltages. Furthermore, this function is LTI if the cable effects are linear and time invariant.

Now, let's focus on the coupling actions:

$$u_{d_{out}}|_{u_{d_{in}}=0} = \frac{2}{3}\begin{bmatrix} c_\theta & c_{\theta-\alpha} & c_{\theta+\alpha}\end{bmatrix}\begin{bmatrix} t_D & t_C & t_C \\ t_C & t_D & t_C \\ t_C & t_C & t_D \end{bmatrix}\begin{bmatrix} -s_\theta \\ -s_{\theta-\alpha} \\ -s_{\theta+\alpha}\end{bmatrix} u_{q_{in}} \tag{8.12}$$

that yields:

$$\begin{aligned}\frac{u_{d_{out}}}{u_{q_{in}}} = \frac{2}{3}(t_D(-s_\theta c_\theta - s_{\theta-\alpha}c_{\theta-\alpha} - s_{\theta+\alpha}c_{\theta+\alpha}) + t_C(-c_\theta(s_{\theta-\alpha} + s_{\theta+\alpha}) - c_{\theta-\alpha}(s_\theta + s_{\theta+\alpha})\\ - c_{\theta+\alpha}(s_\theta + s_{\theta-\alpha}c_{\theta+\alpha})))\end{aligned} \tag{8.13}$$

$$\begin{aligned}-s_\theta c_\theta - s_{\theta-\alpha}c_{\theta-\alpha} - s_{\theta+\alpha}c_{\theta+\alpha}&\\ &= -s_\theta c_\theta - (s_\theta c_\alpha - c_\theta s_\alpha)(c_\theta c_\alpha + s_\theta s_\alpha) - (s_\theta c_\alpha + c_\theta s_\alpha)\\ &\quad (c_\theta c_\alpha - s_\theta s_\alpha)\\ &= -s_\theta c_\theta - (s_\theta c_\theta c_\alpha^2 + s_\theta^2 s_\alpha c_\alpha - s_\alpha c_\theta^2 c_\alpha - s_\theta s_\alpha^2 c_\theta)\\ &\quad - (s_\theta c_\theta c_\alpha^2 - s_\theta^2 s_\alpha c_\alpha + s_\alpha c_\theta^2 c_\alpha - s_\theta s_\alpha^2 c_\theta)\\ &= -s_\theta c_\theta - 2s_\theta c_\theta c_\alpha^2 + 2s_\theta s_\alpha^2 c_\theta\\ &= s_\theta c_\theta\left(-1 - \frac{1}{2} + \frac{3}{2}\right)\\ &= 0\end{aligned} \tag{8.14}$$

$$\begin{aligned}-c_\theta(s_{\theta-\alpha} + s_{\theta+\alpha}) - c_{\theta-\alpha}(s_\theta + s_{\theta+\alpha}) - c_{\theta+\alpha}(s_\theta + s_{\theta-\alpha})&\\ &= -c_\theta(s_\theta c_\alpha - c_\theta s_\alpha + s_\theta c_\alpha + c_\theta s_\alpha) - (c_\theta c_\alpha + s_\theta s_\alpha)(s_\theta + s_\theta c_\alpha + c_\theta s_\alpha)\\ &\quad - (c_\theta c_\alpha - s_\theta s_\alpha)(s_\theta + s_\theta c_\alpha - c_\theta s_\alpha)\\ &= -4s_\theta c_\theta c_\alpha - 2s_\theta c_\theta c_\alpha^2 - 2s_\theta s_\alpha^2 c_\theta\\ &= s_\theta c_\theta\left(2 - \frac{1}{2} - \frac{3}{2}\right)\\ &= 0\end{aligned} \tag{8.15}$$

so we can write:

$$\frac{u_{d_{out}}}{u_{q_{in}}} = 0 \tag{8.16}$$

Finally, we compute:

$$u_{q_{out}}|_{u_{q_{in}}=0} = \frac{2}{3}\begin{bmatrix}-s_\theta & -s_{\theta-\alpha} & -s_{\theta+\alpha}\end{bmatrix}\begin{bmatrix}t_D & t_C & t_C\\ t_C & t_D & t_C\\ t_C & t_C & t_D\end{bmatrix}\begin{bmatrix}c_\theta\\ c_{\theta-\alpha}\\ c_{\theta+\alpha}\end{bmatrix}u_{d_{in}} \tag{8.17}$$

that yields:

$$\begin{aligned}\frac{u_{q_{out}}}{u_{d_{in}}} &= \frac{2}{3}(t_D(-s_\theta c_\theta - s_{\theta-\alpha}c_{\theta-\alpha} - s_{\theta+\alpha}c_{\theta+\alpha}) + t_C(-c_\theta(s_{\theta-\alpha} + s_{\theta+\alpha}) - c_{\theta-\alpha}(s_\theta + s_{\theta+\alpha})\\ &\quad - c_{\theta+\alpha}(s_\theta + s_{\theta-\alpha}c_{\theta+\alpha})))\end{aligned} \tag{8.18}$$

being (8.18) equal to (8.13), we can conclude similarly:

$$\frac{u_{q_{out}}}{u_{d_{in}}} = 0 \tag{8.19}$$

Now it is possible to write the transfer function from $u_{d_{in}}$ and $u_{q_{in}}$ to $u_{d_{out}}$ and $u_{q_{out}}$ as following:

$$\begin{bmatrix}u_{d_{out}}\\ u_{q_{out}}\end{bmatrix} = \begin{bmatrix}t_D - t_C & 0\\ 0 & t_D - t_C\end{bmatrix}\begin{bmatrix}u_{d_{in}}\\ u_{q_{in}}\end{bmatrix} \tag{8.20}$$

The last equation represent very important results: it shows that the output voltages are related to the input voltages by a symmetrical equation, which is not only linear and time-invariant in the cable tfs, but also decoupled.

Bibliography

[1] ATLAS - A Toroidal Lhc ApparatuS, URL: http://www.atlas.ch/

[2] CMS - The Compact Muon Solenoid, URL: http://cms.web.cern.ch/

[3] ALICE - A Large Ion Collider Experiment, URL: http://aliceinfo.cern.ch/

[4] LHCb - The Large Hadron Collider Beauty experiment, URL: http://lhcb.web.cern.ch/lhcb/

[5] TOTEM - Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC, URL: http://totem.web.cern.ch/Totem/

[6] LHCf - Large Hadron Collider forward, URL: http://home.web.cern.ch/about/experiments/lhcf/

[7] W. Herr, B. Muratori, "*Concept of luminosity*".

[8] J. L. Sirvent Blasco, "*Design of an optical fibre based angular position sensor for wire scanners complying with ultra-high vacuum, high temperature and radiation conditions of the CERN's accelerators*", Master's thesis, Escuela Politécnica Superior de Elche, 2012

[9] H. Wiedemann, "*Particle Accelerator Physics*", 3^{rd} edition, Springer.

[10] C. Fischer, J. Koopman, "*Ionization profile monitor tests in the SPS*", Proc. European Particle Accelerator Conf., Vienna, 2000, CERN-SL-99-046 BI (2000).

[11] F.Roncarolo et al. "*Synchotron Radiation Measurements at the CERN LHC*", Prod. DIPAC 2011, Germany.

[12] J. Emery et al., "*A fast and accurate wire scanner instrument for the CERN accelerators to cope with severe environmental constraints and an increased demand for availability*", 8-10 October 2014 Antibes, France.

[13] J. Bosser. "*Beam Instrumentation*", 1994 Services Educatifs-Engeignement Technique.

[14] S.E. Forde "*Wire Scanner Motion Control Card*", 2007 Bergen University Collete (Norway).

[15] J. Koopman et al. "*Fast Wire Scanner calibration*", CERN, Geneva, Switzerland

[16] M. Sapinski, F. Cerutti, K. Dahlerup-Petersen, B. Dehning, J. Emery, A. Ferrari, A. Guerrero, E. B. Holzer, M. Koujili, A. Lechner, E. Nebot, M. Scheubel, J. Steckert, A. Verweij, and J. Wenninger "*Lhc magnet quench test with beam loss generated by wire scan*", oai:cds.cern.ch:1379465 no. CERN-ATS-2011-062, p. 3 p, Sep 2011. 13, 22

[17] M. Sapinski, "*Model of carbon wire heating in accelerator beam*", oai:cds.cern.ch:1123363 CERN, Geneva, Tech. Rep. CERN-AB-2008-030-BI, Jul 2008. xii, 24

[18] J. H. Alvarez, "*2 dof wire dynamic model*", CERN, Geneva, Tech. Rep. CERN EDMS Document Id 1246770. 104

[19] J. Puranen and L. Teknillinen Tliopisto "*Induction Motor Versus Permanent Magnet Synchronous Motor in Motion Control Applications: A Comparative Study*", ser. Acta Universitatis Lappeenrantaensis. Lappeenranta University of Technology, 2006.

[20] M. Koujili "*Design and construction of a new actuator for the LHC wire scanner*", Thèse de doctorat en Génie Electrique, Université de Technologie Belfort - Montbéliard, 2013

[21] Analog Devices electronics TDS for AD2S1210 component, Rev. c. 2008–2010, Analog Devices, Inc.

[22] K. Henzer "*Détermination précise de la position d'un faisceau de particules à l'aide d'un résolveur magnétique*", Bachelor's thesis, Haute Ecole d'Ingégnerie et de Gestion du Canton de Vaud, 2013

[23] S. Samuelsson "*Mechanical optimisation of a high-precision fast wire scanner at CERN*", Master's thesis, CHALMERS UNIVERSITY OF TECHNOLOGY, 2013

[24] C. Grosjean "*Motirisation d'un scanner du CERN*", Travail de Bachelor, Haute Ecole d'Ingégnerie et de Gestion du Canton de Vaud, 2014

[25] S. Cantin, "*EMC study for the control of Wire Scanner*", Haute Ecole d'Ingégnerie et de Gestion du Canton de Vaud, no Project: 33206, 2013

[26] J. Emery, "*Beam Wire Scanner Control, Monitoring and Supplies part - Engineering specifications*", CERN, Geneva, Tech. Rep. CERN EDMS Document Id 1318827. 104

[27] D. Jouve and D. Bui, "*Influence of the motor feedback sensor on ac brushless servo drive performances*", May 2003

[28] T. Nussbaumer, M.L. Heldwein, G. Gong, J.W. Kolar, "*Prediction techniques compensating delay times caused by digital control of a three-phase buck-type pwm rectifier system*", In IEEE Industry Applications Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005, volume 2, pages 923–927.

[29] W. Yannick, "*Modelisation des cable d'energie soumis aux contraintes generees par les convertisseurs electroniques de puissance*", Universitèe des Sciences et Technologies de Lille, 2006

[30] W. Y. Svrcek, D. P. Mahoney, B. R. Young, "*A Real Time Approach to Process Control*", 2nd Edition. John Wiley & Sons, Ltd.

[31] R. D. Lorenz, K. Van Patten, "*High Resolution Estimation for All Digital AC Servo Drives*", Conf. Record of the IEEE Industry Applications Society Annual Meeting, Vol. 1, 1988.

[32] M. R. Stojic, S. N. Vukosavic, "*Design of the Observer-based Speed Controller Applied in Servo Drives with Limited Resolution of Position*", Facta Universitatis (Nis) Series: Elec- tronics and Energetics, Vol. 1, 1995.

[33] L. Xu, C. Wang , "*Implementation and ex- perimental 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

[34] 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

[35] A. Bellini, S. Bifaretti, S. Costantini, "*Employment of a Kalman Filter to Reduce the Speed Measurement Noise in a Digitally Controlled Drive*", IASTED Int. Conf. On Modelling, Identification and Control (MIC 2001), Innsbruck, 19–22 February 2001.

[36] C.Vadstrup, "*Stability analysis of a Permanent Magnet Synchronous Machine with inverter output LC filter*", Master's thesis, AALBORG UNIVERSITY, 2013

[37] 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

[38] N. Mohan, T.M. Undeland, W.P. Robbins, "*Power Electronics – Converters, Applications and Design*", Second Edition, John Wiley & Sons Inc., Canada, 1995.

[39] T. Kalker, A. Ansgar, U. Jensen, "*Novel Digital Controller for Induction Machines Considering the Inverter Switching Times and a Fluctuating DC-link Voltage*" , Proceedings of European Conference on Power Electronics and Applications (EPE), Florence, Italy, 1991, pp 2.058- 2.063.

[40] F. Moynihan, "*Fundamentals of DSP-Based Control for AC Machines*", Seminar notes, International Conference on Power Quality and Intelligent Motion (PCIM), Nurnberg, Germany, 1998.

[41] S. Alahakoon, "*Digital Motion Control Techniques for Electrical Drives*", PHD Thesis, Royal Institute of Technology, Department of Electric Power Engineering, Electrical Machines and Power Electronics, Stockholm, 2000

[42] L. Zhong et al., "*A direct torque controller for permanent magnet synchronous motor drives*", IEEE Transactions on Energy Conversion, Vol. 14, No. 3, September 1999

[43] M. S. Merzoug, and F. Naceri, "*Comparison of Field-Oriented Control and Direct Torque Control for Permanent Magnet Synchronous Motor (PMSM)*", World Academy of Science, Engineering and Technology 21 2008

[44] K. Florek, "*SoPC dans une FPGA avec Ethernet haute vitesse*", Bachelor of Science, Haute Ecole d'Ingégnerie et de Gestion du Canton de Vaud, 2014