POLITECNICO DI TORINO

Master of Science
in Automotive Engineering

Master of Science's Thesis

DESIGN OF A VIRTUAL RESOLVER BASED ON HALL EFFECT SENSORS FOR THE MOTOR-IN-WHEEL SOLUTION

Supervisor

Prof. Stefano Carabelli

Candidate

Adelson Santos da Silva Júnior

July 2018

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to God, my Lord, for whom words are insufficient for his care and, above all, for his love and salvation. His promises and designs never fail, his light is a guide for all those who seek him.

My companion, Myths, for what we have lived in the last years, support and help in the most difficult moments and the smiles and hugs in the always present joy, you are my world, and nothing would be possible without you by my side.

To my parents, Adelson and Fatima, who have never measured their efforts throughout my life, for love, care, education and support preparing, accompanying and assisting me on the road of life.

My grandmother Amara (in memoriam) was my second mother, and I thank God for having her for so long by my side. I miss her.

To my grandparents Sérgia and Jurandir, for always believing and waiting for my best.

My sisters, for the incentive and cheering even when I was away.

My family, uncles, aunts, cousins, cousins, father-in-law, sister-in-law and brother-in-law by the significant importance on this journey.

To my friends Juliano and Lucélia, my parents while in Turin.

To the exchange partners, for the friendship and unique moments.

I thank my supervisor during this thesis, Stefano Carabelli, for his important support.

Also of great importance was the help of Milead KarimShoushtari, whose guide was a brilliant lighthouse to carry out this project.

My thanks also go to the Ladispe staff that provided a place to work and they were always willing to help.

Last but not least, all those who contributed directly and indirectly to the realization of this thesis and especially those who prayed and continue praying for my life and my family.

"That all may see, and know, and consider, and together understand that the hand of the Lord hath done it ..." Isa 41.20

ABSTRACT

The great growth in the last years in the development, in the industry and in the market of electric cars, taking into account its variety of topologies and architectures makes necessary the development of unconventional technologies with regard to the automotive sector.

One of these technological solutions is called the motor-in-wheel, a design of a permanent magnet synchronous electric motor located inside the ring of each wheel of the vehicle. It acts as the vehicle's traction system, leaving aside the need for mechanical transmission elements such as shaft and differential gears, resulting in simplification of design and reduction of production and maintenance costs.

However, it is notable that such solutions imply the need to develop technologies more specific to the new applications. Among them, the traction and power control, new driving techniques, data acquisition and sensing.

As a means of innovation in this context, the *Risciò* project aims to develop and build a light electric vehicle in practice by applying new technological solutions such as the motor-in-wheel. The project started three years ago, and other works were developed over that time, including the implementation of a One Pedal Drive and concerns about unsprung masses.

Thus, the development of electronics and embedded systems play a crucial role at this point as well as the development of software and, today, improved design technologies.

The V-Model of development is an effort in this direction. This is composed by phases, among them: requirements acquisition and their analysis, high level and detailed modeling, implementation, unit tests, integration tests and validation.

As part of the *Risciò* Project, this thesis focuses on the design of a Virtual Resolver (angular position and speed estimator) based on Hall effect sensors for the *Electronic Kontrol Unit* of the vehicle on replacement of a non-implementable physical resolver on the motor-in-wheel design and configuration.

Generally, encoders (rotational pulse transductors) and resolvers (rotational analogic transductors) must be mounted directly on the motor’s shaft and their mechanical integration increases the last price of these elements. However, on regards the proposed motor-in-wheel, its design makes, at instance, the integration or mechanical coupling with a shaft and thus with a physical encoder or resolver hardly achievable.

Therefore, Hall effect sensors are used as edge-field detectors. The Hall sensor board is mounted on the open stator available space for the brake calliper, hence, it can be positioned and do not have any interference from the induced magnetic field of the stator. Thus, the sensors detect the changes on the magnetic field on each passage of the magnets of the rotor. After that, a signal processing is performed to estimate the angular position and speed.

After the problem situation being introduced with the proposed solution, this work is divided in three main parts. Firstly, the Hall sensors signal acquisition and processing are presented. Then, a first virtual resolver is proposed, followed by the Adaptive Notch Filter, the Phase Locked Loop and the Speed Estimator discussion. After that, models and simulations are developed tuning the system to the automatic code generation and programming. Finally, the hardware-in-the-loop tests are performed, and the results and conclusions are discussed.

Four main systems compound the overall Virtual Resolver: An Acquisition and pre-processing system, a Speed Estimator, an Adaptive Notch Filter and a Phase Locked Loop.

The discussion about the acquisition and pre-processing system takes into account the Analog-to-Digital converser of the *Electronic Kontrol Unit*. Moreover, the shapes and frequencies of the Hall sensor board signals are considered to perform a proper sense of the environment. The irregular nature of the three phase signals is evaluated for future considerations. Finally, a Low Pass Filter is introduced on the first conditioning subsystem.

The Speed Estimator algorithm is an important part of the entire system. It is responsible to provide reliable information about the velocity of the vehicle. Hence, other systems of the Virtual Resolver can work properly with the speed feedback. Different models of estimation are proposed and evaluated.

As a direct sequence of the acquisition system, an Adaptive Notch Filter is responsible for filter out the third-order harmonics of the Hall sensor signals. Such a filter subtracts the exactly harmonic of the input signal by means of the estimation of the Fourier coefficients. However, the name Adaptive comes from the fact that it needs a feedback information of the estimated angular position to close the loop. The attenuation of the unwanted third-order harmonic noise adds accuracy to the final angular position estimation as well.

Thence, a Phase Locked Loop works with the Adaptive Notch Filter closing the overall loop. The Phase Locked Loop calculates the angular position and velocity at its input. Its principal components are the Phase Detector, that uses sine and cosine arithmetic with some feedback information, and the Loop Filter, that is responsible to determine the dynamic behavior of the system filtering out the high frequency ac components from the Phase Detector output. Finally, an integrator is used to estimate the angular position.

Following the V-cycle software development model, the modelling, simulation and comparison of the proposed systems play a fundamental role on the overall process and is performed on *Matlab-Simulink* environment. Besides the identification of bugs and errors in the modelling, the simulations of each system model and the overall Virtual Resolver are also important on tuning the models in order to generate a better improved code. Thus, they were performed taking account speed inputs with physical constraints and a real urban cycle as references.

Changes on the Speed Estimator, the locations of the Phase Locked Loop poles and Controllers improve the results decreasing the error estimation. The Phase Locked Loop needs an accurate frequency input in order to follow properly the phase of the signals, as well the location of its poles guarantees a larger range of lock. Finally, Controllers are used to control the phase displacement caused by the system transfer functions.

Then, in order to implement the systems on the *Electronic Kontrol Unit* board that does not work with the continuous-time approach, model discretization is necessary. Therefore, *Matlab* tools as the c2d command changes filters and controllers continuous transfer functions into discrete ones.

Afterwards, automatic code generation using the *Simulink* tools and handwritten programming by the *Code Composer Studio v6* implements the signal processing system on the *Risciò Electronic Kontrol Unit.*

The final steps are to perform real tests of the hardware and software implementations. Hardware-in-the-loop tests are performed using Development boards with the same Digital Signal Processor as the *Electronic Kontrol Unit* board of the *Risciò*. Hence the results are well depicted, accomplishing the main objectives of this thesis and preparing the project to future works.

Summarizing, the listed objectives are well accomplished in this thesis. The proposed systems and algorithmics presents reasonable results both for simulations and Hardware Loop tests. Once the problem situation is solved by the Virtual Resolver approach, the resulting code application is useful to be implement as part of the *Risciò* EKU system and makes part of its innovative solutions for electrical vehicles.

RESUMO

Com o grande crescimento nos últimos anos no desenvolvimento, na indústria e no mercado de carros elétricos, levando em consideração sua variedade de topologias e arquiteturas se faz necessário o desenvolvimento de tecnologias não convencionais no que se refere ao setor automotivo.

Uma dessas soluções tecnológicas é chamada de motor-in-Wheel, um projeto de um motor elétrico síncrono de imas permanentes localizado dentro do anel de cada roda do veículo. Este atua propriamente como o sistema de tração do veículo deixando de lado a necessidade de elementos mecânicos de transmissão tais como engrenagens eixo e diferencial, resultando numa simplificação de projeto e redução de custos de produção e manutenção.

No entanto, é notável que tais soluções implicam na necessidade do desenvolvimento de tecnologias mais especificas para as novas determinadas aplicações. Entre elas estão o controle de tração e potência, novas técnicas de direção, aquisição de dados e sensoriamento.

Como um meio de inovação nesse contexto, o projeto *Risciò* tem o objetivo de desenvolver e construir na prática um veículo elétrico leve aplicando-se novas soluções tecnológicas como por exemplo o motor-in-wheel. O projeto teve início três anos atrás e outros trabalhos foram desenvolvidos ao longo desse tempo, incluindo a implementação de um algoritmo de Direção com um pedal.

Assim, a eletrônica e o desenvolvimento de sistemas embarcados têm um papel crucial nesse ponto assim como o desenvolvimento de softwares e, atualmente, tecnologias aprimoradas de design.

O Modelo-V de desenvolvimento é um esforço nesse sentido. Este é composto de fases, entre elas: a aquisição e análise de requerimentos, modelagem em alto nível e detalhado, implementação, testes unitários, testes de integração e validação.

Como parte do Projeto *Risciò*, esta tese enfoca o projeto de um Resolver Virtual (estimador de posição e velocidade angular) baseado em sensores de efeito Hall para a Unidade Eletrônica de Controle do veículo em substituição de um resolver físico não implementável no projeto e configuração do motor-in-wheel.

Geralmente, os encoders (transdutores rotacionais de pulso) e os resolvers (transdutores rotacionais analógicos) devem ser montados diretamente no eixo do motor e sua integração mecânica aumenta o preço final desses elementos. No entanto, no que diz respeito ao proposto motor-in-wheel, seu design faz com que, por exemplo, a integração ou acoplamento mecânico com um eixo e, portanto, com um encoder físico ou resolver seja dificilmente alcançável.

Portanto, sensores de efeito Hall são usados como detectores de campo de borda. A placa do sensor Hall é montada no espaço aberto do estator aberto para a pinça de freio, assim, pode ser posicionada e sem interferência do campo magnético induzido do estator. Assim, os sensores detectam as mudanças no campo magnético em cada passagem dos ímãs do rotor. Depois disso, um processamento de sinal é realizado para estimar a posição e a velocidade angular.

Após a situação problemática ser introduzida com a solução proposta, este trabalho é dividido em três partes principais. Primeiramente, a aquisição e o processamento do sinal dos sensores Hall são apresentados. Em seguida, é proposto um primeiro resolver virtual, seguido

pelo Filtro Notch Adaptável, pelo Phase Locked Loop e pela discussão do Estimador de Velocidade. Depois disso, modelos e simulações são desenvolvidos ajustando o sistema para geração automática de código e programação. Finalmente, os testes hardware-in-the-loop são executados e os resultados e conclusões são discutidos.

Quatro sistemas principais compõem o Resolver Virtual: Um sistema de Aquisição e pré-processamento, um Estimador de Velocidade, um Filtro Notch Adaptável e um Phase Locked Loop.

A discussão sobre o sistema de aquisição e pré-processamento leva em consideração o conversor Analógico-Digital da Unidade Eletrônica de Controle. Além disso, as formas de onda e frequências dos sinais da placa do sensor Hall são consideradas para uma percepção adequada do ambiente. A natureza irregular dos sinais trifásicos é avaliada para considerações futuras. Finalmente, um filtro Passa Baixas é introduzido no subsistema de pré-condicionamento.

O algoritmo do estimador de velocidade é uma parte importante de todo o sistema. É responsável por fornecer informações confiáveis sobre a velocidade do veículo. Assim, outros sistemas do Virtual Resolver podem funcionar corretamente com o feedback de velocidade. Diferentes modelos de estimação são propostos e avaliados.

Como uma sequência direta do sistema de aquisição, um Filtro Notch Adaptável é responsável por filtrar os harmônicos de terceira ordem dos sinais do sensor Hall. Tal filtro subtrai exatamente o harmônico do sinal de entrada por meio da estimativa dos coeficientes de Fourier. No entanto, o nome Adaptável vem do fato de que ele precisa de uma informação de feedback da posição angular estimada para fechar o loop. A atenuação do ruído harmônico de terceira ordem indesejado também adiciona precisão à estimativa final da posição angular.

Logo, um Phase Locked Loop funciona com o Filtro Notch Adaptável fechando o loop geral. O Phase Locked Loop calcula a posição angular e a velocidade da entrada. Seus componentes principais são o detector de fase, que usa aritmética de seno e cosseno com algumas informações de feedback, e o filtro de loop, que é responsável por determinar o comportamento dinâmico do sistema filtrando os componentes de alta frequência da saída do detector de fase. Finalmente, um integrador é usado para estimar a posição angular.

Seguindo o modelo de desenvolvimento de software do ciclo V, a modelagem, simulação e comparação dos sistemas propostos desempenham um papel fundamental no processo global e é realizada no ambiente *Matlab-Simulink*. Além da identificação de bugs e erros na modelagem, as simulações dos modelos de cada sistema e do Virtual Resolver global são também importantes no ajuste dos modelos para gerar um código melhorado. Assim, eles foram realizados considerando as entradas de velocidade com restrições físicas e um ciclo urbano real como referências.

Alterações no Estimador de Velocidade, nos locais dos polos do Phase Locked Loop e Controladores melhoram os resultados diminuindo o erro de estimativa. O Phase Locked Loop necessita de uma entrada de frequência precisa para acompanhar adequadamente a fase dos sinais, assim como a localização de seus polos garante uma maior faixa de travamento. Finalmente, os Controladores são usados para controlar o deslocamento de fase causado pelas funções de transferência do sistema.

Então, para implementar os sistemas na placa da Unidade Eletrônica de Controle, que não funciona com a abordagem de tempo contínuo, é necessária a discretização do modelo. Portanto, as ferramentas do *Matlab*, como o comando c2d, transformam as funções de transferência contínuas dos filtros e controladores em funções discretas.

Depois do que, a geração automática de código usando as ferramentas do *Simulink* e a programação escrita à mão no *Code Composer Studio v6* implementa o sistema de processamento de sinais na *Unidade Eletrônica de Controle* do *Risciò*.

As etapas finais são a execução de testes reais das implementações de hardware e software. Testes de Hardware-in-the-loop são feitos usando placas de desenvolvimento com o mesmo Processador Digital de Sinais da placa da *Unidade Eletrônica de Controle* do *Risciò.* Assim, os resultados são bem mostrados, cumprindo os objetivos principais desta tese e preparando o projeto para trabalhos futuros.

Resumindo, os objetivos listados são bem cumpridos nesta tese. Os sistemas e algoritmos propostos apresentam resultados razoáveis tanto para simulações como para testes de Hardware. Uma vez que a situação problema é resolvida pela abordagem do Virtual Resolver, o código resultante é útil para ser implementado como parte do EKU do *Risciò* e faz parte de suas soluções inovadoras para veículos elétricos.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

Vehicles based on Electric and Hybrid Powertrain are increasingly present in our days and have long ceased to be the technology of the future to be the one we are using these days. Among several factors, concerns about air quality in urban centers or the availability of fossil resources or fuels contribute to the growing demand for alternatives to older traction systems based on internal combustion engines.

This demand is reflected in the vehicle market in recent years, modifying the production chain and even securing a portion of the sector to fresh players. Thus, different architectures and models of electric and hybrid powertrain systems have been extensively studied and explored in the most diverse applications.

A trade-off must be found between the advantages and drawbacks of electric vehicles, including the different applications of its topologies. Hence, Light Electric Vehicles come as an alternative for the urban environment thanks its autonomy range, power, maximum velocity, charging time and dimensions that are suitable for it.

A possible architecture is presented on the *Risciò* Project. That aims to develop and build a light electric vehicle in practice applying new technological solutions and techniques.

The project has been developed by *AIPA (Company Association for Automotive Prototypes)* in collaboration with sector leaders as *Cecomp, Italtecnica, All Design*, and *Actua* located in Turin (Italy). It was proposed for EU L7e category, but similar characteristics were considered in order to be homologate in countries that do not have an L7-like category.

Figure 1 - Risciò Rolling Chassis Back View

The “L category” limits the vehicle performance and technical characteristics aiming urban vehicles and supplying the request for less energy consumption addressed by the electric vehicles. Besides the advantages of being smaller, comfortable and suitable for congested cities with scarce parking space.

At the beginning of this Thesis, the *Risciò* Project presented many technical solutions already designed, modelled and realized such as the light weight chassis structure, powertrain, suspension, under body, energy storage system and the actuation control unit. Some of these solutions are shown on the Figure *1* and *2*.

Figure 2 - Risciò Rolling Chassis Front View

A significant effort was done to reduce the weight of the vehicle. The total weight is to be less than 700 kg including battery. Hence, its upper body and interior design were meticulously intended to be robust and essential, considering the involved shapes and materials as well.

Besides the weight reduction solutions, the powertrain system brings forward another innovative technology. Two main topologies are presented for the *Risciò* electric traction system.

The first one, a high efficiency transaxle based on a differential/reduction gearbox, a high efficiency synchronous electric motor, and an automotive certifiable control and power electronics.

The second one, and the focus of this work, the Open Stator Motor-in-Wheel that is based on a permanent magnet synchronous machine inside the wheel rim. Thus, each wheel is a separately traction system that can work independently.

The motor-in-wheel is characterized by the high efficiency provided on the traction system. It can be thanks the permanent magnet electric motors having a larger efficiency when compared with induction three phase and brushless DC motors.

Moreover, the elimination of the transmission system makes the design as simple as possible. The mechanical efficiency is increased once no transmission elements need to be used as the gearbox and the shaft rotating parts. It is reflected as well in the maintenance costs and execution, since the powertrain presents a minimum number of parts.

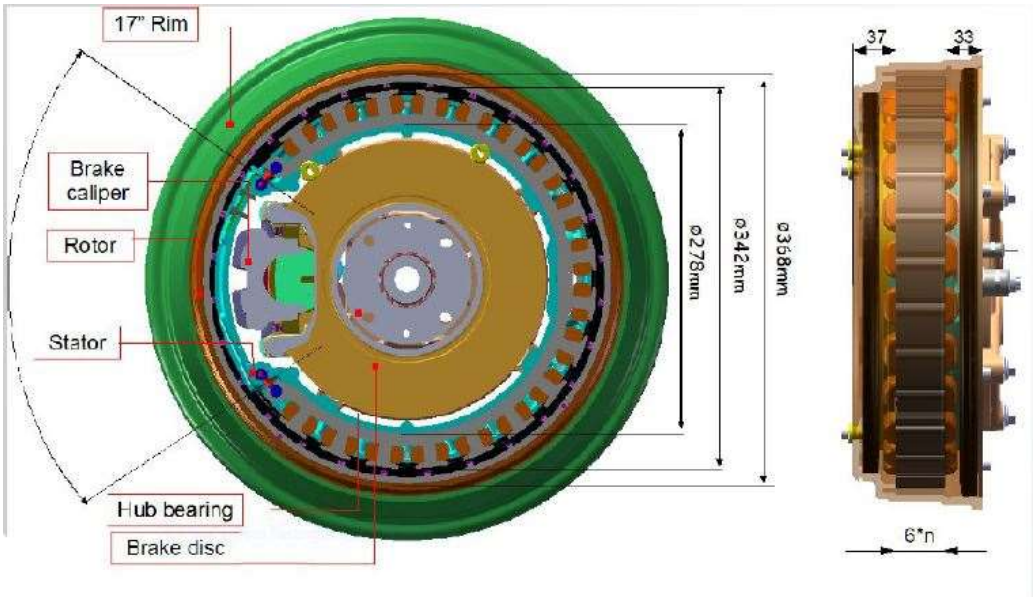

Figure 3 - Motor-in-wheel structure

Another problem is solved on regards the refrigeration of the motor. The air flux through the wheel is used to cool down the electric motor during the working cycle without problems. It makes the motor-in-wheel does not depending on a dedicated refrigeration system.

The above advantages bring forward a fourth one about the space constraints. The *Risciò* must to be as light and compact as possible; hence, the motor-in-wheel solution eliminates unnecessary accessories and provides a large available space, transferring the powertrain from the vehicle chassis directly to the wheel.

The latter brings on the surface a drawback that needs to be solved posteriorly. There is a great transference of mass from the vehicle chassis to the wheels that increases significantly the unsprung mass of the vehicle. Hence, a special attention is required on its suspension design.

The Figures 3 and 4 shows the motor-in-wheel structure.

However, the split topology allows the implementation of an Integrated Differential Electrical Axle for curves, conversions and anti-slip purposes increasing the vehicle active safety capability. The actuation kontrol unit can deliver for each wheel a different torque based on the system needs or requests, even different sign of torque can be exploited by the short vehicle in an urban environment.

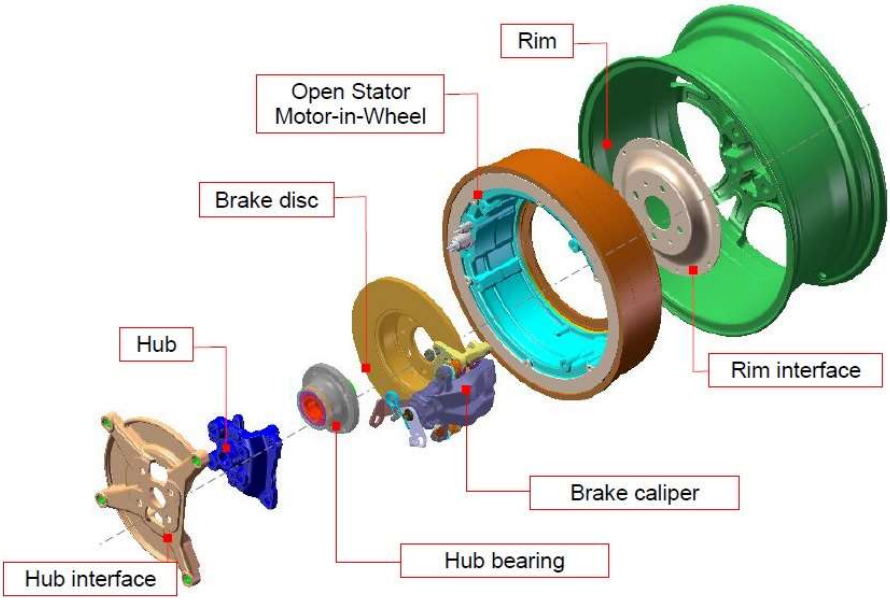

Figure 4 - Motor-in-wheel exploded view

Furthermore, some considerations can be done on regards the electric motor architecture. Firstly, the rotor involves the stator from the outside in an inverse structure. The stator support is connected by the hub to the vehicle suspension while the rotor support is linked to the wheel rim.

The second relevant remark related to the motor architecture is the "Open" configuration. In order to fill in at the maximum the space inside of a wheel rim with the brake caliper, the stator needs to be open, or in other words the stator does not form a complete circle, there are some windings gap containing the caliper.

The design of an appropriated control system for the Motor-in-wheel solution is fundamental to guarantees the advantages of its innovative project. Hence, this thesis will study a solution for its sensing system, taking into account its non-usual and odd design.

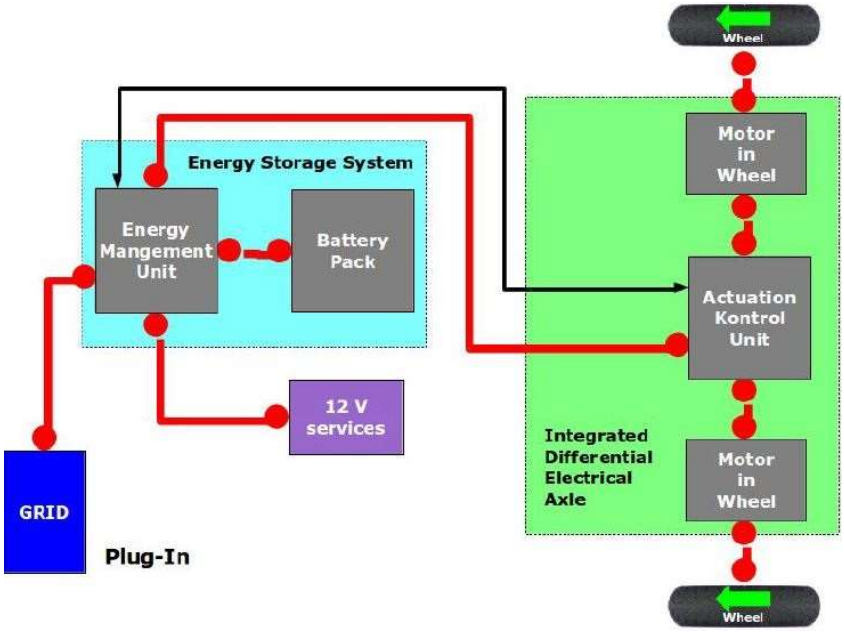

Figure 5 - Motor-in-wheel Powertrain block diagram

CHAPTER 2

THESIS OBJECTIVES

2. THESIS OBJECTIVES

The development of this thesis has as main goals the points listed below:

1. To design a Virtual Resolver (angular position and speed estimator) based on Hall effect sensors for the *Electronic Kontrol Unit* of the *Risciò* Project on replacement of a non-implementable physical resolver on the PMSM motor-in-wheel design and configuration.

2. To interface real signals using the *Electronic Kontrol Unit* of the vehicle, reading and conditioning the Hall sensors signals.

3. To add accuracy on the angular position estimation of the resolver by means of an Adaptive Notch Filter to attenuate the unwanted third-order harmonic noise that are present on the Hall sensors signals.

4. To include a Phase Locked Loop on the virtual resolver to work with the Adaptive Notch Filter increasing reliability.

5. To perform modelling, simulation and comparison of the proposed systems on *Matlab-Simulink* environment.

6. Automatic code generation (*Simulink* tools) and software programming (*Code Composer Studio v6*): to implement the signal processing that includes the pre-processing conditioning, the Adaptive Notch Filter and the Virtual Resolver (angular position and speed estimator) on the *Risciò Electronic Kontrol Unit.*

7. To perform real tests of hardware and software implementations: before the hardware-in-the-loop test can be done, all the signals must be readable, and the implemented algorithms must be written and run properly.

This thesis work was structured in such a way to accomplish the above main objectives, following almost completely the order in which they appear. Firstly, the problem situation is introduced with the proposed solution. Next, the Hall sensors signal acquisition and processing are presented. Then, a first virtual resolver is proposed, and it is followed by the ANF, the PLL and the speed estimator discussion. After that, models and simulations are developed tuning the system to the automatic code generation and programming. Finally, hardware-in-the-loop tests are performed, and the results and conclusions are discussed.

CHAPTER 3

PROJECT SITUATION

3. PROJECT SITUATION

3.1. Base configuration - Motor-in-wheel

The motor-in-wheel solution on the *Risciò* Project presented in the introduction of this thesis is based on a Permanent Magnets Synchronous Motor. It is interesting to mention that the advantages of these machines compared with Brushless DC motors such as higher efficiency and lower torque ripple are also due to the different control techniques applied on PMSM.

These techniques or methods of control require feedback information about the rotor absolute angular position in order to develop the control itself and to apply correctly the amount of energy at the right instance. Usually this information comes from encoders or resolvers on the most commonly applied systems, besides sensorless control techniques.

Considering the first ones, the encoders can provide a high resolution angular position feedback and are based on counting pulses or, on absolute encoders, generates a specific binary code for each angular position. Diverse topologies can be found such as: absolute or incremental, optical, magnetic and magnetoresistive encoders.

In the case of the resolvers, the resolution is even higher, and they are used on high precision systems. The sensing principle is based on the generation of two signals that are identified as sine and cosine of the position angle. Generally, two poles resolvers are used, although multiple poles (2*p) resolvers are applied on multiple poles motors.

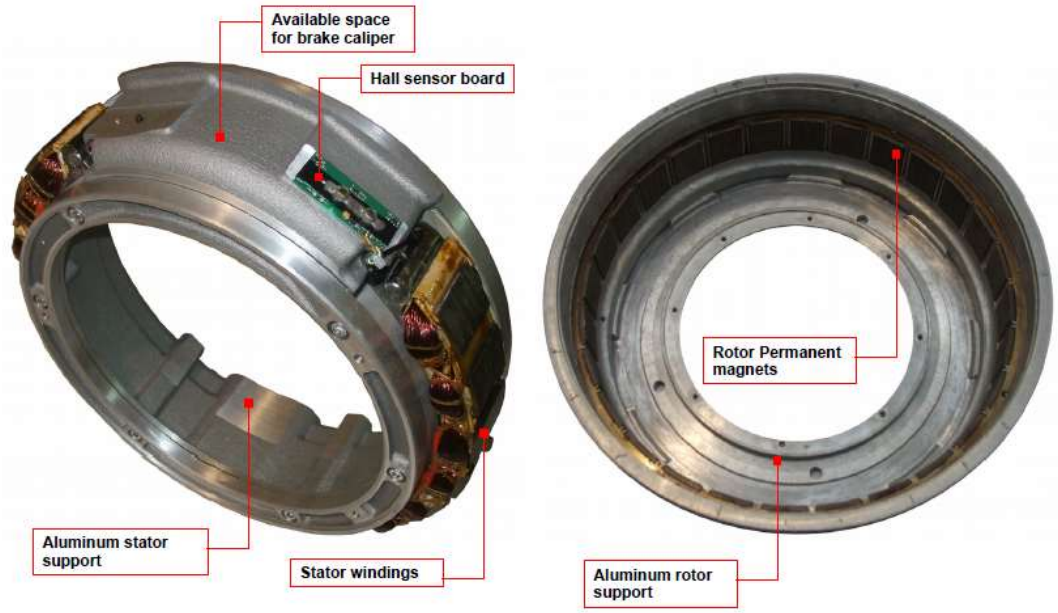

Figure 6 - Motor-on-Wheel electric motor parts and Hall sensor board position

Encoders and resolvers must be mounted directly or by means of gears on the motor's shaft and their mechanical integration increases the final price of these elements. At this point, sensorless techniques are not applicable due to the wild range of speeds and load changes on automotive field.

However, on regards the proposed solution on the *Risciò* project, the motor-in-wheel

has a particular and interesting characteristic. Its design makes, at instance, the integration or mechanical coupling with a shaft and thus with a physical encoder or resolver hardly achievable, besides the disadvantage of making the project more expensive.

Therefore, another configuration is introduced to overcome these drawbacks. In this one, Hall effect sensors are used as edge-field detectors. Moreover, without the shaft structure and for the proposed objectives it is not possible to introduce the generally used configuration of rpm hall sensors as part of the system. Thus, the Hall sensor board must be mounted on the stator body itself. Fortunately, the open stator of the motor-in-wheel solution has a gap, an available space for the brake calliper where the sensor can be positioned and do not have any interference from the induced magnetic field of the windings of the stator.

The Hall board is responsible for identifying the changes on the magnetic field on each passage of the equally spaced permanent magnets of the rotor. These changes produce a variation on the induced voltage on the output of the sensor. Thus, the sensors detect the edges that are subjected to a signal processing in order to estimate the angular position.

The different Hall sensors have a wide range of output signals such as: digital, analog, differential and single and three-phase signals. The Hall sensor board and system are not finally defined; however, the three-phase configuration is preferable and that one will be used in the next sections.

Other considerations can be done such as: the positioning of the board on the stator gap, the mechanical angle or distance between the sensors on the board and the quality of the sensors output signal, that has a significant role on angular position estimation. These considerations will be properly discussed in the next sections as well.

3.2. *Electronic Kontrol Unit* – EKU

Before the Virtual Resolver discussion, it is necessary to introduce the work instrument. The proposed virtual resolver solution will be implemented on the *Risciò Electronic Kontrol Unit* as part of the code already implemented. Such *EKU* is hardware-based on the Texas Instruments C2000 Real-time DSPs that uses a central 32-bit CPU core, called C28x, coupled with a highly-optimized peripheral and interrupt management bus. Between its peripherals, it presents high resolution PMW generators and AD Converters.

The Analog to Digital Convertor plays a significant role on the *EKU*. Analog signals are generated by the sensors and must be converted into digital discrete data to be processed by the DSP. The availability of this information must be as fast as possible so that it can be performed a real-time control. Thus, the specific sensors AD Conversion for the Virtual Resolver routines are executed on a narrow interruption routine. Finally, the ADC references voltages are 0 and 5V and it has a resolution of 12 bits; hence, adequate gains and offset must be configured in a pre-processing system.

Lastly, the Virtual Resolver and its accessory routines will be automatic generated by the *Matlab-Simulink* tools and/or written on the *TI Code Composer Studio v6* to be guaranteed a satisfactory performance and perfect match with the already developed *EKU* interface and applications.

CHAPTER 4

VIRTUAL RESOLVER

4. VIRTUAL RESOLVER

Resolvers are basically rotating electric transformers that converse the rotational movement on electric signals used to give information about the angular position of a shaft. On other words, resolvers are angle transducers mounted on a motor shaft to get its absolute angular position. Such mechanical coupling is not possible on motor-in-wheel due to the technical design issues discussed previously. Thus, this section elaborates a Virtual Resolver, an alternative way to get the angular position without the unrealizable physical resolver.

Moreover, since this Virtual Resolver is based on a three-phase Hall sensor board, next sections will not only considerate the angle estimation working principle of resolvers, but a brief revision of Hall sensors theory will be shown, and the signal acquisition process is going to be explained with a certain attention.

4.1. Working Principle

As electric transformers, resolvers are made of coils, generally, three of them: one rotor coil that is excited by a fixed frequency source and two others coils on the stator, these ones are 90-degrees spaced on the armature to create an orthogonal configuration.

Thus, when the shaft rotates the rotor coil induces on the coils of the stator two 90-degrees phase shifted sinusoidal output signals. In fact, these signals are an amplitude modulation of the excitation signal input and this modulation can be taken as the sine and cosine of the mechanical shaft angle.

Therefore, resolvers are able to provide information about the absolute angular position of a rotating shaft by means its sine (x_β) and cosine (x_α) waveforms.

4.2. Angle Estimation

A simple method of angle estimation from resolvers' signals can be introduced. The resolver-to-digital conversion can be implemented on DSPs boards and includes the well-known Analog to Digital conversion. The demodulation process requires a little more attention on sampling theory and can be used together with the ADC sampling, however, it is not the subject of this document. Nonetheless, it is important to mention that undersampling and oversampling methods can be used to get the sine and cosine signals and to determine the accuracy of the estimation.

Once the signals x_α and x_β are demodulated, the angular position can be calculated by an inverse tangent function of the relation:

$$\frac{\sin(\theta)}{\cos(\theta)} = \frac{x_\beta}{x_\alpha} \tag{1}$$

Considering, obviously, the four-quadrant tangent principle, the estimated angular position can be found according to the system below.

$$\hat{\theta} = \begin{cases} \arctan\left(\frac{x_\beta}{x_\alpha}\right), & if\ x_\alpha \geq 0\ and\ x_\beta \geq 0 \\ \pi + \arctan\left(\frac{x_\beta}{x_\alpha}\right), & if\ x_\alpha < 0 \\ 2\pi + \arctan\left(\frac{x_\beta}{x_\alpha}\right), & if\ x_\alpha \geq 0\ and\ x_\beta < 0 \end{cases} \quad (2)$$

An overall scheme of this angle estimator is shown in Figure *7*.

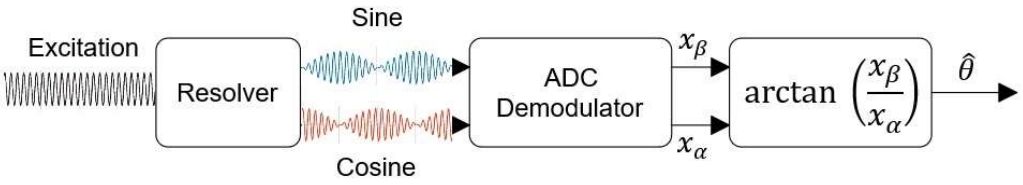

Figure 7 - Angle estimator scheme

It is important to emphasize that this scheme considers the angle estimation as from the signals that are provided by a physical resolver, namely the modulated signals by the sine and cosine waveforms of the rotor angle. Since the proposed Virtual Resolver is based on a three-phase configuration of Hall Sensors instead the original physical resolver, it is relevant to do a brief discussion about the related transformations that are applicable.

4.3. From three-phase signals to sine and cosine waveforms

One first consideration can be done on regards the three phase signals. These signals are three sinusoidal waves shifted on phase by 120-degrees between them, thus the sum of the three signals at any instance is zero. Therefore, let's consider the signals as:

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = H \begin{bmatrix} \cos(\omega t) \\ \cos(\omega t - (2\pi/3)) \\ \cos(\omega t - (4\pi/3)) \end{bmatrix} \quad (3)$$

Another representation that is commonly used is a dc system in a rotating reference frame. This transformation is shown on Figure *8*.

In order to perform the above transformation, a first step is necessary. The so-called Clarke transform take the projections of the three signals on an orthogonal axis and transforms the three-phase signals into an orthogonal component system (alpha, beta). Thus, two signals are given by this stationary reference frame. According to Equation 4, these signals are the cosine and the sine of the angle $\theta(t) \equiv \omega t$.

$$\begin{bmatrix} H_\alpha \\ H_\beta \\ H_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & \cos(2\pi/3) & \cos(4\pi/3) \\ 0 & \sin(2\pi/3) & \sin(4\pi/3) \\ (1/2) & (1/2) & (1/2) \end{bmatrix} \times \begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \\ 0 \end{bmatrix} H \quad (4)$$

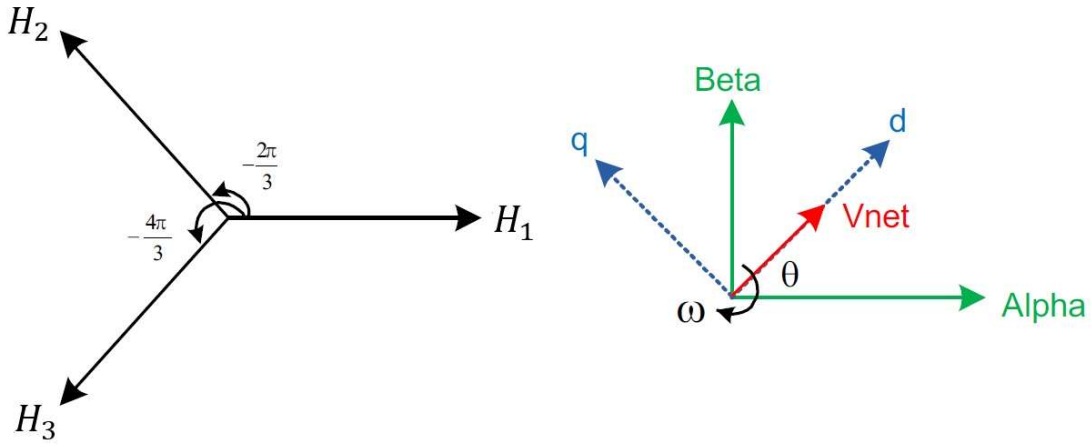

Figure 8 - Transformation from Three Phase to Stationary and Rotating Reference Frame

Therefore, from the three phase Hall sensors signals it is possible to apply the Clarke transform to get $x_\alpha \equiv H_\alpha$ and $x_\beta \equiv H_\beta$ in order to be used on the angle estimator previously explained.

Finally, it is worth noting that the transformation into the dc system is not complete at this point. The second step would be to perform the Park transform. This second transformation will be presented later this thesis due to its relevance as the phase detector on PLL systems.

4.4. Hall Effect Sensors

On this section, a brief introduction on the theory behind the Hall effect sensors takes place and the signal acquisition process will be explained.

4.4.1. Sensing principle and theory

The American physicist Edwin Hall discovered this effect in 1879, attempting to prove the influence of magnets on electric currents. Hall developed an experimental device that is very similar what is found on the now days' sensors, an electric current passing through a leaf of conductive material embedded on an orthogonally magnetic field, and finally a galvanometer connected across the leaf at equipotential points.

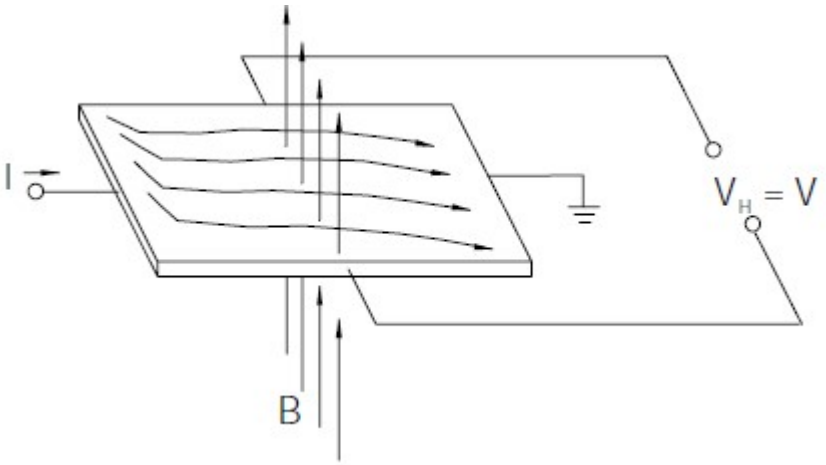

Figure 9 - Hall effect principle

Hall observed an electromotive force in the leaf causing a deflection on the current distribution and consequently a difference of potential (voltage) identified by the galvanometer.

This force, known as Lorentz's force, acts on particles carrying an electrical charge moving in a magnetic field. Considering a conductor in which one current flows, it is possible calculate the Lorentz's force across the leaf.

$$F = l\, \boldsymbol{i}\, \boldsymbol{B} \tag{5}$$

Where, l is the length of the conductor, $\boldsymbol{i}$ the current flowing through it, and $\boldsymbol{B}$ the orthogonal magnetic field.

The Lorentz's force deflects the electrical charges towards one side of the leaf, accumulating them until the force be balanced by the electrostatic repulsion force.

$$F_{el} = -Q\frac{V}{w} \tag{6}$$

Where Q is the flowing charge in the conductor, V is the voltage across the sides of the leaf and w the width of the leaf.

Thus, equalizing the two forces and being $S = wl$ the area of the leaf, the induced voltage between the sides of the leaf is:

$$V = \frac{BSi}{Q} \tag{7}$$

4.4.2. Signal acquisition and pre-processing

The signal acquisition process and its involved issues are described in this section including the placement of the sensors, the positioning of the Hall board itself on the motor body and the quality of the sensors output signal. The scheme on the Figure *10* illustrates that process until a signal pre-processing on the EKU board itself.

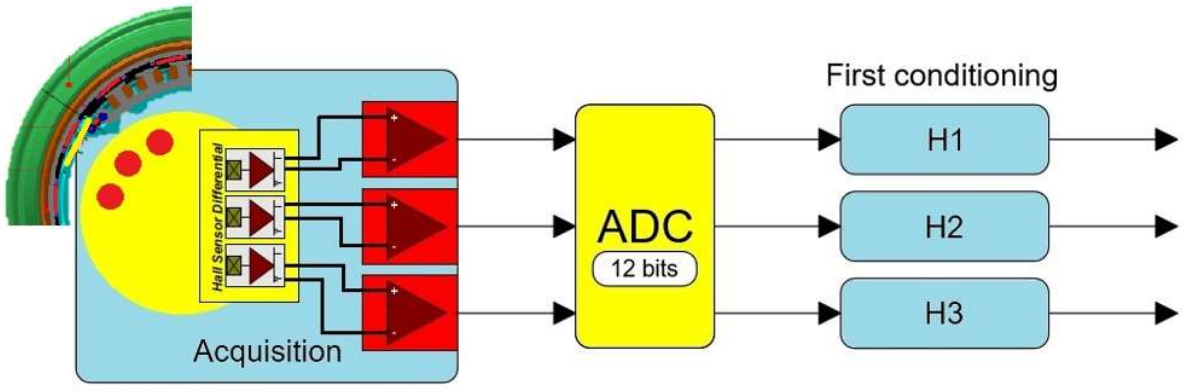

Figure 10 - Signal acquisition process

The chosen Hall board is made of three differential Hall elements that are spaced in such way that each sinusoidal output corresponds in frequency to one of the three-phase driver currents of the motor.

The magnets of the rotor and consequently the rotor itself are driven by the rotational magnet field that are generated on the stator windings. Thus, on the passage of the permanent magnets, the sensor signals are shaped according to the currents by the variable induced voltage of the Hall elements that identifies the changes on the magnetic field where the sensors are embedded.

In order to generate the three-phase signals, the Hall elements must be placed 120-degrees electrically apart. For the sake of the limited available space and considering the multiple pole PMSM, it is necessary to calculate the minimum mechanical angle between the sensors expressed bellow.

$$\theta_m = \frac{\theta_e}{n_p} \tag{8}$$

Where, θ_m is the required minimum mechanical angle, θ_e is the desired electric angle and n_p the number of pole pairs of the selected motor.

Hence, since the motor-in-wheel has 14 pole pairs, the Hall sensors must be placed, approximately, 8.57 degrees mechanically apart.

Therefore, a Hall sensor board is executed in order to accomplish the above requirement, maybe considering a linear topology instead an angular one. The Figure *11* shows this Hall board in a detail of the motor-in-wheel.

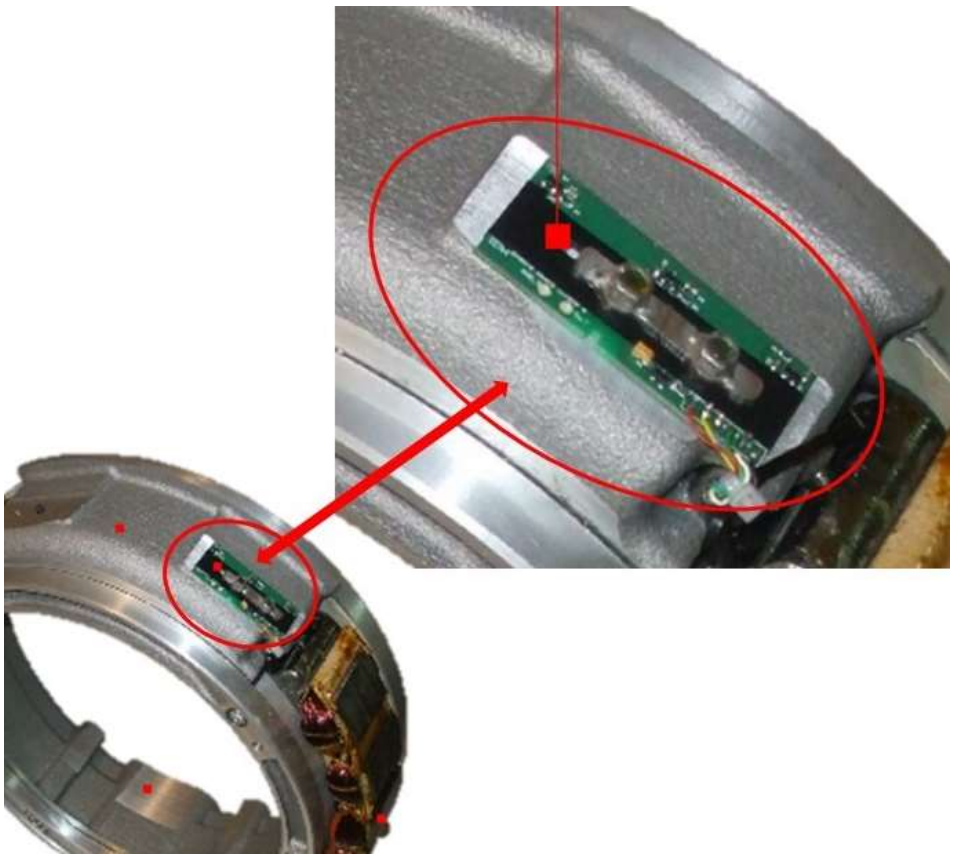

Figure 11 - Hall sensor board in detail

Another discussion is related to the shape of the signals. As was said before, it is

desirable that the sensors produce sinusoidal waveforms. However, the distance between the Hall elements and the permanent magnets affects the final amplitude and form of the signals. The proximity to the magnets causes a field saturation condition, thus a rectangular signal is generated. In contrast, as the distance increases the signal bears a large resemblance to the sinusoidal wave until the limit where the influence of the magnetic field is almost zero.

Therefore, suitable distance and position to the Hall sensor board on the stator were found as result of various experiments. The final shape of the Hall sensors output signals, although the optimization process, unfortunately, is not a perfect sinusoidal wave, giving place to a quality argumentation.

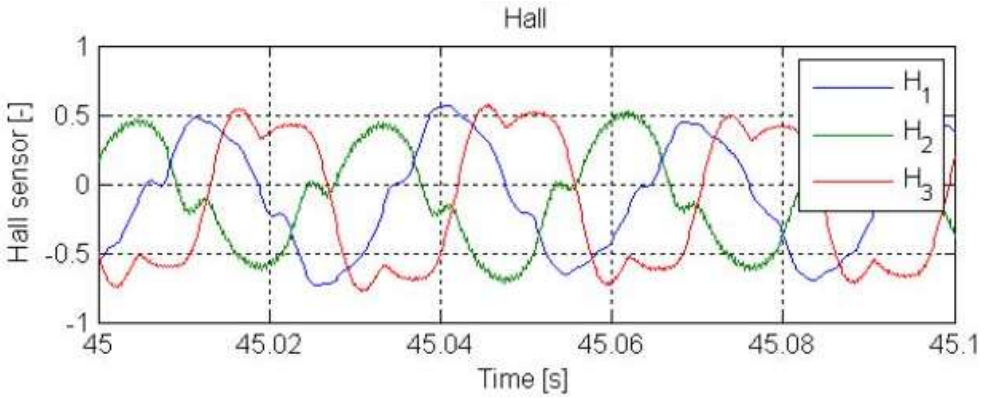

Figure 12 - Three Phase Hall sensors output signals

The three-phase Hall board output signals are depicted on Figure *12*. These signals are, as shown, a composition of different sine and cosine waves with different amplitudes and frequencies. However, the fundamental wave, which corresponds to the motor driver current, is the most relevant and notable component. Moreover, it is important notice that the three signals are different from each other.

Hence, the Clarke transform does not result on perfect sine and cosine waves and so, the angle estimation is compromised.

The estimation accuracy can be increased with the non-fundamental waves filtering and the frequency and phase synchronization. However, since the frequencies are not constant and vary widely, low pass filters are not effective at this point. Therefore, other techniques will be introduced, such as an Adaptive Notch Filter used together with a Phase Locked Loop.

Before that, two last considerations take place: the signal transmission and a pre-processing system.

On regards the first one, the noise cancelation provided by the differential hall sensor topology is enlarged by the shielded six-signal cables that are used there. Furthermore, this configuration aids reliability and robustness relative to the vehicle environment.

Finally, a pre-processing system can be drawn for the DSP board. Taking into account that the differential signals are compared outside the board itself, the EKU inputs would be only three signals for each motor. The EKU has a 12 bits ADC, where the analog signals are converted into quantized integer numbers. Thus, in order to be effectively processed, the signals must be shifted and scaled properly. Then, a low pass filter is configured to the highest desirable frequencies of the signals. The Figure *13* shows the overall pre-processing scheme.

Therefore, considering its resolution and voltage references the gain can be calculated.

$$Gain_{FC} = \frac{2^{12}}{V_{ref+} - V_{ref-}} \cong 820 \tag{9}$$

On regards the Low Pass Filter, a first order Butterworth approximation was chosen to the frequency of 10000 rad/s, to preserve the shape of the signal filtering out only the high frequencies of the noise. Hence, the transfer function of the filter is:

$$LPF_{FC} = \frac{10^4}{s + 10^4} \tag{10}$$

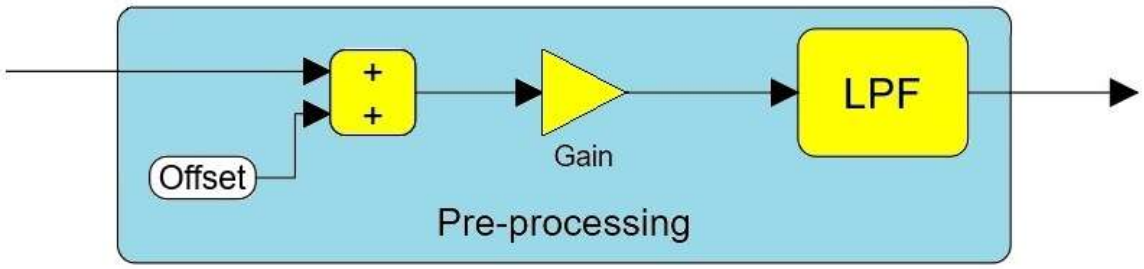

Figure 13 - Pre-processing system

CHAPTER 5

ADAPTIVE NOTCH FILTER

5. ADAPTIVE NOTCH FILTER

On regards the quality discussion initiated previously, the angle estimation produces errors. Therefore, to perform a better estimation, firstly, a notch filter is introduced to cancel the non-desirable secondary components of the signal and thus, to approximate its shape to the sinusoidal waveform.

A notch filter aims to transmit most frequencies without a significant magnitude loss and to attenuate a specific frequency range also known as stop-band. Nevertheless, on the subject system, the unwanted frequency is not constant and varies according to the input fundamental frequency. Hence, the stop-band must follow this frequency and it is necessary to design an Adaptive Notch Filter.

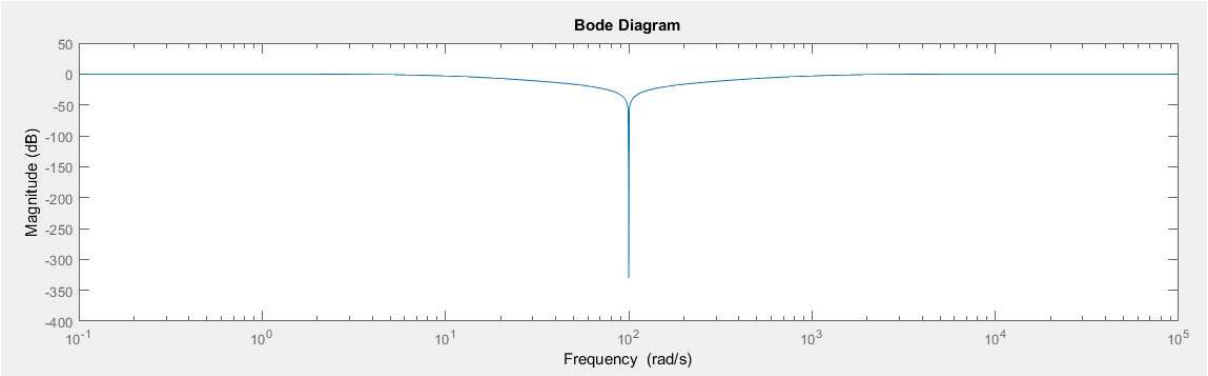

Figure 14 - Frequency response of a notch filter for 100 rad/s

Considering that the not filtered signal is a sum of sines and cosines, a Fourier series can be written for each Hall sensor. Thus, the follow system is generated.

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} H\cos(wt) + \sum H_{1\alpha i}\sin(a_i\omega t) + H_{1\beta i}\cos(b_i\omega t) \\ H\cos(\omega t - \frac{2\pi}{3}) + \sum H_{2\alpha i}\sin\left(a_i(\omega t - \frac{2\pi}{3})\right) + H_{2\beta}\cos\left(b_i(\omega t - \frac{2\pi}{3})\right) \\ H\cos(\omega t - \frac{4\pi}{3}) + \sum H_{3\alpha i}\sin\left(a_i(\omega t - \frac{4\pi}{3})\right) + H_{3\beta i}\cos\left(b_i(\omega t - \frac{4\pi}{3})\right) \end{bmatrix} \quad (11)$$

In order to get only the fundamental terms, it's necessary to identify the involved harmonics. Therefore, the Fourier analysis was performed to characterize the signals on the frequency domain. The results are shown on Figure *15*.

High third-order harmonics are present there. Thus, considering these harmonics, it is possible to resume the previous system to the follow one.

$$\begin{bmatrix} H_1 \\ H_2 \\ H_3 \end{bmatrix} = \begin{bmatrix} H\cos(wt) + H_{1\alpha 3}\sin(3\omega t) + H_{1\beta 3}\cos(3\omega t) \\ H\cos(\omega t - \frac{2\pi}{3}) + H_{2\alpha 3}\sin(3\omega t) + H_{2\beta 3}\cos(3\omega t) \\ H\cos(\omega t - \frac{4\pi}{3}) + H_{3\alpha 3}\sin(3\omega t) + H_{3\beta 3}\cos(3\omega t) \end{bmatrix} \quad (12)$$

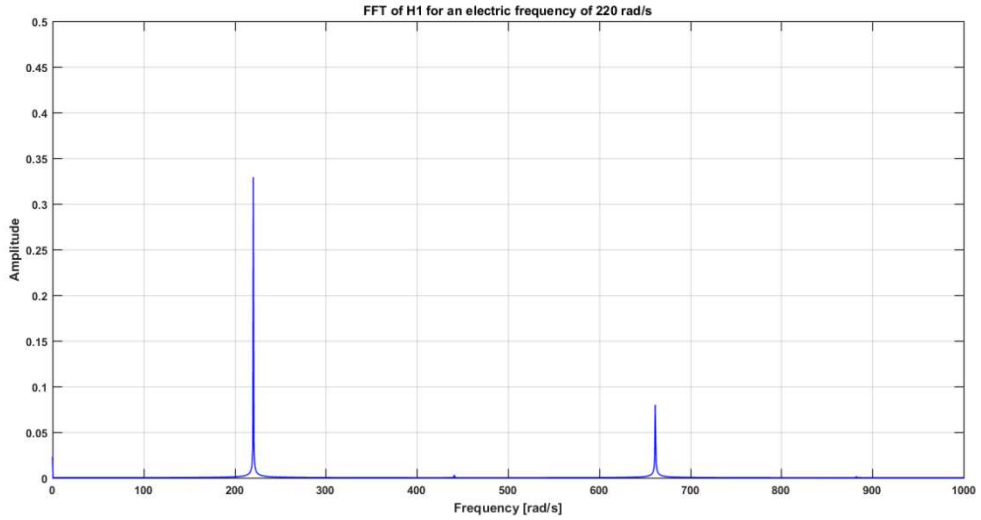

Figure 15 - FFT of the H1 Signal for an electric frequency of 220 rad/s

The designed ANF is based on two closed loops that use the estimated angular position to generate the proper secondary components, with both the correspondent harmonic and its coefficient according to the Fourier coefficients calculation bellow.

$$H_{\alpha 3} = \frac{2}{L}\int_0^L f(t) * \sin(3\omega t)\, dt \tag{13}$$

$$H_{\beta 3} = \frac{2}{L}\int_0^L f(t) * \cos(3\omega t)\, dt \tag{14}$$

Therefore, each signal is filtered out by the two loops. It is necessary the two closed loops for each input signal to complete the filtering, they are the sine and cosine third harmonic feedbacks that are multiplied by the computed coefficients.

Moreover, another component is introduced as a gain. This one is responsible to set up the sharpness of the notch filter and it is taken as the inverse of the Q-factor. The Figure *16* illustrates the different responses that can be achieved varying the σ gain.

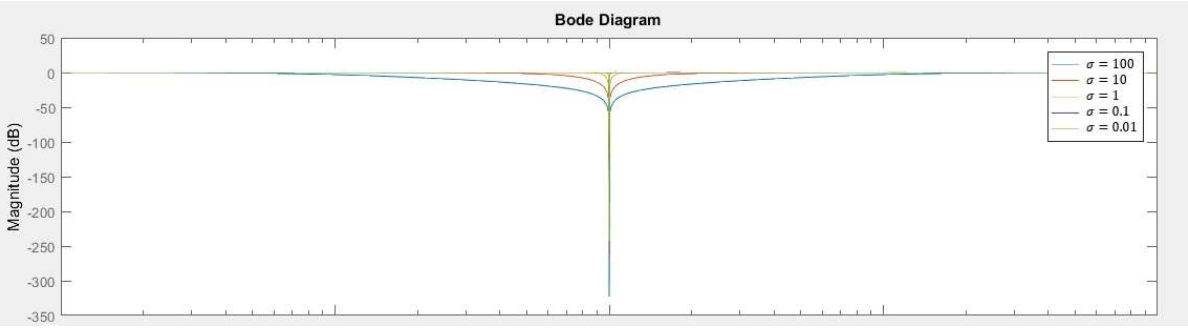

Figure 16 - Frequency response to different sharpness gains

Finally, the input signal is subtracted by the generated harmonics. An overall block diagram of the ANF is depicted on Figure *17*.

The closed loop transfer function of the ANF is calculated based on the two negative feedbacks.

$$\bar{H}(s) = H(s) - F_1(s) - F_2(s) \tag{15}$$

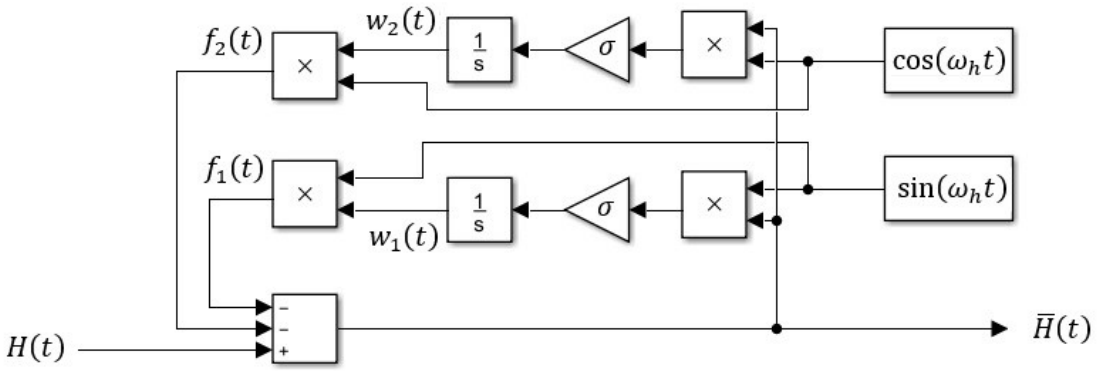

Figure 17 - ANF overall scheme

Where, on terms of Laplace transform, $H(s)$ is the not filtered signal, $\bar{H}(s)$ is the filtered one and $F_1(s)$ and $F_2(s)$ are the sine and cosine harmonic feedback. Considering $\mathcal{L}$ as the Laplace operator and ω_h the harmonic frequency, the transfer function can be calculated applying the Laplace transform on $w_1(t) = \widehat{H_{\alpha 3}}$ and $w_2(t) = \widehat{H_{\beta 3}}$ that are the estimated Fourier coefficients. For the cosine:

$$F_2(s) = \mathcal{L}[w_2(t) \times \cos(\omega_h t)] \tag{16}$$

$$F_2(s) = \mathcal{L}\left[\frac{1}{2} w_2(t) \times \left(e^{-j\omega_h t} + e^{j\omega_h t}\right)\right] \tag{17}$$

$$F_2(s) = \frac{1}{2}\{W_2(s + j\omega_h) + W_2(s - j\omega_h)\} \tag{18}$$

Hence,

$$F_2(s) = \frac{1}{2}\left[\frac{\sigma}{2(s + j\omega_h)}\{\bar{H}(s + 2j\omega_h) + \bar{H}(s)\} + \frac{\sigma}{2(s - j\omega_h)}\{\bar{H}(s) + \bar{H}(s - 2j\omega_h)\}\right] \tag{19}$$

Analogically, for the sine

$$F_1(s) = \frac{1}{2}\left[\frac{\sigma}{2(s + j\omega_h)}\{-\bar{H}(s + 2j\omega_h) + \bar{H}(s)\} + \frac{\sigma}{2(s - j\omega_h)}\{\bar{H}(s) - \bar{H}(s - 2j\omega_h)\}\right] \tag{20}$$

Thus,

$$\bar{H}(s) = H(s) - \frac{\sigma s}{s^2 + {\omega_h}^2}\bar{H}(s) \tag{21}$$

Therefore, the overall transfer function of the ANF is

$$\frac{\bar{H}(s)}{H(s)} = \frac{s^2 + {\omega_h}^2}{s^2 + \sigma s + {\omega_h}^2} \tag{22}$$

The Figure *18* illustrates a bode diagram of this ANF. As a regular second order notch filter, its output do not have any interference at frequencies different from ω_h, but it is attenuated at the frequency ω_h. However, the harmonic frequency here is variable and adapts itself to the input signal. Finally, as before mentioned, σ determines the sharpness of the filter and are the inverse of the Q-factor.

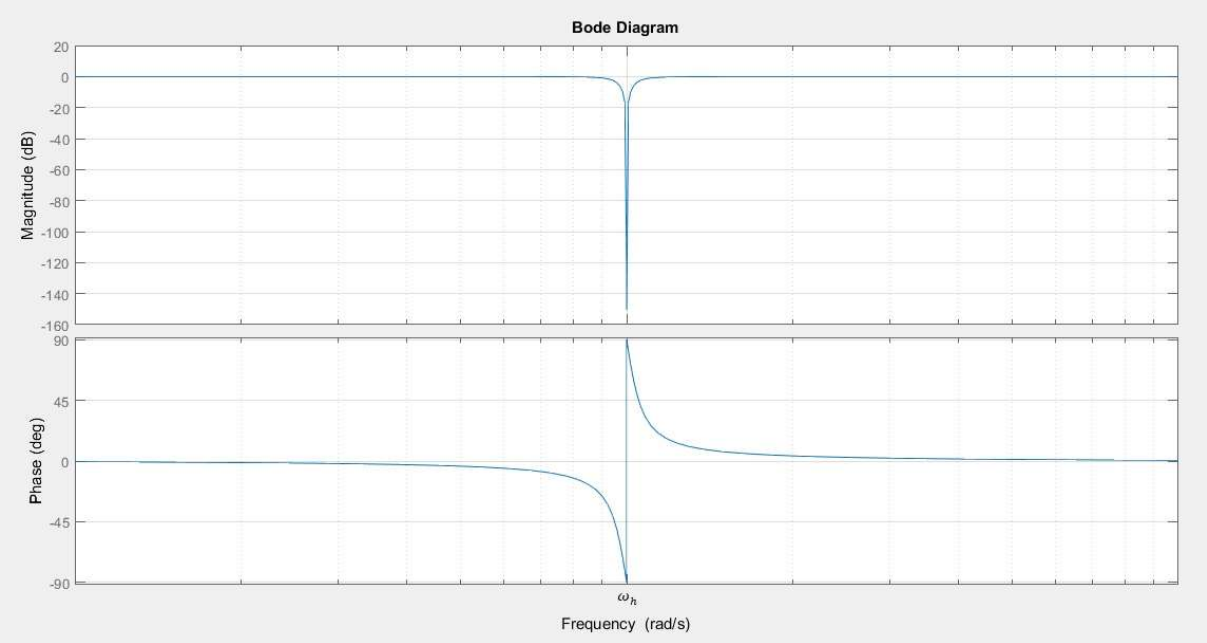

Figure 18 - Bode Diagram of the ANF transfer function

CHAPTER 6

PHASE LOCKED LOOP

6. PHASE LOCKED LOOP

A Phase Locked Loop is a control system that provides frequency and phase synchronization between an input signal and its output, such synchronization is required on the virtual resolvers to get a more precise angle and speed estimation. In fact, the PLL tracks the input signal frequency and generates sinusoidal waves with the frequencies as multiples of the input signal. Therefore, the PLL itself estimates the electric angular position and in some cases, it is able to provide the rotational speed of the motor.

The block diagram of a PLL system is depicted on the Figure *19*. Basically, it is composed of a Phase Detector, a Loop Filter and a Voltage Controlled Oscillator.

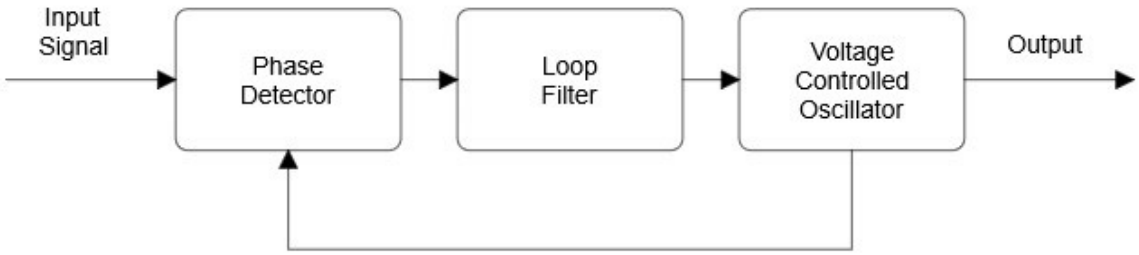

Figure 19 - Phase Locked Loop Block Diagram

The Phase Detector identifies the phase difference between the estimated and real signals.

A PI controller plays the role of the Loop Filter. It determines the dynamic behavior of the PLL system filtering out the high frequency ac components from the PD output $\varepsilon(t)$.

The Voltage Controlled Oscillator estimates the angular position and generates the required signal loop feedback.

In motor control applications, where the input frequency varies with the speed, an orthogonal PLL configuration is preferable instead a normal one that is useful to fixed frequencies and commonly applied on power grid systems. The orthogonal PLL process an orthogonal pair, such as sine and cosine signals and requires two feedbacks streams, thus, increasing its accuracy.

The Figure *20* shows the applied orthogonal PLL system that are discussed from now on.

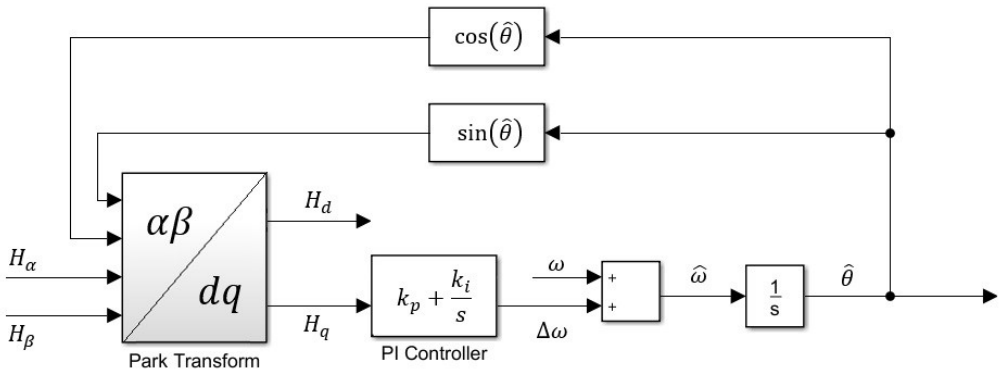

Figure 20 - Orthogonal PLL Block Diagram

The second step of the previously mentioned transformation from the three phase Hall signals to a dc system takes place now as the PLL phase detector. This first PLL component outputs an error signal proportional to the phase difference between the actual and the estimated angular position. Therefore, in order to do that, the Park transform is performed and utilizes the two feedback loops. Considering pure sinusoidal waveforms to the inputs:

$$\begin{bmatrix} H_d \\ H_q \\ H_o \end{bmatrix} = \begin{bmatrix} \cos(\hat{\omega}t) & \sin(\hat{\omega}t) & 0 \\ -\sin(\hat{\omega}t) & \cos(\hat{\omega}t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} H_\alpha \\ H_\beta \\ H_0 \end{bmatrix} \tag{23}$$

Thus,

$$\begin{bmatrix} H_d \\ H_q \\ H_o \end{bmatrix} = \begin{bmatrix} \cos(\hat{\omega}t) \times \cos(\omega t) + \sin(\hat{\omega}t) \times \sin(\omega t) \\ -\sin(\hat{\omega}t) \times \cos(\omega t) + \cos(\hat{\omega}t) \times \sin(\omega t) \\ 0 \end{bmatrix} H \tag{24}$$

$$H_d = H\cos((\omega - \hat{\omega})t) \tag{25}$$

$$H_q = H\sin((\omega - \hat{\omega})t) \tag{26}$$

Changing the notation of Equations *25* and *26.*

$$H_d(t) = H\cos\left(\theta - \hat{\theta}\right) \tag{27}$$

$$H_q(t) = H\sin\left(\theta - \hat{\theta}\right) \tag{28}$$

Where,

$$\theta = \int \omega\, dt = \int (\omega_n + \Delta\omega)\, dt \tag{29}$$

$$\hat{\theta} = \int \hat{\omega}\, dt = \int (\omega_n + \Delta\hat{\omega})\, dt \tag{30}$$

Hence,

$$H_d(t) = H\cos\left(\Delta\theta - \Delta\hat{\theta}\right) \approx H \tag{31}$$

$$H_q(t) = H\sin\left(\Delta\theta - \Delta\hat{\theta}\right) \approx H(\Delta\theta - \Delta\hat{\theta}) \tag{32}$$

Assuming steady-state condition and so, for small values of θ, $\sin(\theta) \approx \theta$, the phase difference is found. Moreover, the transformation into the dc system is complete now and the d and q components are available. Indeed, the q component is the phase difference error $\varepsilon(t)$ and it is used as the Loop Filter input and the d component is a measure of the input amplitude.

Considering the Equations *29 to 32* and the PLL block diagram on Figure *20,* a linearized model of the PLL is obtained as shown on Figure *21*.

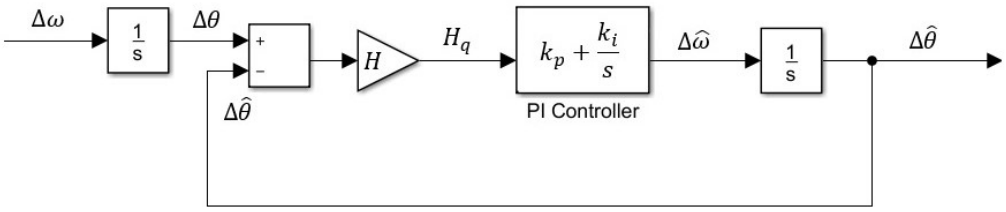

Figure 21 - Linearized PLL model

From that, the input signal of the VCO block is the estimated frequency. Considering $H = 1$, the closed-loop transfer function relating $\Delta\hat{\theta}$ to $\Delta\theta$ is:

$$TS(s) = \frac{\Delta\hat{\theta}(s)}{\Delta\theta(s)} = \frac{k_p s + k_i}{s^2 + k_p s + k_i} \tag{33}$$

This PLL system has two open-loop poles at the origin, that means a zero steady-state phase error when tracking phase-angle jumps and frequency steps. However, in the case of frequency ramps, the steady state phase error is still present. Considering C_a the slope of the frequency ramp, for the linearized model the error can be calculated as:

$$e_{ss}^{ramp} = \lim_{s\to 0} s \frac{1}{1 + G(s)} \times \frac{C_a}{s^3} \tag{34}$$

Where, $G(s)$ is the open-loop transfer function. Hence,

$$e_{ss}^{ramp} = \lim_{s\to 0} s \frac{1}{1 + \frac{Hk_p s + Hk_i}{s^2}} \times \frac{C_a}{s^3} = \frac{C_a}{Hk_i} \tag{35}$$

Considering the nonlinearity,

$$e_{ss}^{ramp} = \sin^{-1}\left(\frac{C_a}{Hk_i}\right) \tag{36}$$

The PLL bandwidth is increased with the k_i gain, which reduces the steady-state ramp error. However, a trade off must be found since enlarging the bandwidth increases the PLL susceptibility to noise.

On regards the choice of the values for the proportional and the integral gain, the Equation *33* can be written as:

$$TS(s) = \frac{\Delta\hat{\theta}(s)}{\Delta\theta(s)} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{37}$$

With:

$$\omega_n = \sqrt{k_i}\ ;\ \xi = \frac{k_p}{2\sqrt{k_i}} \tag{38}$$

Thus, the gains can be tuned as function of the damping and of the settling time.

$$k_p = \frac{-2\ln(tol * \sqrt{1-\xi^2})}{T_s} \; ; \; k_i = \left(\frac{-\ln(tol * \sqrt{1-\xi^2})}{T_s \xi}\right)^2 \tag{39}$$

Selecting $\xi = 0.7$, $T_s = 30\,ms$ and the tolerance of 5%, the Loop gains are: $k_p = 222.16$ and $k_i = 25181$.

Another important consideration can be done on regards the harmonics that are generated by the sines and cosines multiplication on normal PLL's. Once again, the orthogonal PLL proves to be more efficient them the normal PLL, since this 2nd order harmonic is not present on its q component. Nevertheless, the effects of higher order harmonics of the introduced non-filtered signals are still present even after the Clarke and Park transformations.

Therefore, the Adaptive Notch Filter takes a significant role on PLL systems, filtering out the harmonics, reducing its noise disturbance to enlarge the PLL bandwidth. The PLL that makes use of an ANF is called EPPL, *Enhanced Phase Locked Loop.*

Finally, a brief discussion on speed estimation will be done since the PLL system requires a frequency input ω. As said before, such frequency is not fixed as the grid applications, it varies with the motor rotational speed.

6.1. Speed Estimation

The speed estimation algorithm depends on the type of sensor that is applied and how the induced voltage on the sensor is interpreted by the control. Let's considerate the already mentioned Hall sensors and its available signals on the Kontrol board. Thus, some solutions will be presented.

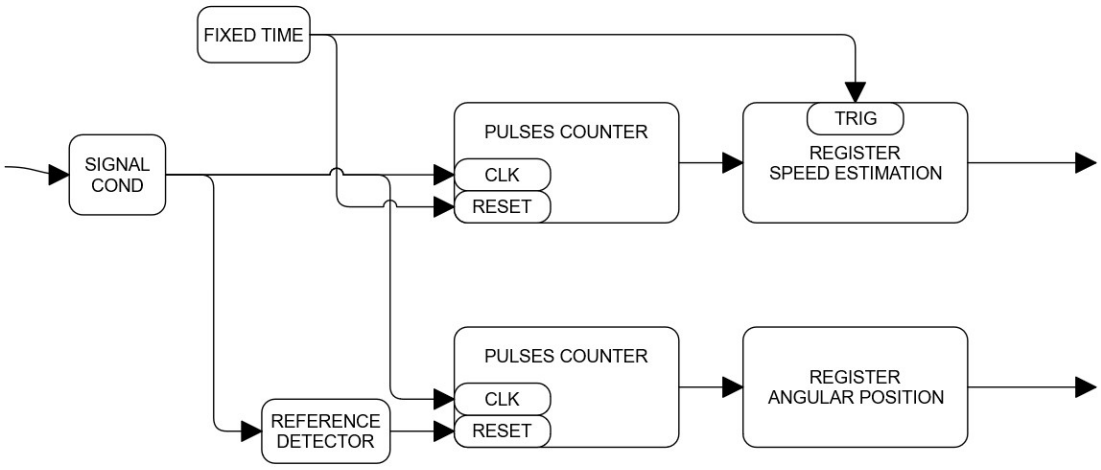

Figure 22 - Fixed Time-based Speed Estimation Algorithm

After the signal conditioning, a first algorithm to the estimation of the speed can be shown on Figure *22*. This one is based on counting the number of pulses in a fixed time interval. These pulses can be identified as the positive region of the sinusoidal waveform by zero crossing elements. Hence, the angular displacement between two consecutives pulses is evaluated as:

$$\alpha = \frac{4\pi}{N} \tag{40}$$

Where, N is the number of magnets in one total revolution. Thus, the estimated angular speed on a time interval T is calculated on base of the counted pulses n on that interval.

$$\omega = \frac{4\pi\, n}{T\, N} \tag{41}$$

It is worth noting that since the time interval is fixed, the speed estimation is subject to some oscillation or variation error due to the not precise sampling of the pulses. For example, an incomplete pulse can or cannot be counted as a complete one and, since the counter does not allow decimation, it is not possible consider a pulse partially.

Hence, another algorithm can be presented as a solution to the oscillation problem. It is possible estimating the speed fixing the angular displacement to that one between two consecutives zero crossings of the input signal, evaluated as:

$$\alpha = \frac{4\pi}{N} \tag{42}$$

Thus, the correspondent time interval is used to calculate the average speed:

$$\omega = \frac{4\pi}{(t_2 - t_1)\, N} \tag{43}$$

Where t_1 corresponds to the first zero-crossing and t_2 to the second one. This variable-time based algorithm is illustrated on Figure *23*.

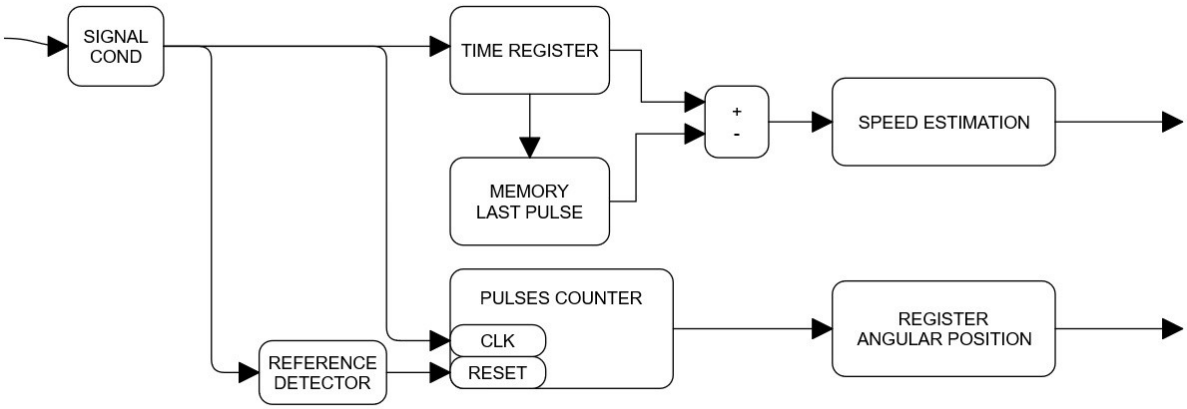

Figure 23 - Variable-time based speed estimation algorithm

Taking into account these algorithms utilize only one of the Hall sensors signals, they cannot provide the rotational direction of the motor and, although the good speed estimation, at low speeds some quantizing error are present since computing the speed depends on the time interval between two zero-crossings that increases in such situation, mostly when starts from zero speed.

An algorithm to calculate the direction of rotation is shown below and makes uses of all

the three-phase signals.

Firstly, it is necessary to considers the square waves generated with the compare to zero logic. Follow that:

```
%% Rotational Direction Detection
%% A, B and C are the squarewave three-phase signals
while(1)
    if (detect changes (A))
        if (A>0)
            direction=!B;
        else
            direction=B;
        end
    end
    if (detect changes (B))
        if (B>0)
            direction=!C;
        else
            direction=C;
        end
    end
    if (detect changes (C))
        if (C>0)
            direction=!A;
        else
            direction=A;
        end
    end
end
```

The speed estimation can be improved using the three-phase signals only after the harmonic filtering, over again an issue when starts from zero speed.

Another option would be the differentiation of the $\arctan(x_\beta/x_\alpha)$ signal that can provide a satisfactory speed estimation even at low speeds. However, such technique is hard to employ on discrete digital processing systems.

Lastly, considering high values on the integral gain of the PI controller, it is possible do not use the frequency input on the PLL since the gain, if high enough, takes care of the dc offset. In this case, to general outside use, the PLL itself provides the angular speed estimation.

The modelling and simulation of the proposed virtual resolver will be presented on the next sections aiming the implementation on the *Electronic Kontrol Unit* of the *Risciò*.

CHAPTER 7

MODELLING IMPLEMENTATION

7. MODELLING IMPLEMENTATION

The previous considerations give rise to the next step on the development of the Virtual Resolver, modelling and simulations are done in the software flow step. Thus, the present and next chapters introduce the modelling implementation and simulations of the previously explained systems on the *Matlab-Simulink* environment.

In order to simulate and analyses the behavior of the proposed systems, the model, from now on called Virtual Resolver Model, is entered on a behavioral simulation environment. Such environment simulates the inputs of the overall virtual resolver system and it's able to provide satisfactory comparison between its inputs and outputs. Therefore, this environment includes the Hall sensor board model, scaling systems, measurement of errors and graph and vector outputs.

The components of the Virtual Resolver Model are specifically: the pre-processing system, the ANF and the Virtual Resolver itself. The latter is composed by the Clark and Park transformations, the arctangent algorithm, the speed estimators and the PLL system.

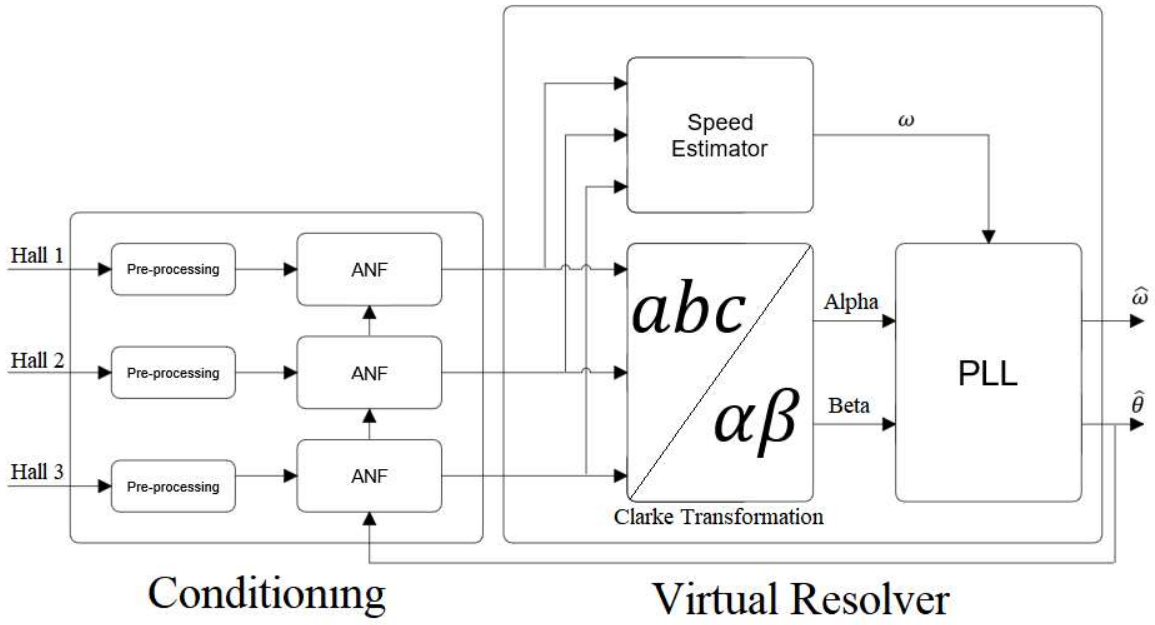

Figure 24 - Virtual Resolver Model Block Diagram

It is important to mention that each system was modeled separately, which aid not only on the design process but also on the preparation of the automatic code generation. Thence, each system gives rise to a specific subroutine to be implemented jointly or separately on the *EKU*. Accordingly, the development of each system will be brought forward separately on next, including the Hall sensor model, and the simulation accessories.

On the other hand, the simulation of the entire system is a key step of the project. On the software flow, besides the identification of bugs and errors in the modelling, the simulations are also important on tuning the models in order to generate a better improved code. Thus, the simulations were performed taking account speed inputs with physical constraints and a real urban cycle as references.

Also, auxiliary .m files are present to introduce the specific parameters of the different

systems on the model and to generate the comparison graphs, vectors and images. Such files are provided on the appendix of this work.

Lastly, it is noteworthy that, for now, a continuous time approach was taken. However, since the *EKU* board is only able to work with discrete data, it is necessary to perform the model discretization, that will be discussed later.

7.1. Hall sensors model

Firstly, it is important to mention that this system is not part of the Virtual Resolver model itself. However, it is a key element on the entire simulation environment. The hall sensors model simulates the three-phase waveform generation of the real Hall sensor board that will be used as the input of the resolver.

The Hall sensor board provides a three-phase analog signal. Such signal, as seen before, is not composed by perfect sinusoidal waveforms. Thus, each one of the three Hall signals is a sum of the principal component and the harmonic components. On the model, they are generated using a math function, as sine and cosine of a specific angle.

The Hall sensor block is based on a frequency or speed input. Such speed is related with the vehicle speed and consequentially with the rotational frequency of the motor-in-wheel. Hence, considering the number of pair poles of the PMSM as *N/2*, the electric frequency of the three-phase signals is *N/2* times the wheel speed.

Finally, the angular position used by the math functions is the integration of the electric frequency. The higher order harmonic amplitudes and the pair poles number are introduced by means of the mat file *Resolver_Parameters_2.m*. The *mod* function is used to scaling Theta between 0 and 2π.

The Figure *25* shows the block diagram of the entire Hall sensors model and its details.

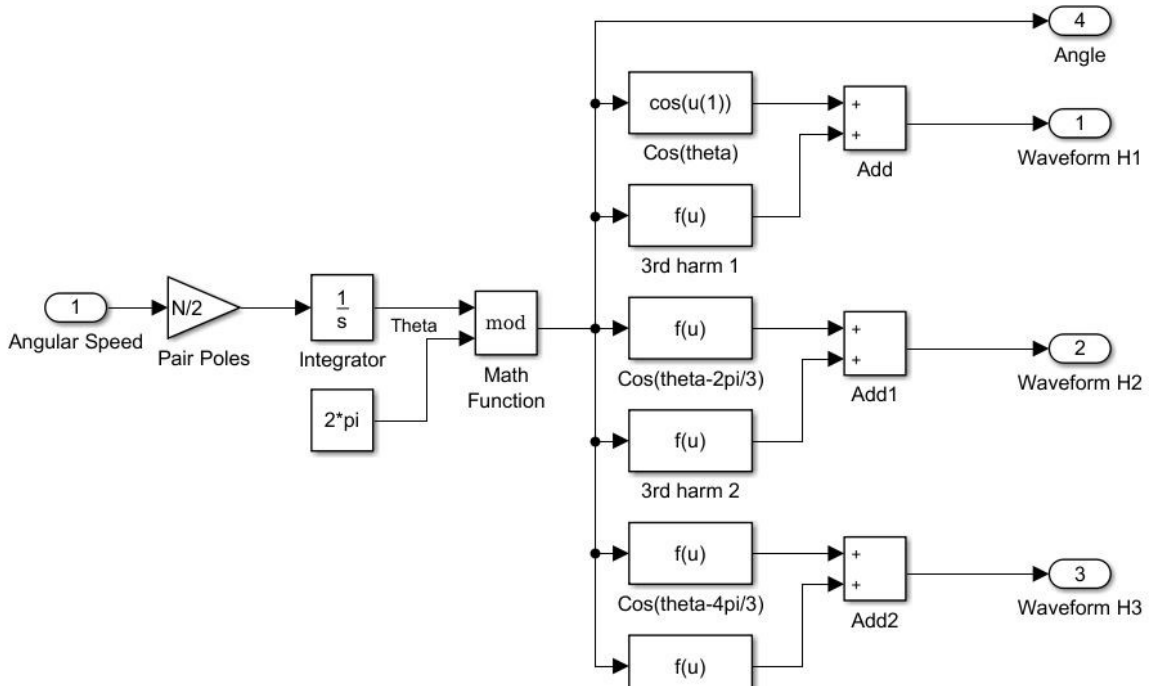

Figure 25 - Hall Sensors Model block diagram

7.2. Acquisition System model

The pre-processing system is the front line from the Virtual Resolver point of view. Therefore, the system input is the ADC conversion value. As was said before, proper shift and scaling must be done. Considering the limit values of the input signals an $Offset_{FC}$ is stablished as well as the $Gain_{FC}$. The index FC indicates the *First Conditioning* system and for each signal it's necessary a separated conditioning. Lastly, the low pass filter is introduced as a transfer function system, its numerator and denominator vectors are described in the mat file *Resolver_Parameters_2.m.*

The First Conditioning system is depicted on Figure *26.* The highlighted *H2* output signal is called *H2_cond,* the second Hall signal after the first conditioning.

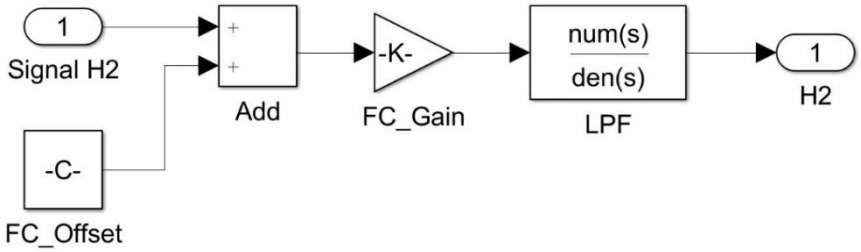

Figure 26 - Acquisition; First Conditioning

It's possible to mention also that a flag is necessary to indicate zero velocity. After one of the conditioned signals a comparator is used to determine if the signal is changing (not zero) or is static (zero speed).

7.2.1. Adaptive Notch Filter

An enabled system is utilized for the ANF implementation, to be active only when the constant input flag indicates the speed different from zero. As input we have the three conditioned signals and the Virtual Resolver estimated angular position. As output, the three Hall Signals after the ANF, *Hi_anf.*

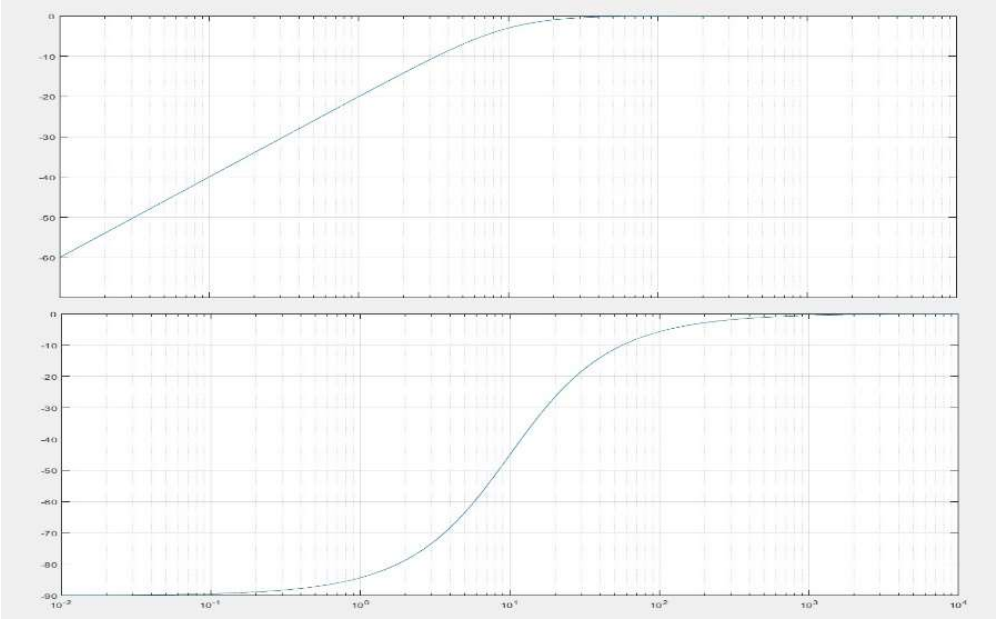

Figure 27 - Bode diagram at specific frequencies of the ANF

Considering the ANF transfer function, a problem comes to light on regards the low frequencies region. The Figure *27* shows the bode diagram of the system considering the specific fundamental frequencies. Since the filter is adaptive, the magnitude and phase of each frequency of interest was taken to compose the below plots.

It is notice a changing on phase and magnitude, the ANF acts as a first order high pass filter at the frequencies of interest, that is the fundamental frequencies of the sinusoidal waves, with 10 rad/s as the cutoff frequency. Therefore, a controller must be designed to overcome this drawback at low frequencies and its transfer function must be introduced at the end of the filter. More details about the design of such controller will be discussed further, on the simulation section.

Finally, the overall ANF block also provide at its output a nonfiltered signal port. The selection between the filtered and the nonfiltered signals takes place outside the block with another flag.

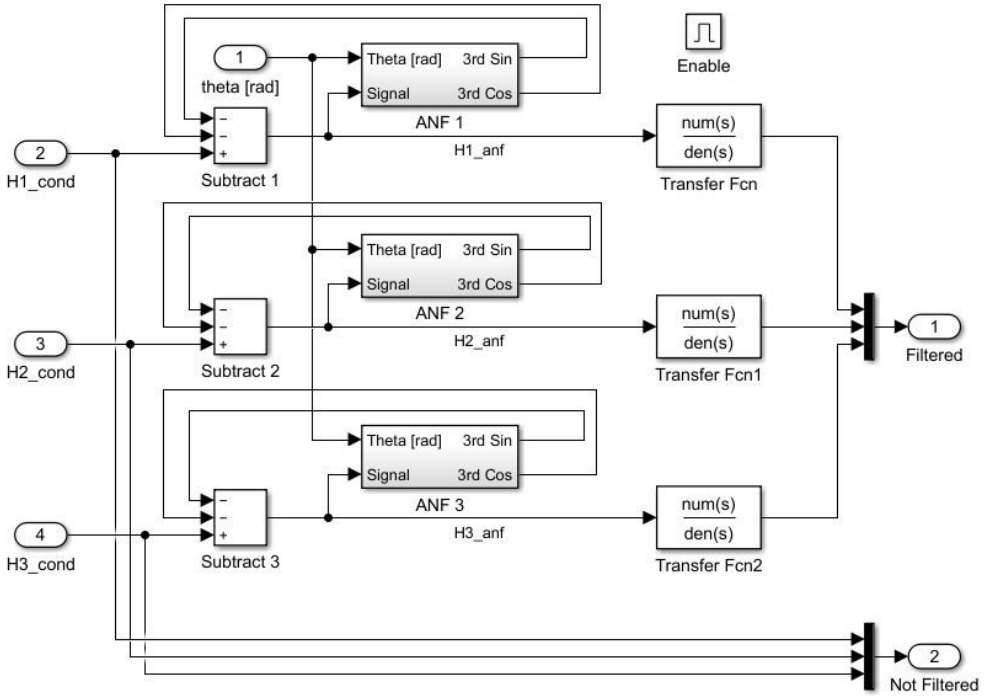

Figure 28 - ANF overall model

The overall ANF model is shown in Figure *28.* The Figure *29* shows the details of the ANF system.

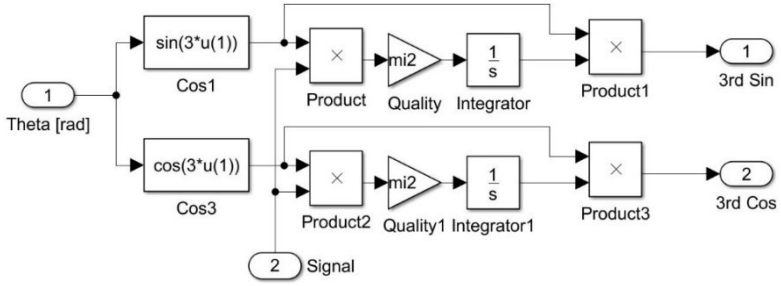

Figure 29 - Detailed ANF

7.3. Clarke Transformation

The transformation matrix shown on Equation *4* is implemented on this block. Hence, the front-end of the Virtual Resolver itself, the Clarke Transformation is simply depicted on Figure *30* as a Three-Phase to Alpha-Beta waveform.

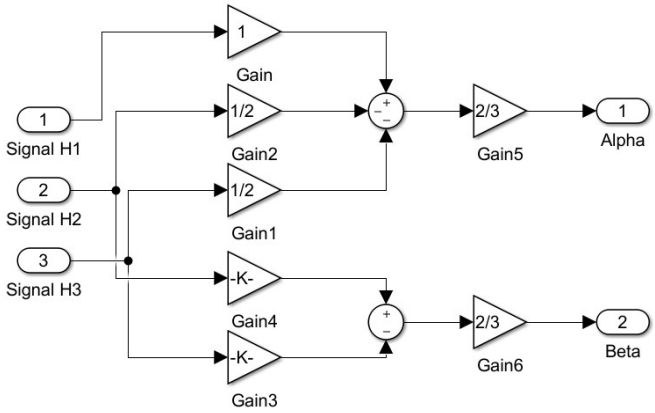

Figure 30 - Clarke Transformation

7.4. Arctangent Logic

The arctangent logic depicted on Equation *2* is now implemented. The estimation of the angular position is done using the transformed Sine and Cosine as inputs.

The Figure *31* shows the implemented logic, each "atan" block uses a different calculation considering the four-quadrant logic explained before.

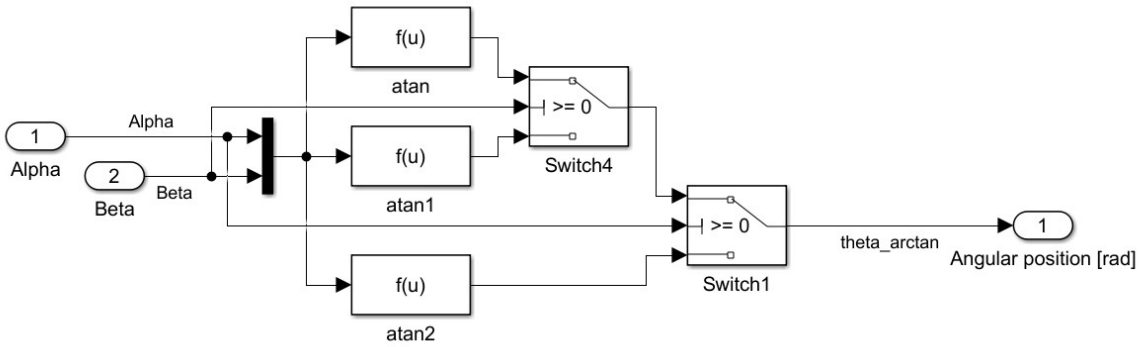

Figure 31 - Arctangent Logic

The system can be improved on execution with a look-up table, discussed further on the discretization section.

One last consideration is that this estimated angular position is slightly better than the one provided by the PLL. However, it cannot be used to close the loop with the ANF since the signal becomes increasingly distorted.

7.5. Speed Estimator

Considering the use of small poles on the PLL, the speed estimator plays a significant role since the integrative gain cannot lead with the dc offset by itself. Therefore, a frequency input is necessary to proper angular position estimation.

7.5.1. Variable-time based algorithm

The variable-time estimator algorithm presented before on this work is introduced on the system.

Using only one of the three-phase signals, after the pulse identification by a hit cross both rising and falling detector, a triggered subsystem, in each crossing, is responsible for calculate the instantaneous period of the signal by means a memory of the last pulse time.

The speed is then computed by the inverse of the period and a gain. Taking into account that each pulse is generated on half of the period., the gain value is $\frac{2\pi}{2}$ to convert the instantaneous angular speed to radians per second.

It is a good practice to introduce a low pass filter to filter out the oscillatory error of the triggered subsystem. A third-order filter is considered, and the Butterworth approximation was chosen with the cut-off frequency of 100 rad/s, it gives the follow transfer function:

$$LP(s) = \frac{10^6}{s^3 + 200s^2 + 2 * 10^4 s + 10^6} \tag{44}$$

Finally, it is notice that a non-zero error takes place every time the triggered subsystem waits a second pulse indefinitely, for example when decreasing the speed and very close to zero the signal maintains constant do not generating a hit cross. Then, a switch block is used to force the speed to zero when the constant input flag is activated.

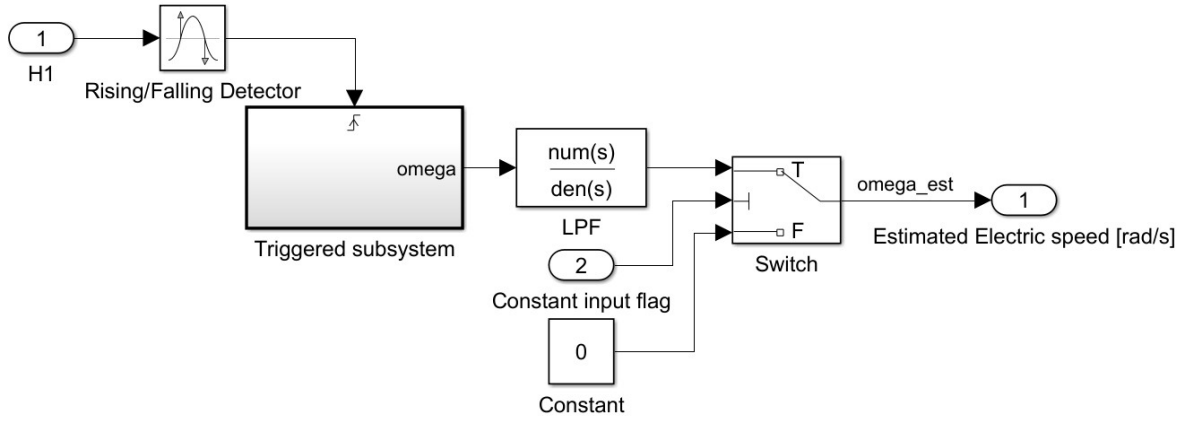

Figure 32 - Speed Estimator

The overall block diagram of this speed estimator is depicted on Figure *32,* with the triggered subsystem detailed on the Figure *33.*

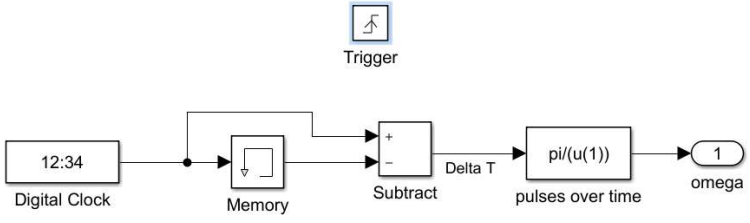

Figure 33 - Triggered Subsystem

The previously scheme can be modified to accomplish the rotational direction problem.

Hence, another triggered subsystem is used with the specific logic responsible for such calculation before explained. The proposed logic makes use of all the three phases, the trigger is the hit crossings of all the three signals working as an interruption branch and it is shown on the block diagram of the Figure *34.* The values 1 and -1 indicates clockwise and counter-clockwise respectively.

One last consideration is about to increasing the estimation resolution with all the three phases signal. Due to the irregularity of the signals and the shape difference between them, the use of the three signals to estimate the velocity cannot guarantees an accurate performance, presenting high oscillatory error. Such oscillatory error must be filtered out and the performance becomes practically the same with more effort.

The use of the properly filtered out signals eliminates the irregularity shape and can improve the measurement. However, the start-up region presents some issues that will be discussed further on the simulation section.

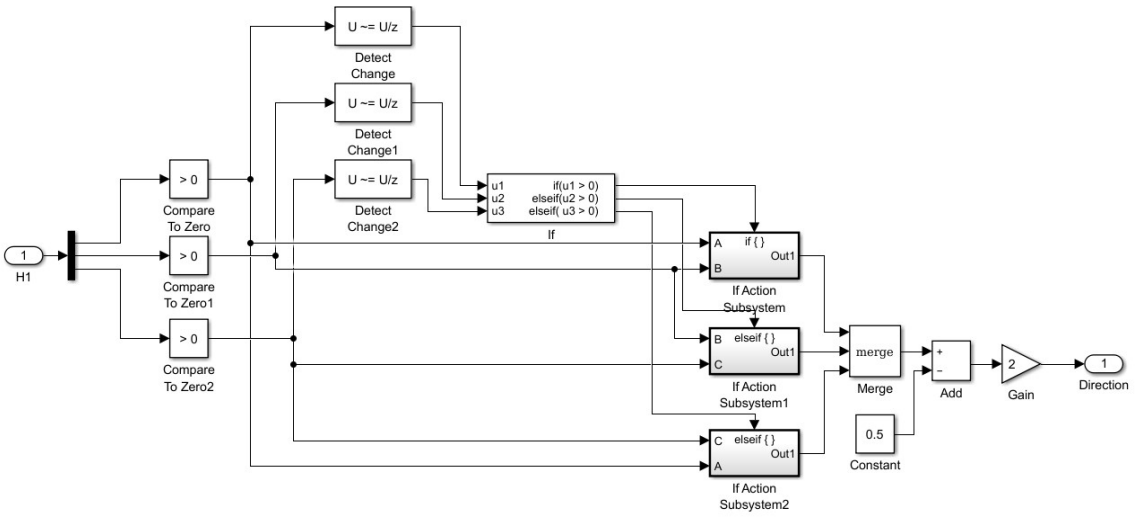

Figure 34 - Direction of Rotation Logic

7.5.2. Arctangent differentiation

Another possible solution, as explained before, would be to use the differentiation of the $\arctan(x_\beta/x_\alpha)$ signal that can provide a satisfactory speed estimation even at low speeds.

However, at least two problems come up. The first one is on regards the closed loop system, it is not possible to use the filtered signals to compute the arctangent and consequently, to estimate the speed on closed loop with the PLL-ANF system because the signal distortion increases as well on the arctangent logic.

Therefore, the non-filtered three-phase signals are introduced on the Virtual Resolver as a second branch that remains opened. Then, the Clarke transform takes place and the arctangent logic previously explained. Finally, the differentiation is performed, and this is where the second problem occurs.

The differentiation process is hardly achieved on digital signal processing. Thus, on an attentive to simplify the process, a simple algorithm is developed. A triggered subsystem was used with a fixed time trigger T. Taking into account that the arctangent logic angular position output goes from 0 to 2π, the follow algorithm was developed.

```
%% Angular displacement along the fixed time 2*T (from theta_n2 to theta_n)

if ((theta_n>theta_n1)  | (theta_n1>theta_n2)) & (theta_n<=theta_n2)
    y_n=theta_n-theta_n2-2*pi
else
    y_n=theta_n-theta_n2
end
```

Where, theta_n1 and theta_n2 are respectively the last and the last but one values of the angular position.

Hence, the angular speed is calculated as:

$$\omega_{atan} = \frac{y_n}{2T} \tag{45}$$

The Figure *35* illustrates the arctangent differentiation block diagram.

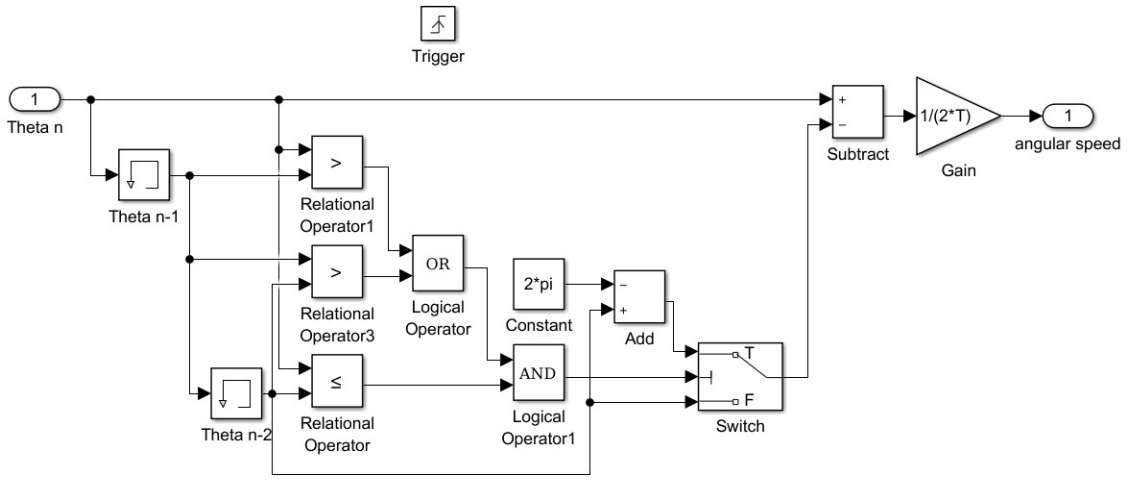

Figure 35 - Arctangent differentiation

Lastly, a low pass filter is used in order to filter out the oscillations on the signal due to the presence of harmonics on the non-filtered signals.

$$LP_{atan}(s) = \frac{1000}{s^3 + 20s^2 + 200s + 1000} \tag{46}$$

Another approach can be taken for the arctangent differentiation based on trigonometric transformations. Since it takes into account discrete signals, it will be explained further on this work.

7.6. Phase Locked Loop

The PLL model is depicted on Figure *36.* Basically, it follows the previous explained concept. The values of the integrative and proportional gain are introduced by the mat file *Resolver_Parameters_2.m.* Finally, the *mod* function is used again to scaling the angular position between 0 and 2π.

The Park Transform is detailed on Figure *37*.

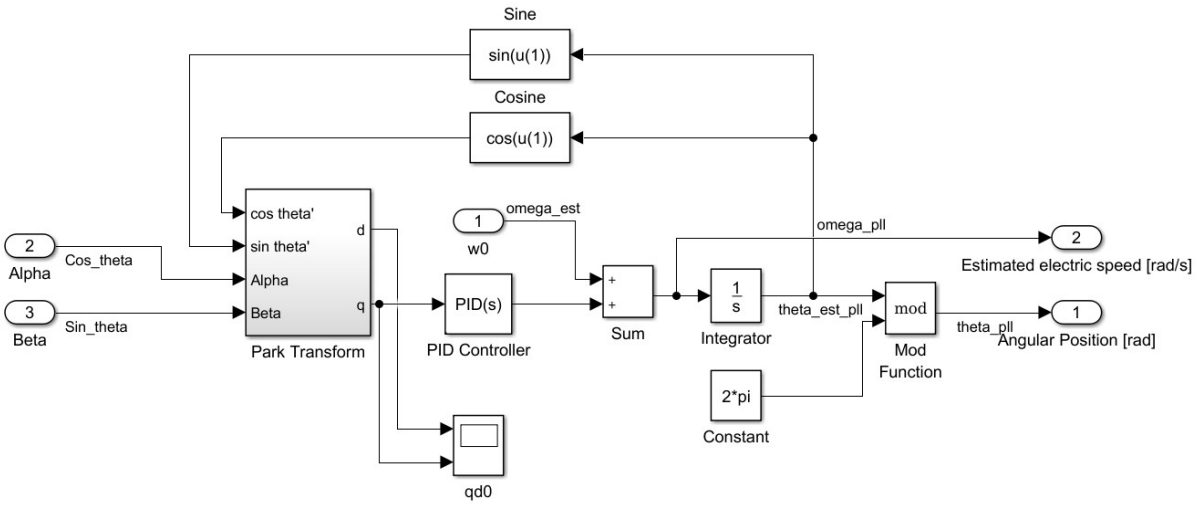

Figure 36 - Phase Locked Loop model

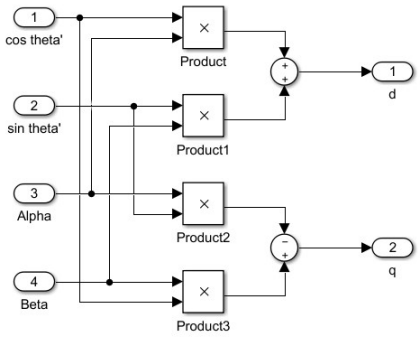

Figure 37 - Park Transformation implementation

7.7. Overall Virtual Resolver Model

All the proposed component models discussed and explained before were put together to compose the Virtual Resolver overall block diagram that is illustrated on Figure *38*.

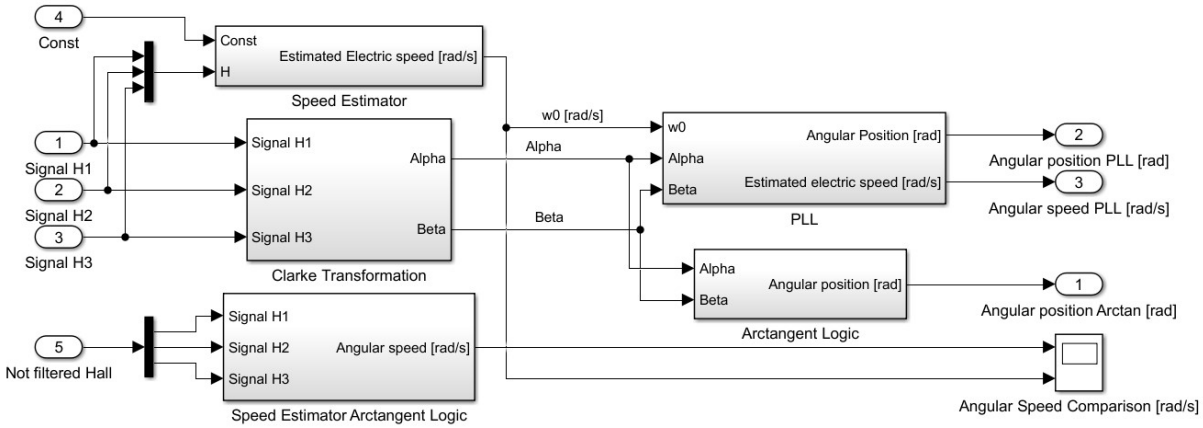

Figure 38 - Virtual Resolver overall model

The two proposed speed estimators are put together in order to evaluate the behaviour and to compare which one presents the better performance on regards the complete loop environment.

The performance evaluation must considerate not only the errors between the generated signals, that is the reference angular speed and position, and the estimated ones, but even the execution time is a crucial factor on the real-time applications.

CHAPTER 8

SIMULATIONS

8. SIMULATIONS

The behavioral simulation environment that was mentioned before allows the execution of several and different tests. Hence, it is possible to change the inputs, the components of the loop and even the specific parameters of each component and evaluate the behaviour of each system separately and/or together.

A series of tests and simulations are performed with different speed inputs (constant, ramp, urban cycle, reverse) and assessing the different components of the entire system.

The firsts tests have the aim of characterize the effects of the ANF and PLL components and verify if the proposed solution will improve the desirable measurements comparing to the purely arctangent logic commonly applied on the physical resolvers.

The subsequent simulations will take account the drawback effects of each component to design, as mentioned previously, a kind of controller to overcome these effects and improve the estimations.

A third series of simulations is done evaluating the performance of the speed estimators and its behaviours at critical work condition regions. Once again with the purpose of improving their operation and choosing the suitable one to the application.

Finally, a considerable set of simulations is executed aiming the improvement of the specific parameters and the overall system. This virtual tuning process is also responsible for a first preparation of the system to the automatic code generation, eliminating unnecessary process and adding crucial ones.

Due to the several simulations and results, this section will only present the principal achievements and results of the above proposed objectives. Nonetheless, this work appendix exhibits some other graphs and comments about the simulations and tests, besides the auxiliary files that were used.

The Figure *39* shows the complete virtual environment that was used in all the performed simulations.

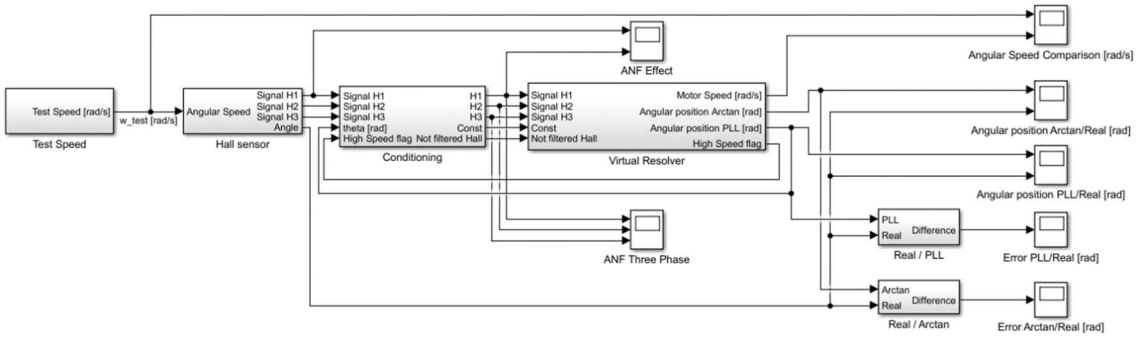

Figure 39 - Behavioral Simulation Environment

8.1. Defining the system inputs

Since the required outputs are the electric angular speed and position of the motor, a speed input comes to match also with the Hall sensors model, making an accessible reference to the measurement process.

The virtual environment allows a wide variety of input signals; however, it is necessary considerate some physical constraints, both for practical situations on the real system and to test the limits on a real urban cycle, considering the vehicle application. Thus, three principal different speed inputs were introduced aiming the complete evaluation of the system.

The first one, a constant input is used to analyse the stead state behaviour of the Virtual Resolver. Although instantaneous changes on speed are not possible on real motors, this study will analyse the behaviour of the system far away from the start-up region.

As the second input, positive and negative ramps are used to study the angle estimation tracking and speed error. Their slopes will depend on the simulation, nonetheless, maximum acceleration and speed are considered, generating a bounded signal. The maximum acceleration can be calculated using the 0 to 100 km/h timing, about 2.8 seconds in electric vehicles, hence, considering the wheel radius, its maximum acceleration is about to 85 rad/s². While the maximum wheel speed is taken as 120 rad/s. Finally, negative values for the speed are also taken into account.

In last case, considering the application, it is worthy to analyse the system from the vehicle point of view. Hence, a well-known urban cycle is introduced as a third type of speed input, the NEDC.

The above-mentioned speed inputs will be discriminated on the respective simulations, considering its values and parameters, when needed. They will be transformed into a reference angle and then, into the three-phase Hall sensor signals. Thus, the inputs for the Virtual Resolver model before the conditioning system and ANF are depicted in Figure *40* as well as the reference electric angular position for a constant speed input of 30 rad/s.

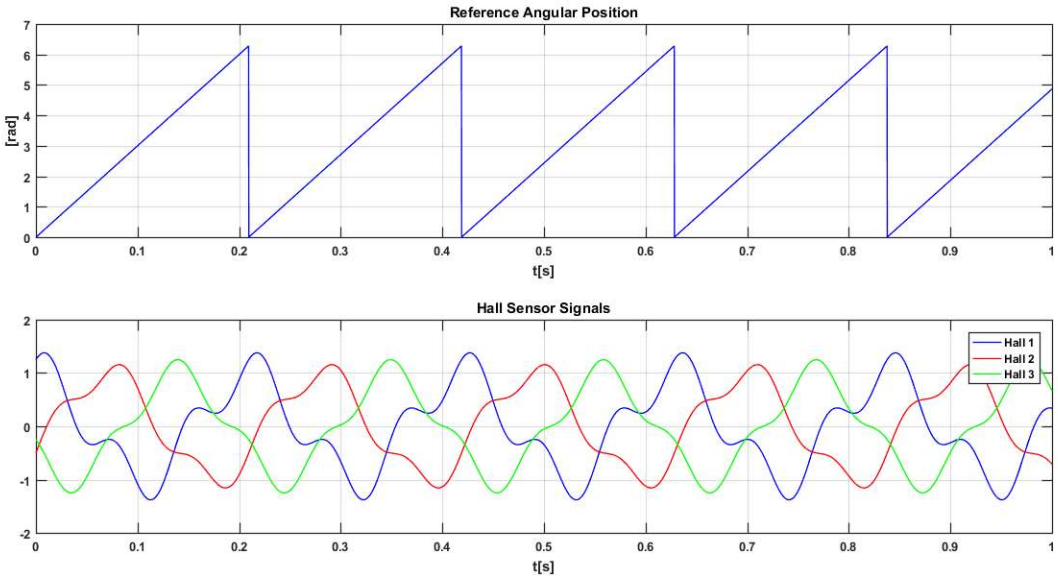

Figure 40 - Reference Angular Position and Hall Sensor Signals

8.2. ANF and PLL Characterization

As means of comparison, a first simulation was performed without the use of the proposed ANF and PLL systems. The position estimation is made based only on the arctangent idea, considering only the Clarke Transformation from three-phase to Alpha-Beta signals as shown in Figure *41*.

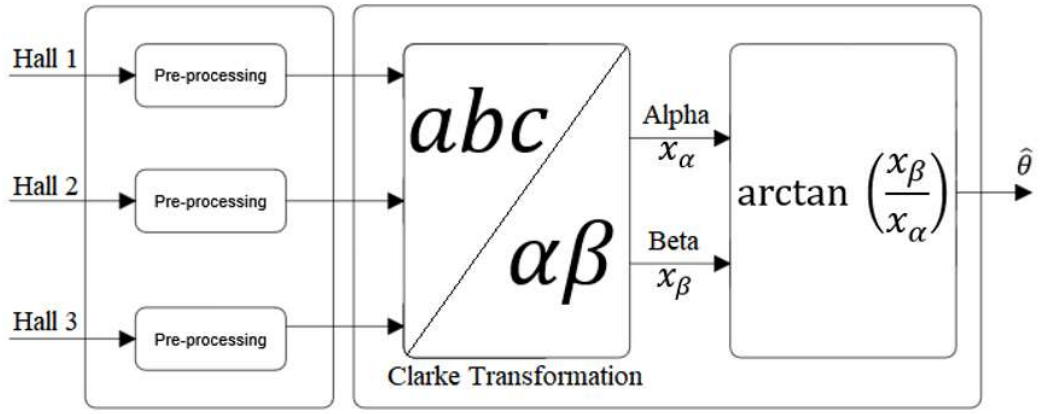

Figure 41 - Arctangent estimation scheme

The output comparison and the position estimation error are shown in the Figures *42*.

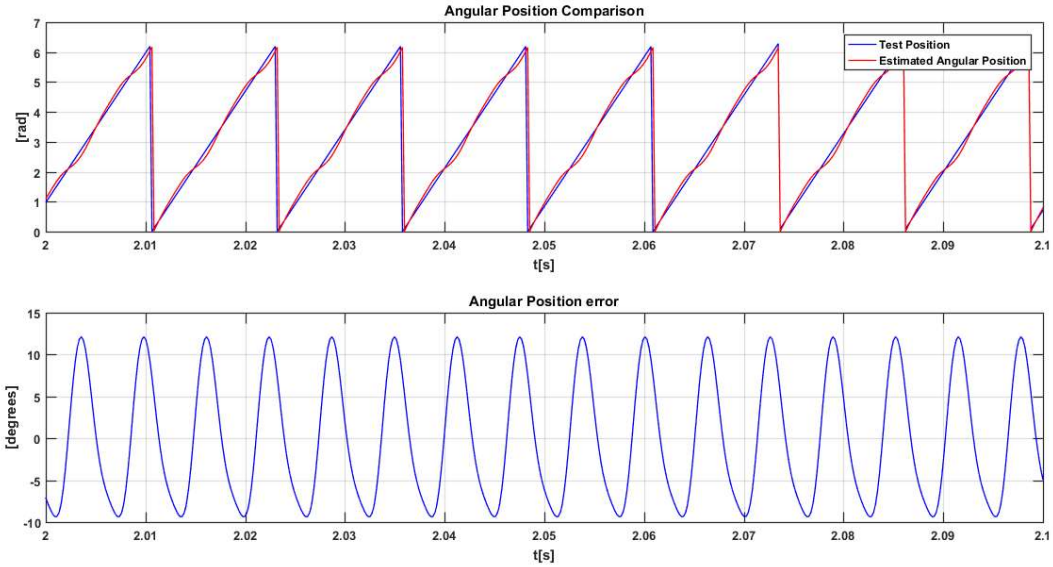

Figure 42 – Arctangent Angular Position Estimation Comparison and Error (without ANF/PLL)

On regards the constant input of 500 rad/s electric speed, the purely arctangent estimation shows, as expected, a high oscillatory error because of the presence of harmonics on the Hall sensors Signals.

In order to characterize the PLL and the ANF effects on such system, the same test was applied to the PLL angular position estimation, firstly, without the ANF. Then, the ANF is activated after 5 seconds. The Figure *43* illustrates the results of this test.

It is notice a decreasing on the error oscillatory amplitude just with the PLL system. However, the use of the ANF increases the PLL accuracy, filtering out the high order harmonics. Moreover, a certain linearity on the estimated angular position is better observed after the ANF activation, although an offset error is also present as an effect of the ANF filtering.

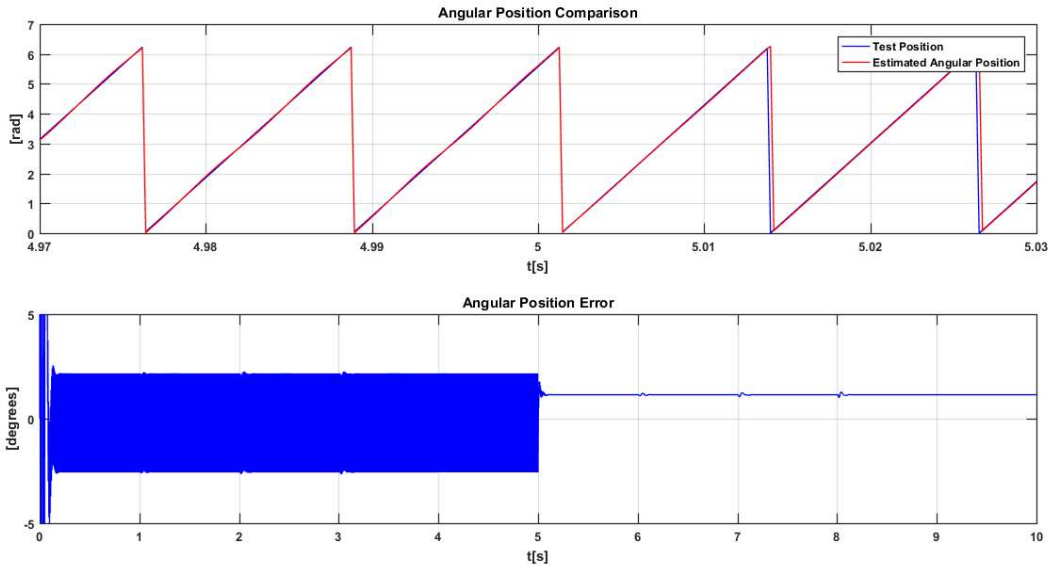

Figure 43 - PLL Angular Position Estimation Comparison and Error (with ANF activation)

Another look can be taken on regards the Hall sensors signals before and after the ANF conditioning. As proposed in the previous sections, the ANF aims filtering out the third order harmonics on the Hall three-phase signals, this is clearly depicted on Figure *44,* after the activation point, the signal acquires the fundamental sinusoidal shape.

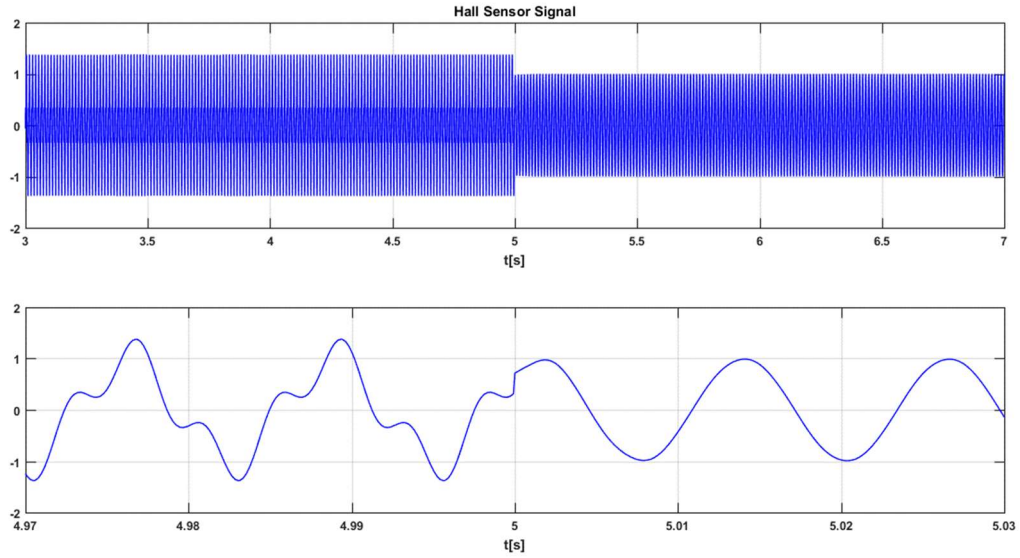

Figure 44 - Hall Sensor Signal 1 before and after ANF activation

The Fast Fourier Transform is performed to evaluate the filtering out of the harmonics. The Figure *45* compares the FFT of the Hall 1 Signal, before and after the ANF filtering.

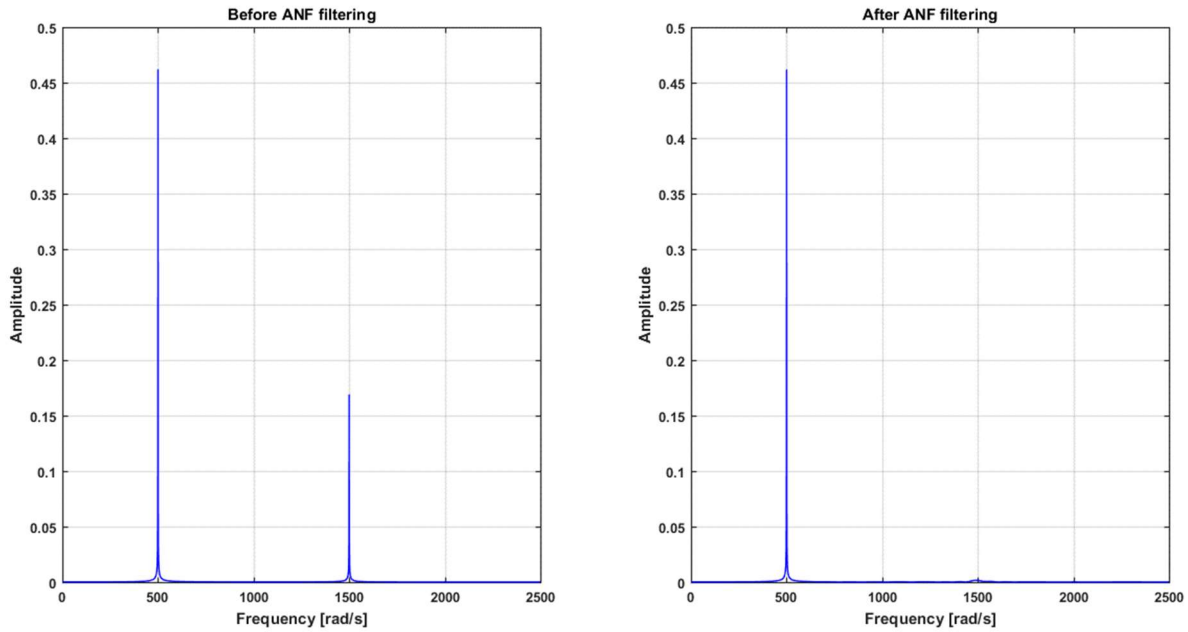

Figure 45 - FFT of the Hall 1 Signal before and after the ANF filtering

Now, let's considerate the ramp input with a slope of 100 rad/s² depicted on Figure *46* and, for instance, the ANF is not activated. Thus, the proposed simulation aims to understand the behaviour of the system along the entire range of possible speeds. The Figure *47* shows the PLL estimation error.

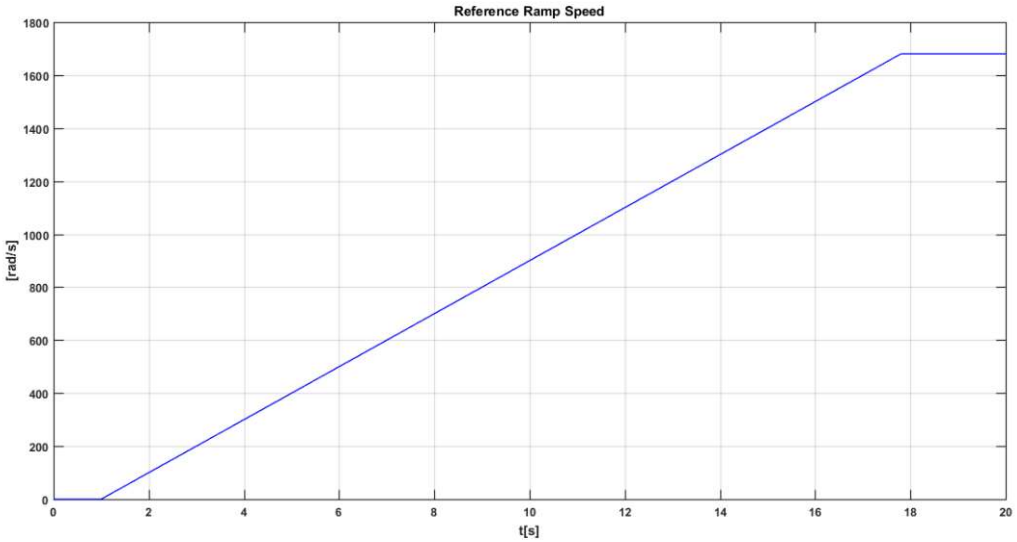

Figure 46 - Reference Ramp Speed - 100 rad/s²

It is notable that the PLL estimation error becomes small as the speed increases. Therefore, the PLL increases the accuracy of the angular estimation even under distortional conditions as the presence of harmonics. However, as saw before, filtering the harmonics by means of the ANF can reduce even more the error amplitude and frequency of oscillation.

The Figures *48* and *49* illustrate the aid on the estimation accuracy guaranteed by the ANF system, at least above a certain minimum speed. Of course, some problems are also depicted and will be discussed below.

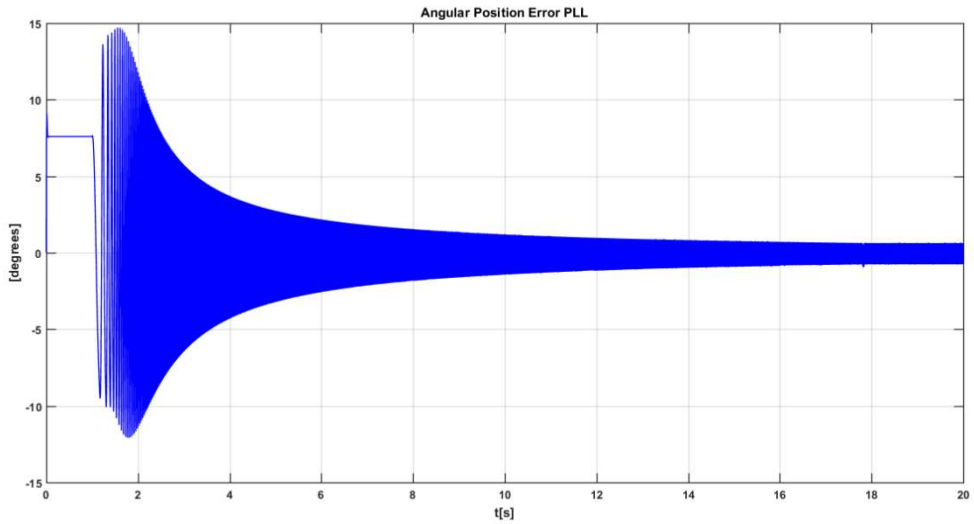

Figure 47 - PLL estimation error without the ANF

These two representations, the arctangent and the PLL estimation error, indicate also the near to zero error region in detail, since the low speed region presents the error larger enough to overscale the graphs.

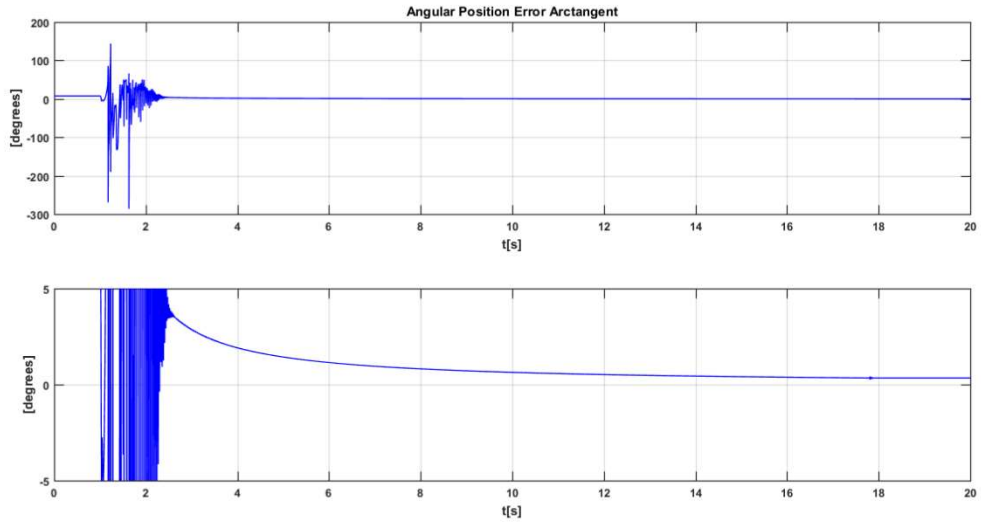

Figure 48 - Arctangent estimation error with the ANF

The previous resolver configuration presents a very high error on the angular position estimation at low-speeds. The peaks are about 300 degrees of difference and the normal amplitude about 100 degrees, decreasing as the speed increases. Above the angular speed of 140 rad/s the error approximates to zero, however a small phase delay is still present.

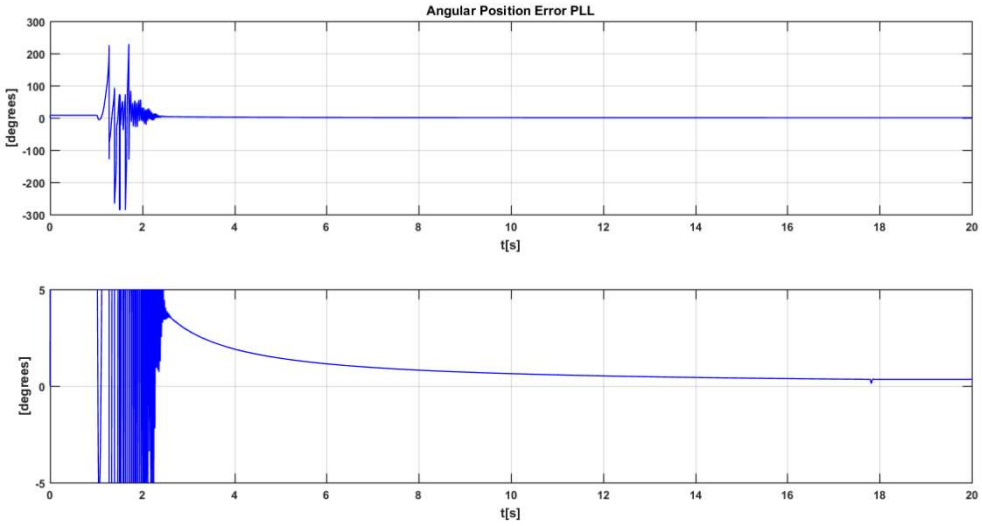

Figure 49 - PLL estimation error with the ANF

8.3. Adaptive Notch Filter distortion

Considering the ANF transfer function, a problem comes to light on regards the low frequencies region. Referring to the Figure *27,* plots of the Magnitude and Phase of the ANF system considering the specific fundamental frequencies, and the Equation *22,* the ANF transfer function expressed once more below, as said before, it is notice a changing on phase and magnitude, the ANF acts as a first order high pass filter at the frequencies of interest with 10 rad/s as the cutoff frequency.

$$\frac{\bar{H}(s)}{H(s)} = \frac{s^2 + {\omega_h}^2}{s^2 + \sigma s + {\omega_h}^2} \tag{47}$$

The quality factor was chosen as 80, to achieve a fast response. Maybe it is the problem. But, we must to consider that low-speeds (ω <140 rad/s) is equivalent to the vehicle linear velocity of about 11 km/h, and that the time in such range of speeds is relatively small. So, fast response is an important constraint.

Therefore, a possible solution would be an Ideal Controller, to correct the phase and amplitude of the signal.

$$C_1(s) = \frac{s - 10}{s} \tag{48}$$

However, such controller introduces a dc offset that is related with the integrative gain and the initial condition of each signal, as shown in Figure *50.* Of course, the ANF angular position input needs to be the reference one due to the subsequent transformations and consequentially the PLL are not able to work with this kind of signals and presents higher errors on the estimation.

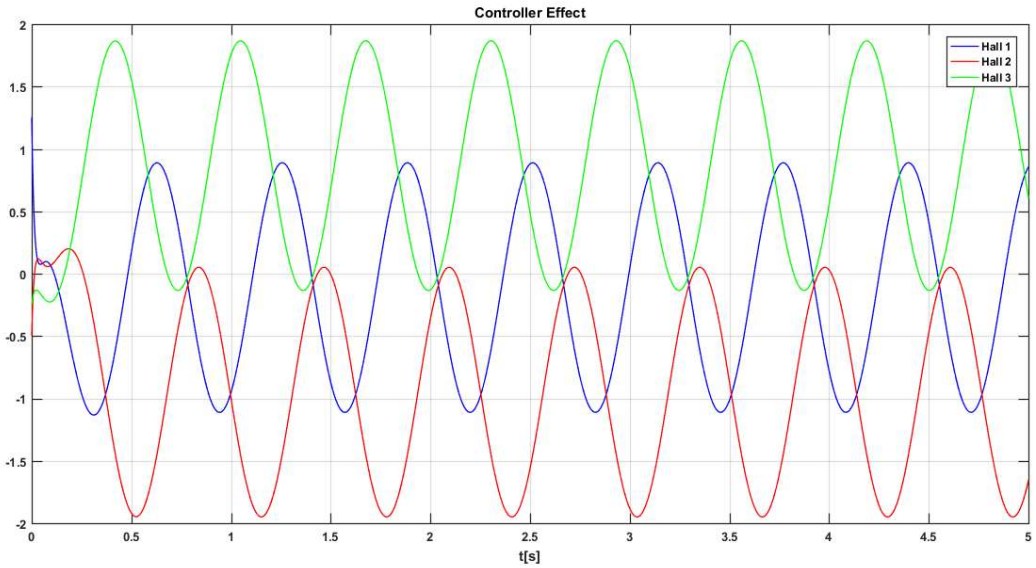

Figure 50 - Ideal Controller effect on the Hall sensors Signals

Then, a solution can be to calculate and subtract the mean value of each signal. The mean value subtraction method to remove the dc offset of the signals is a difficult achievement. The greater problem here is to find the peaks and valleys, but, once they are found, the mean value is calculated as:

$$X = \frac{Peak + Valley}{2} \tag{49}$$

However, after some tests, this method was effective only when the angular position feedback was Ideal (the input angular position).

Thus, another controller is proposed without an integrative gain, to correct at least the phase delay at low frequencies. Considering the ANF bode diagram, the output of the Adaptive Filter in Laplace transform is given as:

$$H_{ANF}(s) = A * \frac{s * \cos(\varphi) - \omega * \sin(\varphi)}{s^2 + \omega^2} \tag{50}$$

Where, A is the amplitude reduction and φ represents the introduced phase delay. They are given by:

$$A = \left(\frac{\omega}{\sqrt{\omega^2 + 100}}\right) \tag{51}$$

$$\varphi = \text{atan}\left(\frac{\omega}{10}\right) - \frac{\pi}{2} \tag{52}$$

Hence, the controller needs to modify the phase to zero, that is introducing $-\varphi$. Thus, a lead-lag controller is introduced, where the zero (z) is much smaller than the pole (10), going to zero.

$$C_2(s) = \frac{s+z}{s+10} \tag{53}$$

Moreover, although not extremely necessary, the amplitude of the signal can be corrected with a gain, function of the angular speed ω since the proposed controller changes the gain of the system with the value of A on Equation *50,* considering the two changes on the Amplitude, the correction factor is given by:

$$Correction\ Factor = \frac{\omega^2 + 100}{\omega^2} \tag{54}$$

However, although these efforts improved the result correcting the phase delay and consequentially the estimation error, they were not very effective on regards the initial oscillatory error, that is still present at low frequencies. The Figure *51* illustrates the phase correction on the PLL estimation.

The error estimation at speeds higher than 140 rad/s is less than 0.05 degrees, that is a very low error at a wide range of speeds. Nonetheless, the low-speed problem is still present. From that, another possible cause is discussed next.

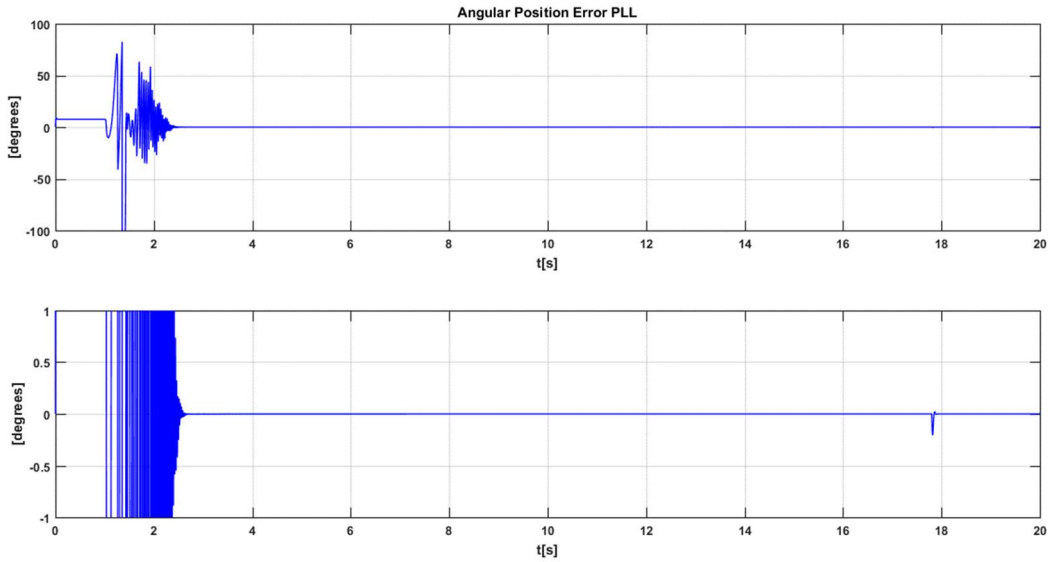

Figure 51 - PLL estimation error with the ANF and controller

8.4. PLL poles location

The PLL poles location were chosen considering the response velocity and damping coefficient only. However, we must to consider that the poles of the loop filter are also the most significant factors on regards the PLL bandwidth.

Considering the proposed Loop Filter and the overall PLL closed-loop transfer function, the PLL Bandwidth is given by:

$$BW = \sqrt{k_i\left(1 + 2\xi^2 + \sqrt{2 + 4\xi^2 + 4\xi^4}\right)} \tag{55}$$

$$BW \approx 2.0490\sqrt{k_i} \tag{56}$$

Where, k_i is the integrative gain of the PI controller.

The lock time decreases with the increase of the bandwidth. The higher the bandwidth the faster the PLL can adjust the output frequency, and so, the faster it can lock. The fast response is interesting when there are changes on the input frequency as on motor applications.

However, the larger the bandwidth the harder is the control of the PLL, increasing output noise disturbance. The problem is increased when the input presents 3rd order harmonics, generating higher orders on the q component of the Park transformation, being necessary the use of the Adaptive Filter. Nonetheless, at the start-up, the signals are not filtered causing a lot of instability, mainly in the low frequency region.

A possible solution would me minimize the closed loop poles and so, the bandwidth to ensure the lock of the PLL by means of the improvement of the interference-rejection. Again, we have some issues: the capture process become slower, the capture acquisition range decreases and the transient response to sudden change of the input become underdamped.

Considering just the first issue, the lock time is increased and the same problem of the slowly response takes place. Again, the operational time in the low-speed region is relatively small, maybe not enough to acquire lock with this small pole configuration.

Nonetheless, it is possible an adaptive PI controller configuration, with changes on the proportional and integrative gains considering the estimated angular speed. The minimum and maximum speeds are chosen as 45 and 140 rad/s. Above the maximum speed, the PI gains remains the same, however below the minimum speed the PI gains are calculated as follow:

$$k_{i2} = \frac{k_i}{400} \tag{57}$$

$$k_{p2} = 2\xi\sqrt{k_{i2}} \tag{58}$$

Within the third region, between the minimum and maximum velocities, the PI gains are calculated by the follow equations.

$$k_{i\_adap} = \frac{((k_i - k_{i2}) * (\omega_0 - 45)}{140 - 45} + k_{i2} \tag{59}$$

$$k_{p\_adap} = 2\xi\sqrt{k_{i\_adap}} \tag{60}$$

The Figure *52* illustrates such configuration.

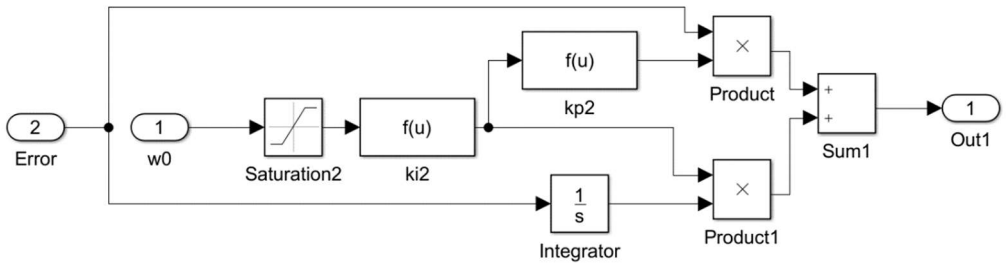

Figure 52 - Adaptive PI Loop Filter configuration

Using the reference speed as the frequency input of the PLL system, the PLL estimation error has a significative reduction, it is below the 10 degrees at the low speed region. That is depicted on Figure *53*.

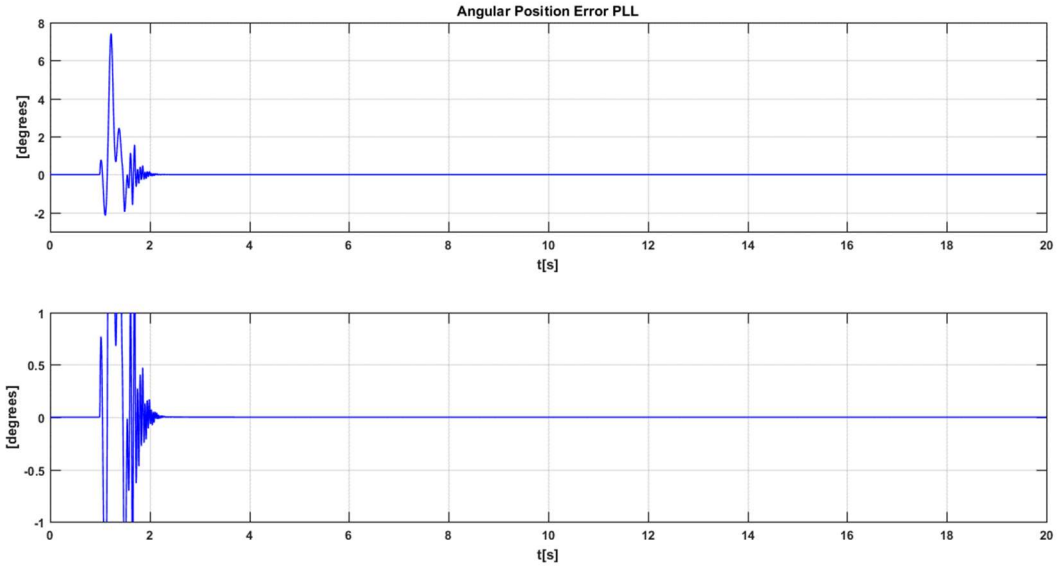

Figure 53 - PLL estimation error with the ANF and adaptive Loop Filter using the reference speed

On regards the use of the estimated speed as the frequency input of the PLL system, is observed an increasing on the error at very low speed region due to the speed estimation error be very high on that region. The Figure *54* illustrates the PLL error considering also changes on the minimum PI gains to achieve better performance.

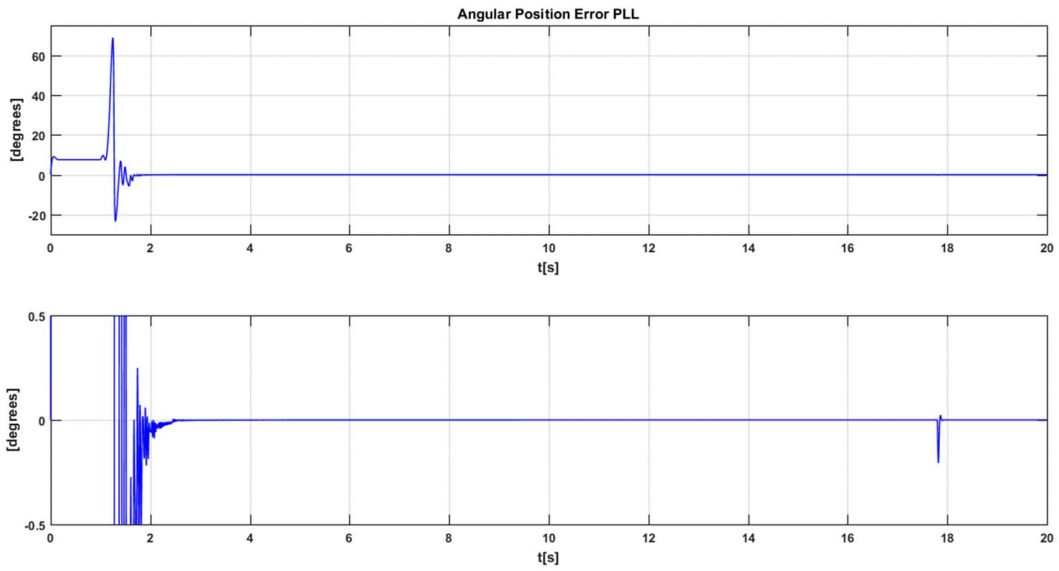

Figure 54 - PLL estimation error with the ANF and adaptive controller using the estimated speed

Hence, according to the last simulation, it is very important increasing the accuracy of the speed estimation, mainly at low speed in order to guarantee a precise angular position estimation on the PLL.

8.5. Speed estimator improvement

The performance of the two speed estimators proposed on section 7.6 is evaluated now. Considering the ramp input of the previous simulations, the estimated speed and the estimation error are plotted for both the variable-time based speed estimator and the arctangent differentiation one, on the Figures *55* and *56* respectively.

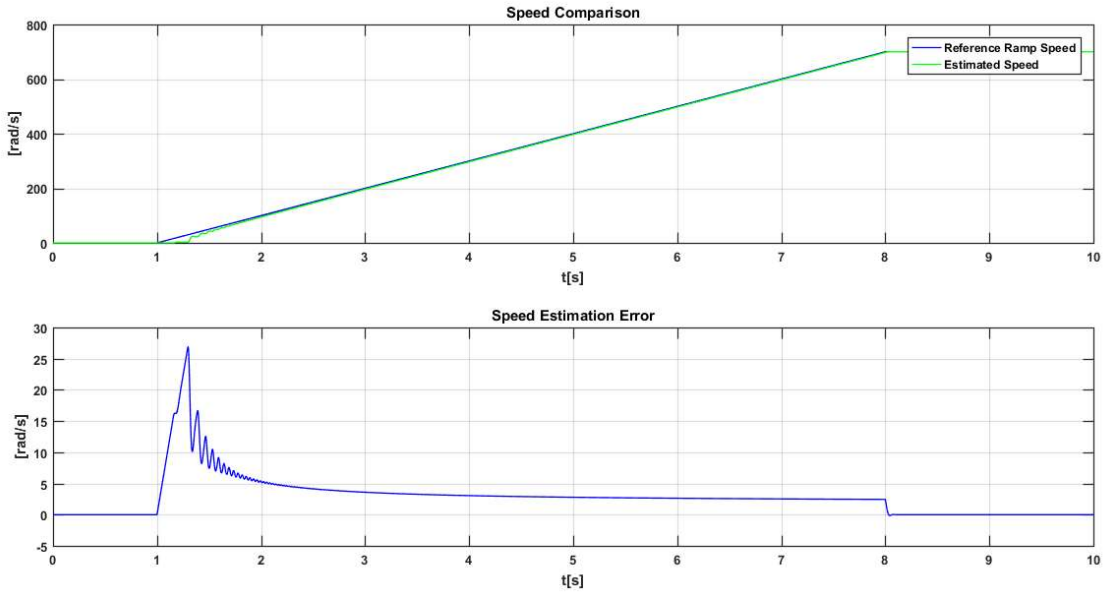

Figure 55 - Variable-time based estimation speed comparison and error

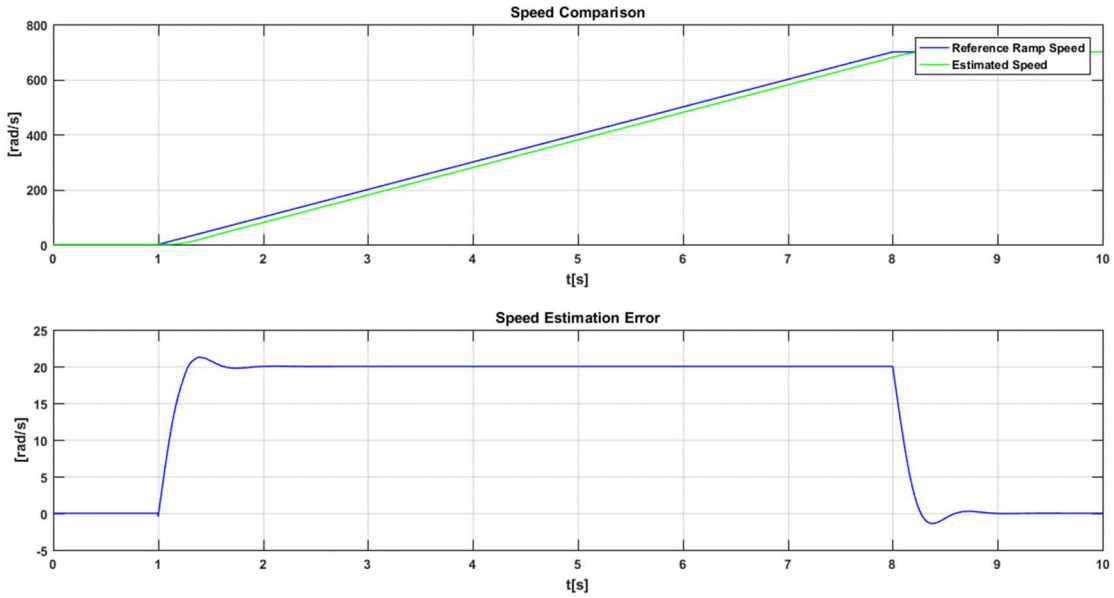

Figure 56 - Arctangent differentiation estimation speed comparison and error

8.5.1 Variable-time based speed estimator

Although the high precision of the Variable-time based speed estimator on regards high speed input even at the ramp, the low-speed region at the start-up needs a special attention. The problem here is, as mentioned before, that the estimator needs to wait a second pulse in order to calculate the speed, such a pulse takes a long time to be identified, thus, for example, zero speed is estimated instead an actual non-zero one.

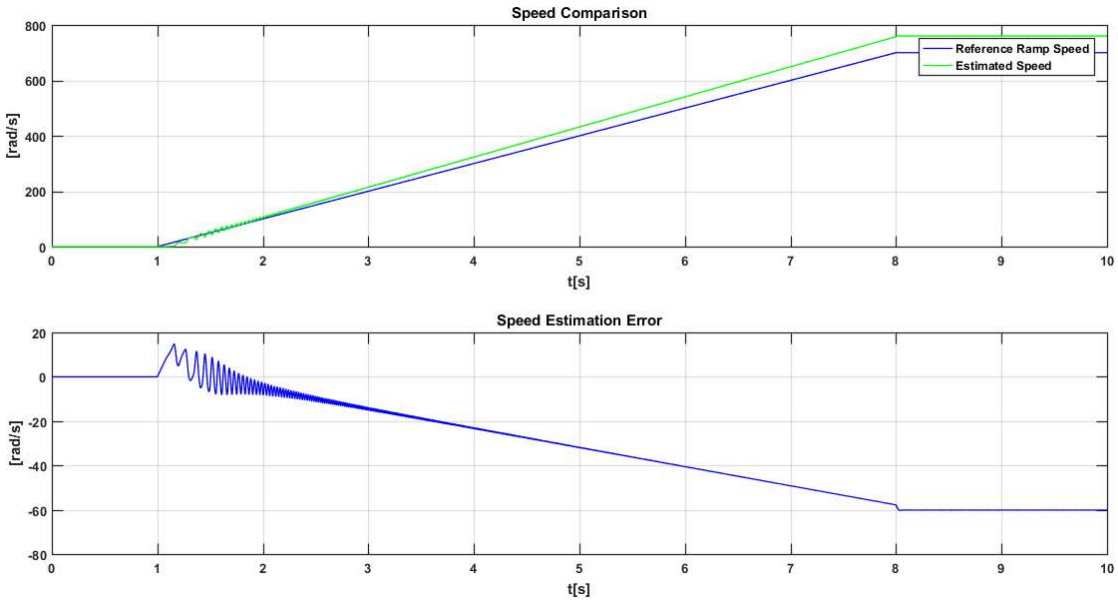

Figure 57 - Variable-time based estimation speed comparison and error using the three-phase Signals

The time between the calculations can be reduced using the three-phase hall signals instead one of them on the estimator. However, as mentioned before and shown on Figure *57*, due to the irregular shape of the nonfiltered hall signals the estimation suffers with high oscillations that increases as the speed increases and does not give a precise estimation but a slightly higher one after the filtering.

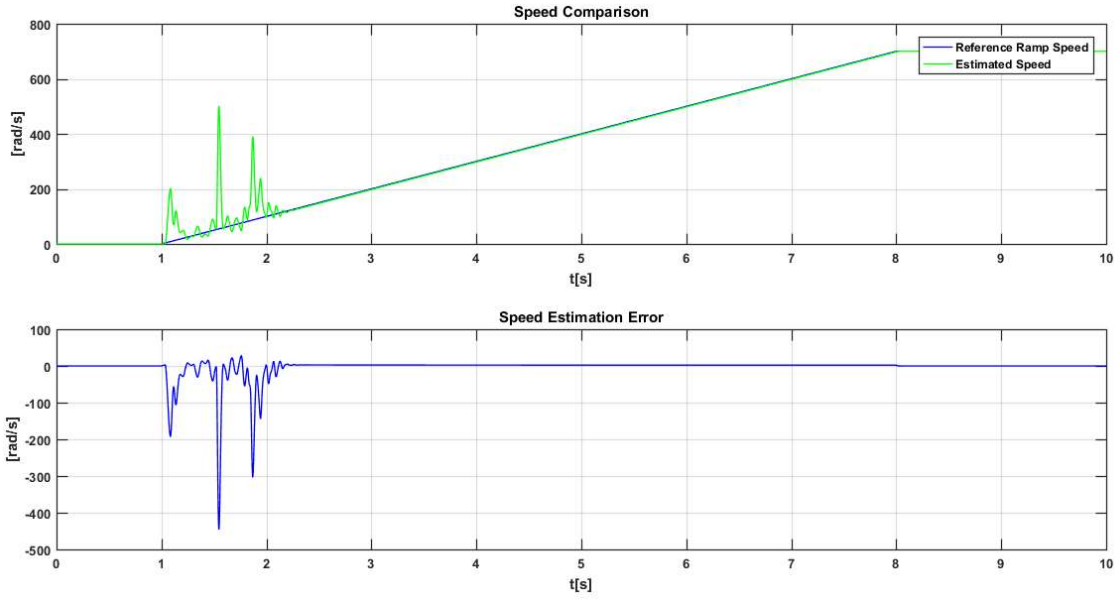

Figure 58 - Variable-time based estimation speed comparison and error using the filtered out three-phase Signals

Moreover, considering the filtered signals, the start-up region can be rather difficult to deal. That work condition is characterized by the even higher irregular shape of the signals due to the calibration of the ANF system and the unlock of the PLL. Thus, as shown on Figure *58*, higher oscillation is present there, although the larger resolution on the high-speed region on regards the original one-signal configuration.

A possible solution would be changing the cut-off frequency of the estimator filter to a

smaller one about 8 rad/s instead the 100 rad/s. Nonetheless, the speed estimation error on the ramp becomes noticed due to the filter effect. Moreover, the smaller cut-off frequency is only necessary before the PLL lock at the low-speed region. Therefore, after that, the high cut-off frequency is preferable for a better performance being necessary a smoothly transition between the signals of the two filters.

The Figure *59* shows the estimation speed and error of this solution. The estimator performance at the very low-speed region, $\omega < 40\, rad/s$, is the best until now. Moreover, although not completely, the error peak is in part neglected by the PLL system.

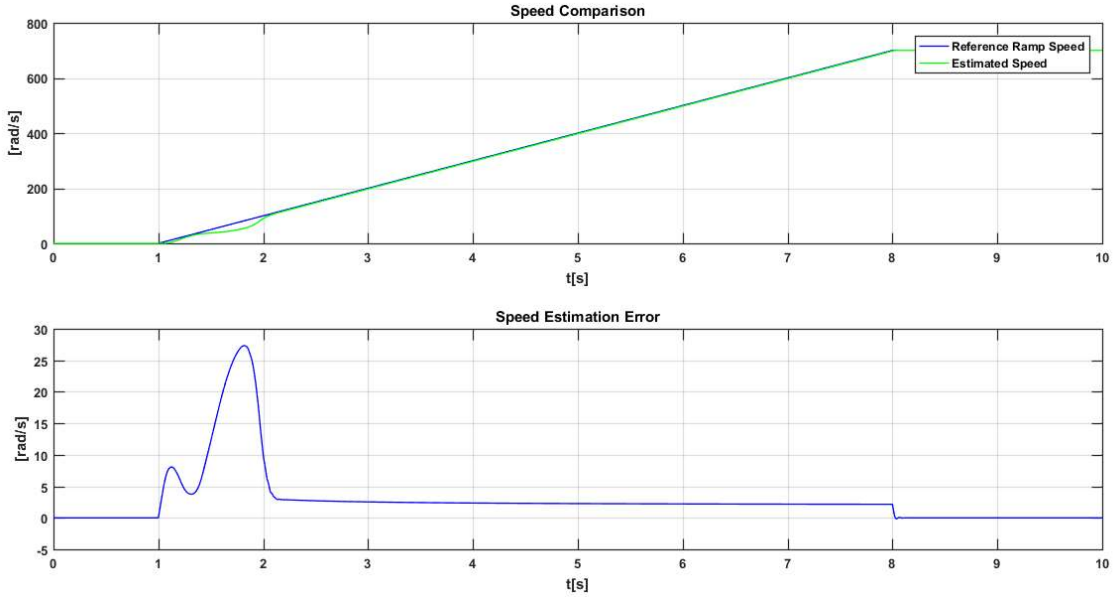

Figure 59 - Variable-time based estimation speed comparison and error with the filtered out three-phase Signals and adaptive filter

Thus, the PLL angular position estimation error for this configuration is shown on Figure *60*.

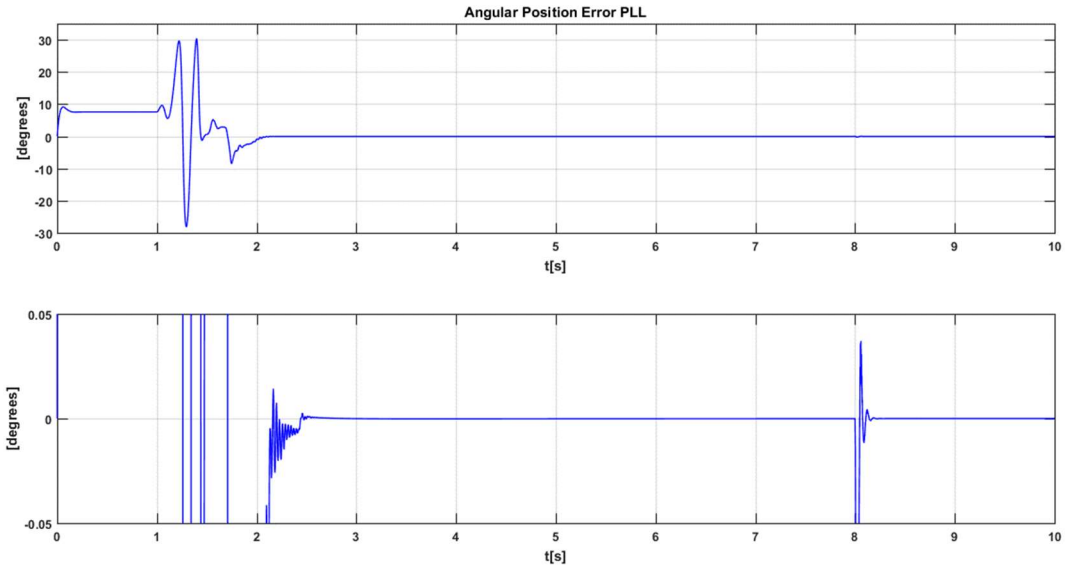

Figure 60 - PLL estimation error with the ANF, the adaptive controller and the modified speed estimator

The maximum error peak from the start-up point of view is about 30 degrees, a significant improvement until now. However, another work condition must be considered: low

speed region when decreasing the speed to zero.

Hence, to be better evaluated, lets considerate the reference speed profile depicted on the Figure *61*.

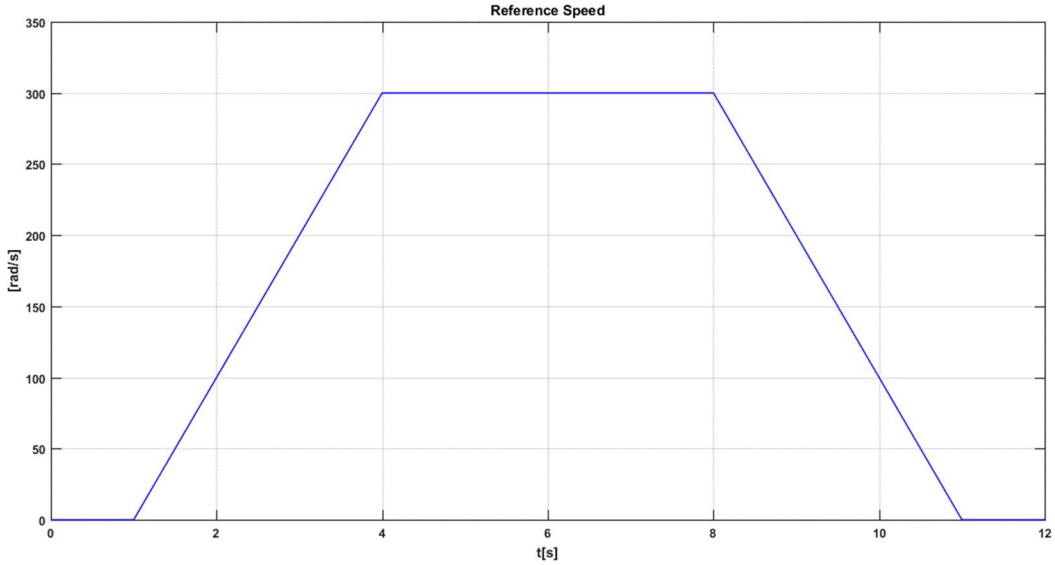

Figure 61 - Reference Speed Positive and Negative ramp

The PLL angular position estimation error is depicted on Figure *62*. The close to zero speed decreasing region shows a significant error of about 40 degrees.

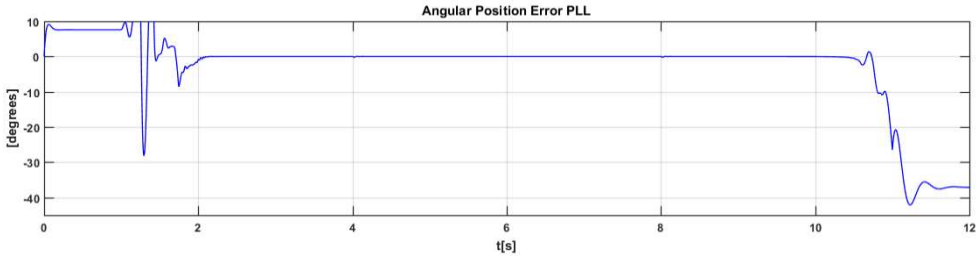

Figure 62 - PLL estimation error. Detail of close to zero speed decreasing region.

8.6. Bypassing the Adaptive Filter

On regards the previously considerations, for now, a suitable choice would be not to use the ANF on the low-speed region. Thus, when the speed overcome 140 rad/s, the ANF is activated and, when decreasing, if the speed becomes smaller than 120 rad/s, the ANF is disactivated.

Hence, the angle estimation shows an oscillatory error due to the not filtering of the signals like on Figure *47*. Nonetheless, the Figure *63* shows that the amplitude of the presented error is below the 15 degrees and the accuracy of the high-speed region is not compromised.

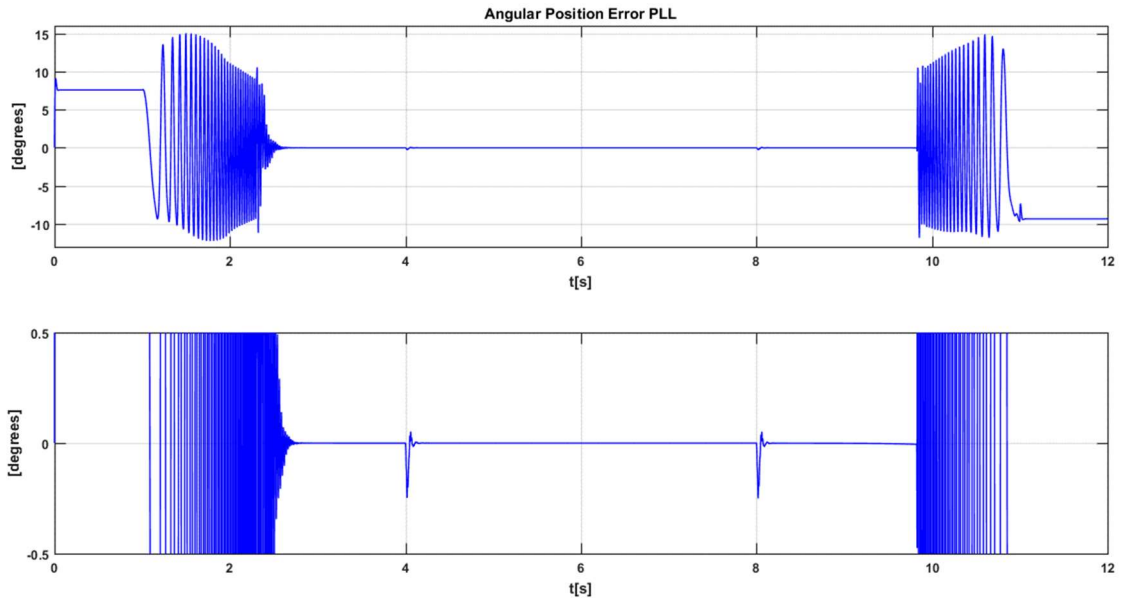

Figure 63 - PLL estimation error bypassing the ANF at low-speeds

CHAPTER 9

CODE GENERATION AND PROGRAMMING

9. CODE GENERATION AND PROGRAMMING

The present chapter aims to develop the process of code generation and programming of the Virtual Resolver. As said before, the Virtual Resolver will be implemented on the *Risciò* EKU. Thus, some steps are needed in order to generate an optimized code in C language that can be well matched with the already implemented applications on the mentioned EKU.

The *Simulink* Virtual Resolver Model implemented on the chapter 7 of this thesis is now used as base for an automatic C code generation by the *Matlab-Simulink* development tools. Thus, since the EKU board does not work with the continuous-time approach that was used to develop the models, it is necessary the Model discretization. As mentioned before, a certain attention will be taken in the specific case of the arctangent differentiation.

Moreover, although all the paid attention to develop the models aiming the code generation, some issues are found. Therefore, some modifications are done in order to accomplish the good modelling practices to optimize the overall system.

After that, the C code is generated separately for each subsystem of the Resolver Model, considering the target EKU application and board.

Finally, the *Code Composer Studio* is used to a final matching with the EKU applications. In such an environment, the generated code will be put inside the overall EKU Code using a small handwritten part of code.

9.1. Discretization and modifications of the models

The Virtual Resolver components, that is, the First Conditioning system, the Adaptive Notch Filter, the Clarke Transformation, the Phase Locked Loop, the Arctangent Logic and the Speed Estimator must to be discretized to accomplish the board requirements.

The first constraint is on regards the frequency of execution. In order to follow properly the input signals, both in frequency and shape, a sample time *Sample_Time* is calculated using the Nyquist frequency and considering the highest input frequency of the Hall Signals.

	Continuous Transfer Function	**First Order Discrete Transfer Function**
First Conditioning $\boldsymbol{\omega_c = 10000\ rad/s}$	$\frac{10^4}{s + 10^4}$	$\frac{0.3533 - 0.3533z^{-1}}{1 - 0.2934z^{-1}}$
ANF phase controller	$\frac{s + 0.0001}{s + 10}$	$\frac{1 - z^{-1}}{1 - 0.999z^{-1}}$
Speed Estimator $\boldsymbol{\omega_c = 8\ rad/s}$	$\frac{4096}{s^3 + 128s^2 + 1024s + 4096}$	$\frac{0.0003998 + 0.0003998z^{-1}}{1 - 0.9992z^{-1}}$
Speed Estimator $\boldsymbol{\omega_c = 50\ rad/s}$	$\frac{12.5 * 10^4}{s^3 + 100s^2 + 5000s + 12.5 * 10^4}$	$\frac{0.002494 + 0.002494z^{-1}}{1 - 0.995z^{-1}}$

Table 1 - Continous to discrete convertion

The model modifications were made to substitute the continuous-time blocks to the

discrete ones, considering the sample time of $1 * 10^{-4}$ seconds. Therefore, the *Matlab* command "*c2d"* was very effective to convert the already designed filters and controllers in each subsystem to discrete systems. A special attention must be taken when discretizing high order filters, it is preferable to work with a cascaded configuration of lower order filters.

Considering the first order transfer function of the cascaded configuration, the continuous-to-discrete conversion is shown on the Table *1*. Moreover, discrete-time integrators were put in the place of the continuous ones.

Another issue to be considered was the redundancy or unnecessary calculations, for example the third order sine and cosine calculations on the ANF system. The math functions were running for each phase of the signal; thus, they were modified to be performed before the estimation of the harmonics coefficients for each signal, just one time.

On regards the use of the arctangent differentiation to estimate the speed, a discrete time approach gives rise to overcome the issues before mentioned. The angular speed is calculated as:

$$\omega_n = \frac{\theta_n - \theta_{n-1}}{\Delta T} \tag{61}$$

Where, θ_n is the angular position at the instant *n,* θ_{n-1} the previous angular position and ΔT is the sample time. The angular position difference can be evaluated by the arctangent function of sine over cosine values:

$$\theta_n - \theta_{n-1} = \arctan\left(\frac{\sin(\theta_n - \theta_{n-1})}{\cos(\theta_n - \theta_{n-1})}\right) \tag{62}$$

It follows that:

$$\sin(\theta_n - \theta_{n-1}) = \sin(\theta_n) * \sin(\theta_{n-1}) - \cos(\theta_n) * \cos(\theta_{n-1}) \tag{63}$$

$$\cos(\theta_n - \theta_{n-1}) = \sin(\theta_n) * \cos(\theta_{n-1}) + \cos(\theta_n) * \sin(\theta_{n-1}) \tag{64}$$

Finally, lets considerate the Alpha and Beta signals, ideally, the cosine and the sine of the fundamental angular position, respectively. Then, in a discrete approach, it gives:

$$\cos(\theta_n) = Alpha_n \tag{65}$$

$$\sin(\theta_n) = Beta_n \tag{66}$$

$$\cos(\theta_{n-1}) = Alpha_{n-1} \tag{67}$$

$$\sin(\theta_{n-1}) = Beta_{n-1} \tag{68}$$

Hence, ω_n can be properly calculated with the signals provided by the Clarke transform and its previous values.

Moreover, the Arctangent Logic can be improved on time of execution using a pre-calculated look-up table instead the math function calculation. A bi-dimensional look-up table

was chosen since there is two inputs: the Clarke Transformation Alpha and Beta signals. The same algorithm presented on *Equation 2* was used to generate the table and the auxiliary *.m* file is presented below.

```
Beta=-1:0.001:1;
Alpha=-1:0.001:1;
for i=1:length(Alpha)
    for j=1:length(Beta)
        if ((Beta(j)>=0) & (Alpha(i)>=0))
            tab_atan(i,j)=atan(Beta(j)/Alpha(i));
            elseif (Alpha(i)<0)
            tab_atan(i,j)=pi+atan(Beta(j)/Alpha(i));
        else
            tab_atan(i,j)=2*pi+atan(Beta(j)/Alpha(i));
        end
    end
end
```

Finally, the variable names and the variable types were changed to make the generated code as readable and reusable as possible, and to make the most of the board available resources respectively. The outputs' names of each subsystem must to match with the inputs' names of the next subsystem. Moreover, these inputs and outputs must to be treated as the variables of the code and to be named properly.

9.2. Automatic code generation

In order to automatically generate the C code, as said before, each subsystem is treated as a unit. A separated *Simulink* model is open and configured for each component. Therefore, the final models of the principal subsystems are presented below, from the Figure *64* to the Figure *67*.

Hence, the C code is generated using the available tools on the *Simulink* environment. Further configurations on regards the code generation involve the selection of the compiler, the style of generated code, sample time selection, embedded target selection, and others.

Finally, the automatic code generation outputs two main files with the extensions .c and .h for each subsystem, they need be included on the overall EKU project.

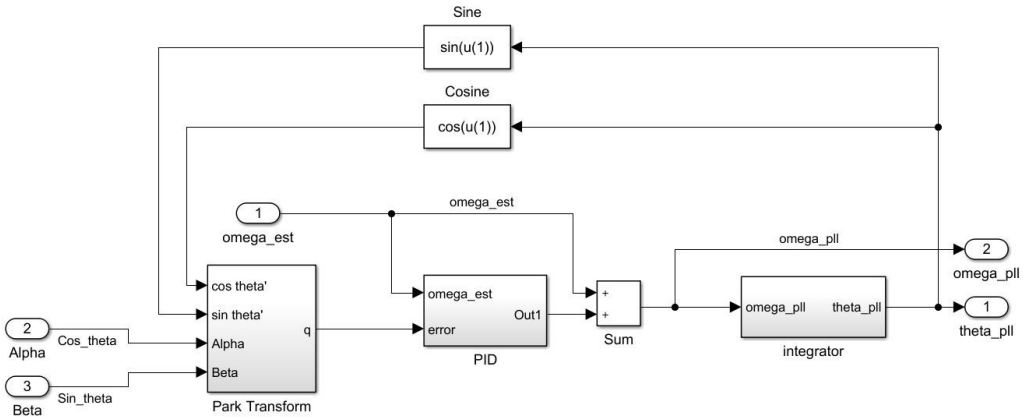

Figure 64 - Modified PLL

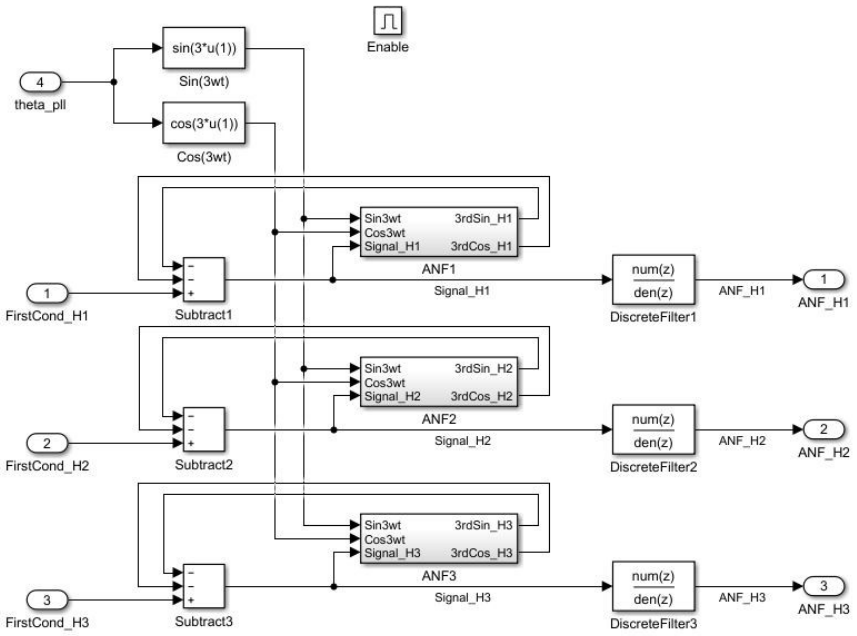

Figure 65 - Modified ANF

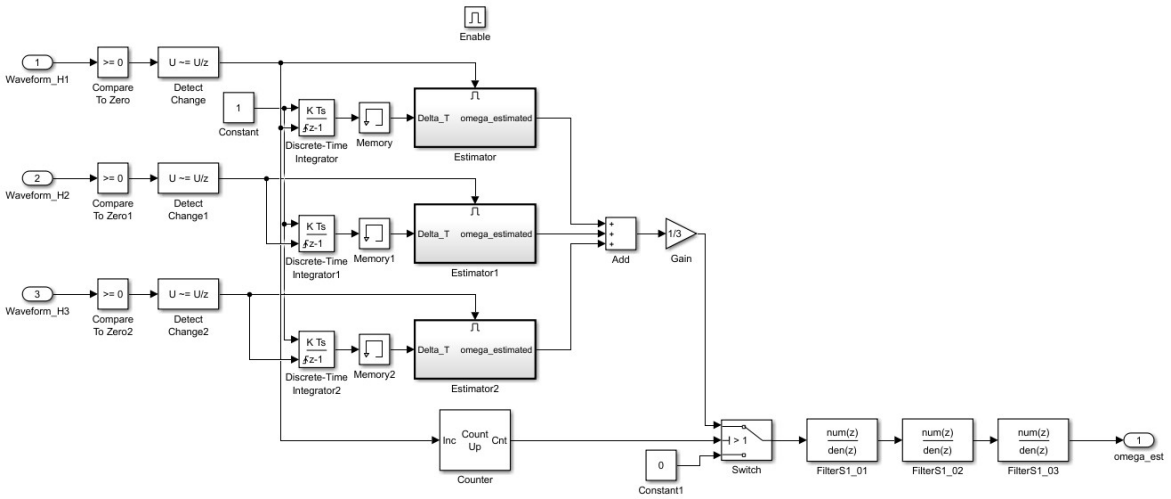

Figure 66 - Modified Speed Estimator

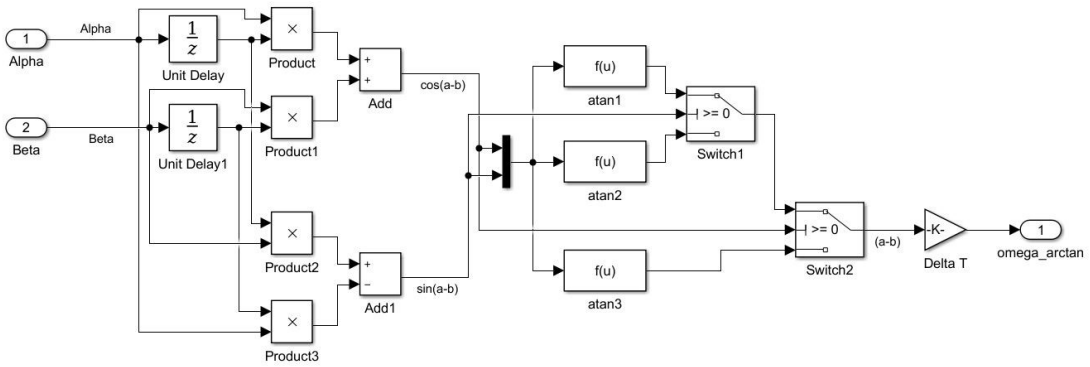

Figure 67 - Modified Arctangent Logic and Differentiation

9.3. Programming

A handwritten programming is necessary to include the generated code on the system. Among others, the declaration of the functions, definition of the variables, attribution of first values and establishment of a sequence for the execution of the subsystems are needed in such a step. They were written also in C language inside the already built interrupt routine USER_TIMERA of the EKU base program.

As said before, the handwritten code is made in such a way to match with the already present applications on the EKU base program. It was considered the time of execution, sequence priorities, available variables inside the code, organization, among others.

A special attention was taken on regards the Speed Estimator. Since the interruption subroutine of the code allows a more effective time counting, it is used to measure the time between the pulses.

Moreover, both, the generated code and the handwritten one can be re-used and part-used as well. Since the features of the *Simulink* code generation allows a readable and re-usable, it's possible to extract the exact portion of the program that is interesting for the different applications, which is plenty exploited in the next section for tests and validation purposes.

At this point, the Virtual Resolver is ready to be implemented on the EKU board.

CHAPTER 10

HARDWARE IN THE LOOP

10. HARDWARE IN THE LOOP

The development of embedded systems for the electronic control unit of a vehicle requires the execution of several tests considering the complexity of these systems, whether in relation to the control or sensing. However, it is notable that such complexity introduces to these tests factors related to the increase of time and cost, impossibility of execution and repeatability of specifications.

On the other hand, the requirements defined at the beginning of V-model need to be tested, confirming the effectiveness of the system developed for validation purposes. For these purposes, the simulations made in chapter 8 of this work allowed an improved view of the Virtual Resolver through exhaustive tests and modifications.

The virtual environment is one of the most important tools in the general context of the development of these systems. Thus, a reduction in time and costs is possible, as well as, at this point, a physical system or the construction of specific hardware is not necessary for these tests.

However, the embedded system hardware needs to be tested just like in the virtual environment. The interface with external systems and signals is of utmost importance in this context. Thus, we arrive at the model called Hardware-in-the-Loop, where a virtual environment is realized in order to interact with the embedded system of the control unit.

Two main parts of this type of test are highlighted: firstly, the embedded system itself, ie the electronic control unit of the vehicle, secondly, a simulation and emulation system of external signals and conditions.

Simulations of dynamic systems are possible thanks to the enhancement of digital and analog systems including embedded systems with signal converters. It is necessary, however, that such simulations, although based on a discrete time approach, can be felt by the systems under test as actual measurements of the environment. This implies the need for a relatively short sampling time for the simulation systems in relation to the systems tested.

Thus, specific hardware is needed with large processing capacity, digital-to-analog converters, PWM, among others. A virtual environment is prepared and programmed to test the interface of the control unit with an external system.

10.1. Development Boards

At this point, development boards are very useful for the tests purposes. Presenting the same DSP as the *Risciò* EKU and providing inputs and outputs as well analog-to-digital converters, the Delfino development kit allows a controlled platform to test the hardware implementation of the Virtual Resolver. Hence, the generated code and the handwritten one is uploaded to the board by means the *Code Composer Studio v6.*

Considering the second branch of the HIL tests. Hardware-in-the-loop machines are away too expensive for this step of the project. Furthermore, since the involved signals are too fast compared with the series communication protocol, the usual *Simulink* environment will not be effective to simulate the external system without a dedicated platform.

Therefore, to overcome this problem a new configuration was proposed. Another Delfino kit is used in order to generate and emulate Hall sensors signals. The model of the Hall

sensors presented previously builds the second board code allowing a simulation of the required signals in a real environment. Thus, a PWM with variable duty cycle followed by physical low pass passive filters produces the three phase Hall sensors signals with its harmonics.

Moreover, in order to evaluate the effectiveness of the system, first order sine and cosine signals are also generated. They will be processed, and its angular position and speed will be compared with the Virtual Resolver estimated values.

Figure *68* shows this HIL scheme while the Figure *69* shows its physical assembly.

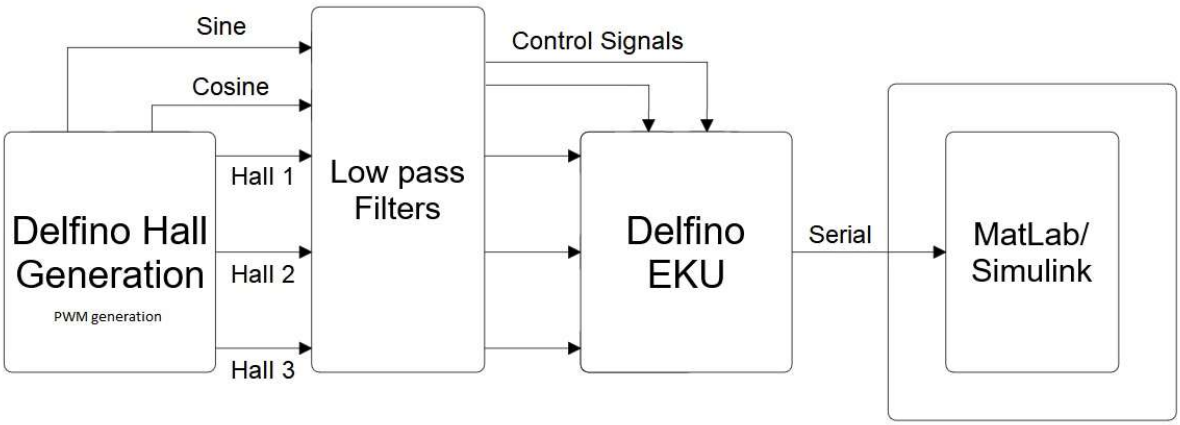

Figure 68 - HIL Scheme with development boards

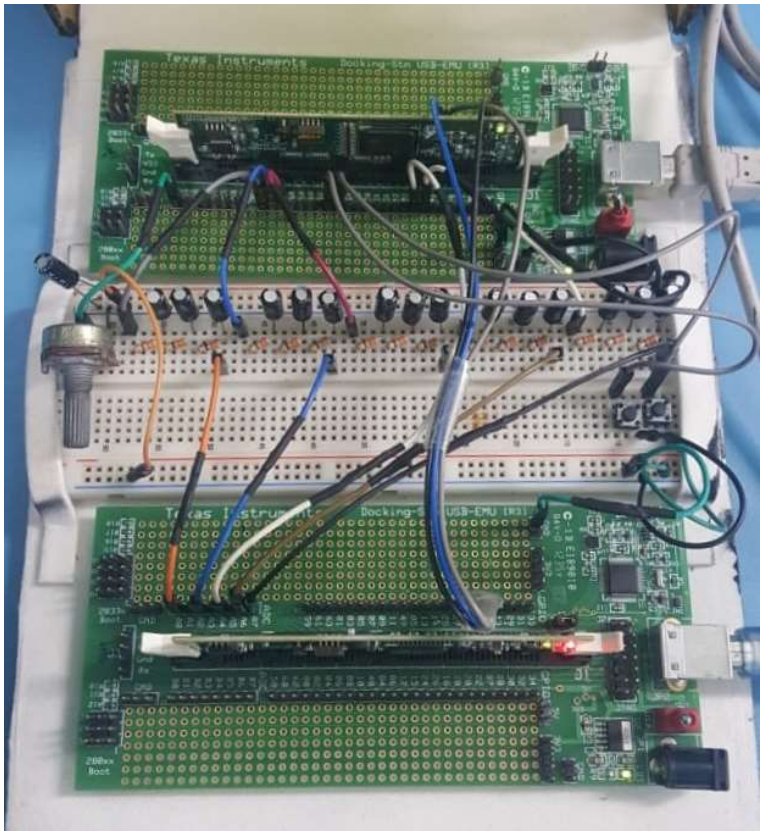

Figure 69 - Physical assembly, HIL tests

10.2. Real Signals Interface

After the hardware configuration is completed, the *Simulink* interface was created to receive the board signals. As the behavioural environment, this scope model is intended to acquire and conditionate the board variables and compare them. Serial interface, unbuffer blocks and scopes are the principal components of the model, as well an Export to Workspace block.

It is noteworthy the limited resource availability. Firstly, the code only allows a small window of a few measures and variables to be transferred at the same time. Secondly, estimation error increases in two main points: although the high processing capacity of the DSP, the Hall signals generation is not perfect, as well the control sine and cosine waveforms. Besides that, the analog-to-digital conversion introduces quantizing errors on the system that contributes with the error estimation.

As the simulations performed on Chapter *8*, the principal achievements will be depicted on the Figures *70* to 75 following the sequence of the previous one

The angular position and the three-phase generated signals are shown on the Figure *70.*

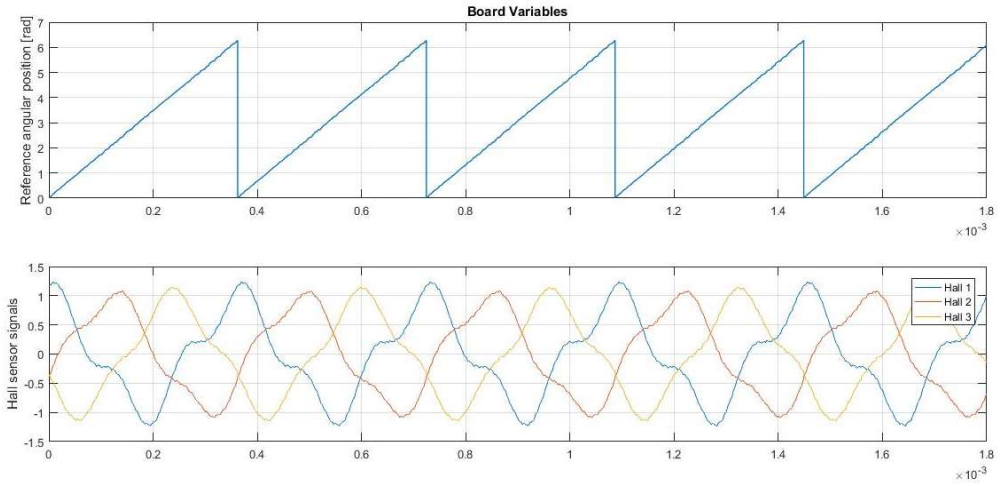

Figure 70 - EKU Reference Angular Position and Hall Sensor Signals

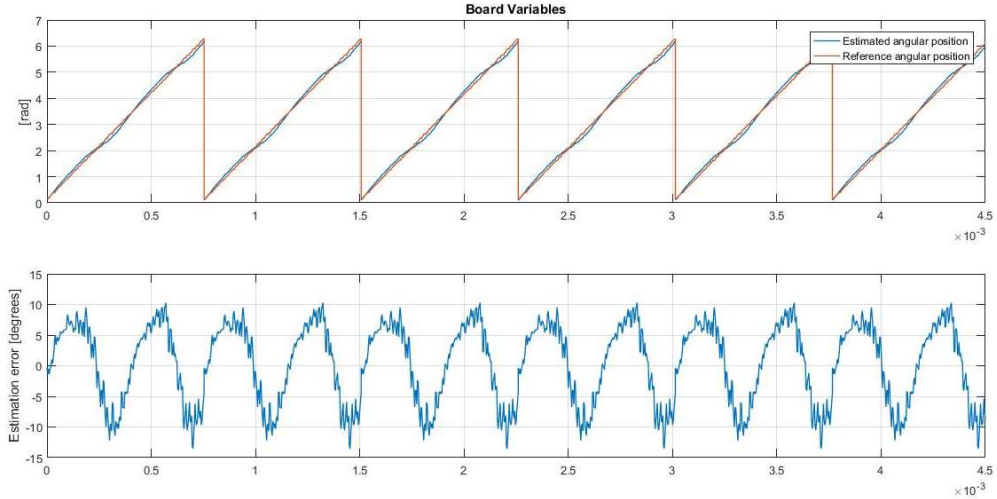

Figure 71 - EKU Arctangent Angular Position Estimation Comparison and Error (without ANF/PLL)

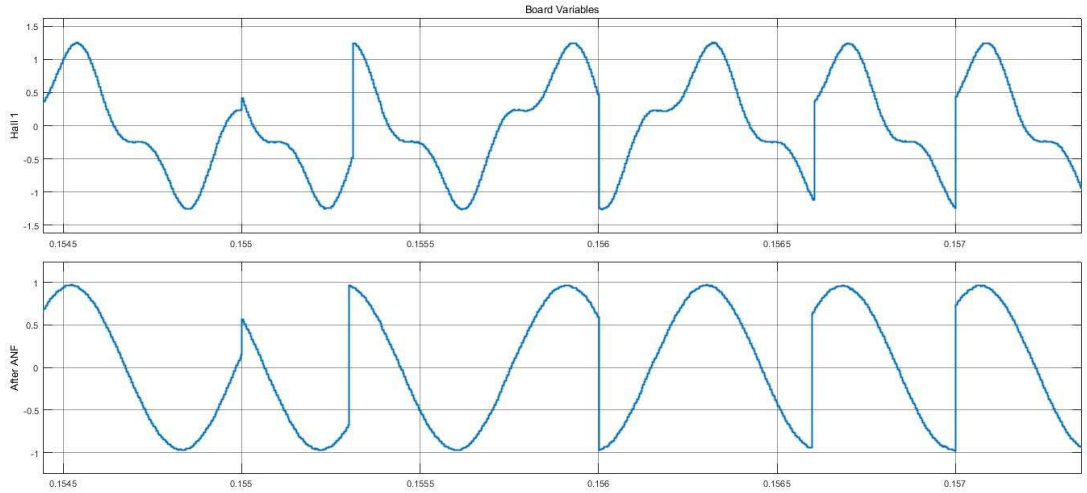

Figure 72 - EKU Hall Sensor Signal 1 before and after ANF activation

Although the window brake in the measures, the Adaptive Filter effectiveness is notable on Figure *72*, where the third order harmonics are filtered out. Moreover, the FFT of the signals are depicted on Figure *73,* the third-order harmonic is imperceptive after the ANF filtering.

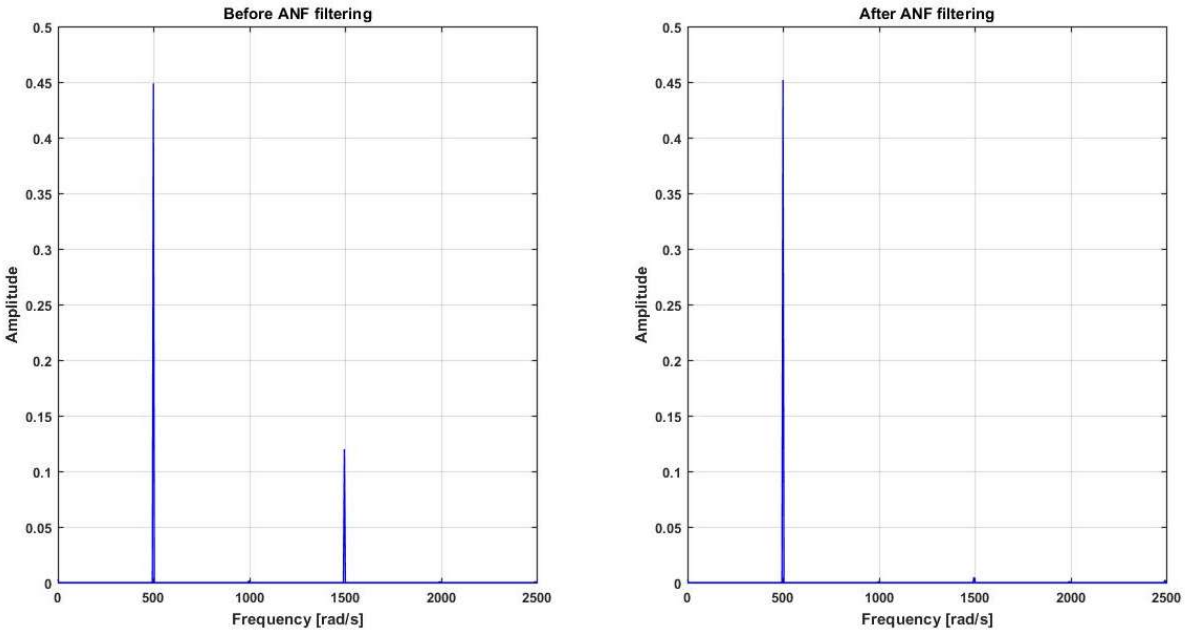

Figure 73 - EKU FFT of the Hall 1 Signal before and after the ANF filtering

As expected, according to Figure *74*, the use of the PLL system increase the estimation precision mainly in the high-speed region ending below 7 degrees when the PLL is locked. Moreover, the ANF is used then to increase even more the accuracy of the Virtual Resolver. Figure *75* shows the ANF estimation error bypassing ANF in the low-speed region. The minimum error is achieved about 1 degree as required, and the low-speed region error is not higher than 17 degrees.

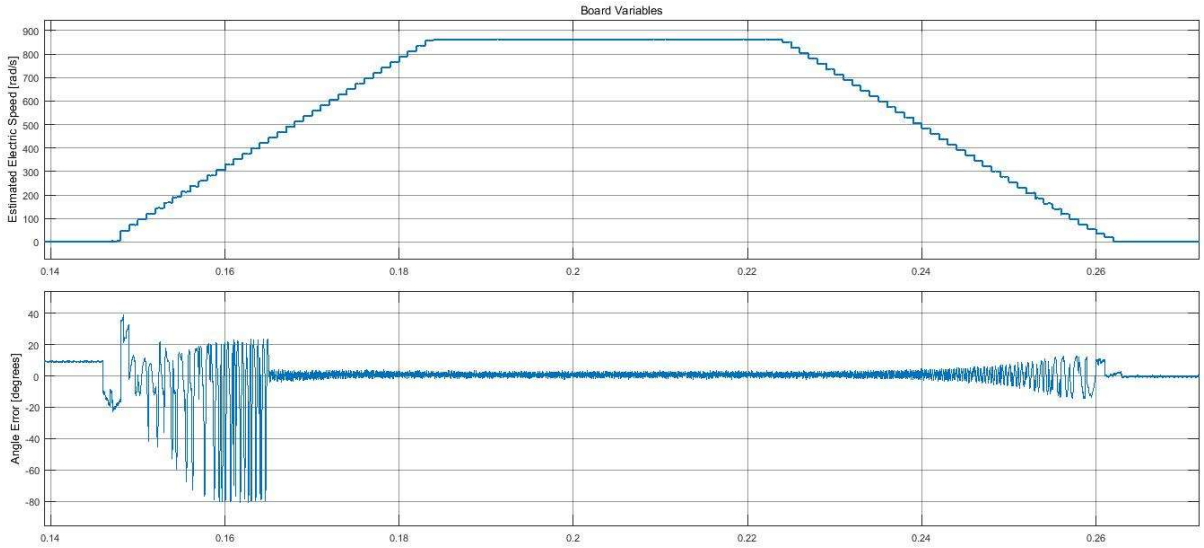

Figure 74 - EKU PLL estimation error without the ANF

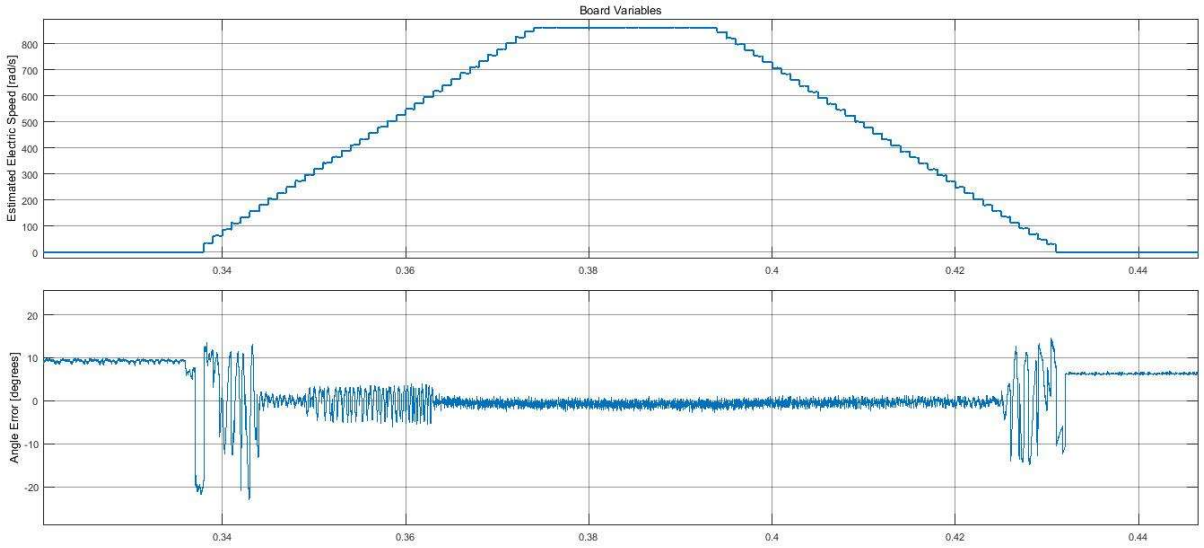

Figure 75 - EKU ANF PLL estimation error bypassing the ANF at low-speeds

CHAPTER 11

THESIS CONCLUSION AND FUTURE WORKS

11. THESIS CONCLUSION AND FUTURE WORKS

The realization of this work allowed a series of improvements on the *Risciò* Project. All the steps of this thesis need to be evaluated in order to discuss the main conclusions statements. Considering the key objectives listed previously, the principal conclusion points are highlighted bellow:

1. Firstly, understanding the problem situation was fundamental to perform the modelling of the physical Hall sensor Board system. According the simulations, the characterization of the signals was the right path to perform a better design. Hence, dividing the system into small subsystems and performing the modelling of each one made possible an inner and overall view. Therefore, analysing these results, this Model-Based Design approach improves the design of the Virtual Resolver itself. However, the accuracy of the Virtual Resolver is conditioned to the precision of each subsystem.

2. Secondly, it is observed that a great degree of accuracy is added when the Virtual Resolver works with its Adaptive Notch Filter and Phase Locked Loop. In perfect conditions, the simulations show a very small angular position estimation error compared with the generated one. Therefore, the implementation of this Virtual Resolver is recommended on projects like the Motor-in-Wheel design. Nonetheless, some problematic areas are identified as the low-speed region before the PLL lock work condition and they need to be better dealt. The speed estimation is a main point of improvement, its enhancement increases significantly the angular estimation as depicted on the simulations results.

3. As the Adaptive Notch Filter, some other adaptive configuration is better evaluated than a fixed and constraint one. It allows the system to follow properly the signals shape and frequency, adjusting itself to different conditions.

4. Finally, the automatic code generation was performed using the *Simulink* tools in order to implement the modelled system in the vehicle *Electronic Kontrol Unit*. However, interfacing the Virtual Resolver with real world signals is not an easy work. Several issue were found on regards the program test interface and signals acquisition. Nonetheless, crossing these lines, an appropriated code is uploaded to the EKU system. Moreover, the systems were tested using the Hardware-in-the-Loop technique. Not too high accuracy is achieved as in the simulations, but the required statement is preserved as 1 electrical degree of precision for the overall Virtual Resolver.

Summarizing, the listed objectives are well accomplished in this thesis. The proposed systems and algorithmics presents reasonable results both for simulations and Hardware Loop tests. Once the problem situation is solved by the Virtual Resolver approach, the resulting code application is useful to be implement as part of the *Risciò* EKU system and makes part of its innovative solutions for electrical vehicles.

The *Risciò* project is in constantly improvements, either for the enhancement of the previously designed solutions and for the design ad implementations of new ones. Considering the theme of this thesis, some proposals to continue its work are presented below:

1. In the first place, the Hardware in the Loop tests were perform using development

boards with the same Digital Signal Processor as the final EKU board. It will be a next step to perform the same tests and evaluate the interaction of the developed code with the hardware of the board and the already implemented code, even if it was designed to match perfectly with them.

2. In second place, following the V cycle, it is necessary to perform a series of tests in a test bench facility with the real motors and sensors signals. Moreover, the estimation precision can be better evaluated on such a facility increasing the reliability of the overall Virtual Resolver and comparing it with the physical solutions. These las test contributes with the validation and homologation of the system.

3. In parallel the Speed Estimator can be improved to increase the Phase Locked Loop accuracy using different algorithmics.

4. Afterwards, the sequence of the angular position and speed estimation could be the development of the motors control. The efficiency of Permanent Magnets Motors is related with the applied control techniques and increases with the availability of feedback information. Once this information is provided by the Virtual Resolver, the control of the motors can be designed considering.

5. Finally, in continuation of the control design, two other discussions can be done. The Differential Electrical Axle that implements a torque split between the motors allowing the separated control of each motor, and a Vector Control to overcome the torque fluctuation due the open stator design

CHAPTER 12

LIST OF FIGURES

12. **LIST OF FIGURES**

CHAPTER 13

REFERENCES

13. REFERENCES

[1] Popovic, R.S. (2004). *Hall Effect Devices, 2nd edn, IOP, Bristol.*

[2] Krishman, R. (2001). *Electric Motors Drives: Modeling, Analysis, and Control, Prentice Hall.*

[3] Avanthi, C. and Chandra Sekhar, O. (2015). "Velocity Estimation from Hall Sensors using Digital Signal Processor and Modelling and Simulation of Electro-Mechanical Actuator". In: *Indian Journal of Science and Technology,* 8(26), 1-7. http://www.indjst.org/index.php/indjst/article/view/71956 [22.3.17]

[4] Shah, K. (2016). "Single Versus Three Hall Sensor Configuration" (Application Report SLAA695). http://www.ti.com/lit/an/slaa695/slaa695.pdf [23.3.17]

[5] Bhardwaj, M. (2013). "Software Phase Locked Loop Design Using C2000™ Microcontrollers for Three Phase Grid Connected Applications" (Application Report SPRABT4A). http://www.ti.com/lit/an/sprabt4a/sprabt4a.pdf95/slaa695.pdf [23.4.17]

[6] Staebler, M. (2000). "TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed" (Application Report SPRA605). http://www.ti.com/lit/an/spra605/spra605.pdf [25.4.17]

[7] Jung, S.Y. and Nam, K. (2011). "PMSM Control Based on Edge-Field Hall Sensor Signals Through ANF-PLL Processing". In: *IEEE Transactions On Industrial Electronics,* 58(11), 5121-5129. http://ieeexplore.ieee.org/document/5715872/ [22.3.17]

[8] Golestan, S., Guerrero, J. M., and Vasquez, J. C. (2016) "Three-Phase PLLs: A Review of Recent Advances". In: *IEEE Transactions on Power Electronics,* 32(3), 1894-1907. http://vbn.aau.dk/en/publications/threephase-plls-a-review-of-recent-advances(333bf4d7-4efa-4e84-b459-4a3d58f33849).html [4.5.17]

[9] Minambres, V., Milanes, M.I., Vinagre, B. and Romero, E. (2009) "Comparison of controllers for a three-phase Phase Locked Loop system under distorted conditions". In *Compatibility and Power Electronics*, Badajoz, 2009, 79-85. http://ieeexplore.ieee.org/document/5156017/ [28.4.17]

APPENDICES

APPENDIX I. FILE Resolver_Parameters_discrete.m

This appendix introduces the .m file used to provide the Virtual Resolver Parameters. It is presented the sample time, the vehicle characteristics, Low pass filters design, gains, offsets, PLL controller, and more.

```
%% Resolver_Parameters
%%
clc
load NEDC.mat
Sample_Time=1/10000;

%% Vehicle characteristics
%%
rw=0.3239;  % Wheel radius
N=28;       % Number of pole pairs
Tc=0.4;     % Change Constant Time
Kc=62.5000; % Change constant coefficient
Tc2=1;      % Change Constant Time
Kc2=4;      % Change constant coefficient

%% Fisrt Conditioning
%%
FC_Gain=1/1365.33;
FC_Offset1=-2290;
FC_Offset2=-2289;
FC_Offset3=-2267;
FC_Offset4=-2267;
FC_Offset5=-2267;

%% Butterworth Filter - Input Signal
%%
% 10000 rad/sec - Max Input Frequency (considering the 3rd Harmonic)

[NumIn,DenIn]=butter(1,(10000/(2*pi))/(1/(Sample_Time*2)));
[NumSin,DenSin]=butter(1,(2000/(2*pi))/(1/(Sample_Time*2)));
%% ANF Quality
%%
mi2=80; % 1/Quality Factor

%% ANF Phase Controller
%%
NumIn2=[1 -1];
DenIn2=[1 -0.999];
% NumU=[1 0.0001];
% DenU=[1 10];
% U=tf(NumU,DenU);
% c2d(U,Sample_Time);
%% PLL PID
%%
zeta=0.7;   % Damping Factor
tol=0.05;   % Tolerance
Ts=0.02;    % Setling Time
kp=(-2*log(tol*sqrt(1-zeta^2)))/Ts;          % Proportional Gain
ki=((-log(tol*sqrt(1-zeta^2)))/(Ts*zeta))^2; % Integrative Gain
kd=0;                                        % Derivative Gain
wn=sqrt(ki);                                 % Natural Frequency
```

```
%% Adaptive PI Gains
%%
ki2=ki/200;          % Minimum Integrative Gain
% ki2=ki/400;        % Minimum Integrative Gain (Ideal PLL speed input)
wn2=sqrt(ki2);       % Minimun Natural Frequency
kp2=zeta*2*wn2;      % Minimun Proportional Gain
kd2=0;               % Minimun Derivative Gain

%% Butterworth Filters
%%
%% 8 rad/sec (speed estimator filter)
% 1st order
[NumS8,DenS8]=butter(1,(8/(2*pi))/(1/(Sample_Time*2)));
% 3rd order
[NumS,DenS]=butter(3,(8/(2*pi))/(1/(Sample_Time*2)));
%% 50 rad/sec (speed estimator filter)
% 1st order
[NumS50,DenS50]=butter(1,(50/(2*pi))/(1/(Sample_Time*2)));
% 3rd order
[NumS1,DenS1]=butter(3,(50/(2*pi))/(1/(Sample_Time*2)));
%% 100 rad/sec (speed estimator filter)
% 1st order
[NumS100,DenS100]=butter(1,(100/(2*pi))/(1/(Sample_Time*2)));
```

APPENDIX II. AUXILIARIES FILES .m

In this appendix, auxiliaries .m files are presented. Such files were used to generate and calculate the results and graphs that were not plotted by the *Simulink* Scopes.

FFT_test_Hall.m

```
clc
load('Hall_out_test.mat')
load('angleHall.mat')
N=length(h1);
H1=fft(h1,N);
b=(1/0.00005)*pi*(0:N-1)/N;
figure(6)
plot(b,abs(H1)/N,'b','LineWidth',1), grid on,axis([0 2000 0 0.5])
title('FFT of H1 for an electric frequency of 220 rad/s')
xlabel('Frequency [rad/s]')
ylabel('Amplitude')
set(gcf,'color','white')
set(gca, 'FontSize',10,'Fontweight','bold','linewidth',1)
grid on
```

Phase_Mag_Calculation.m

```
clc
mi=80;
w0=0.01:0.001:0.1;
w1=0.1:0.01:1;
w2=1:0.1:10;
w3=10:1:100;
w4=100:10:1000;
w5=1000:100:10000;
w6=10000:1000:100000;
we=[w0 w1 w2 w3 w4 w5];
for i=1:length(we)
    MAGanf(i)=20*log10((8*we(i)^2)/(sqrt(64*we(i)^4+6400*we(i)^2)));
    PHASEanf(i)=-180*atan(10/we(i))/pi;
    MAGc2(i)=20*log10((sqrt(we(i)^2+100))/we(i));
    PHASEc2(i)=180+180*(atan(-we(i)/10))/pi-90;
    magcc(i)=20*log10(sqrt(we(i)^2+100)/we(i));
    phasecc(i)=-180*atan(we(i)/10)/pi+90;
    %     [MAGc(i),PHASEc(i)] = bode((s+0.1)/(s+10),we(i));
end
figure(1)
semilogx(we,PHASEanf)
grid on
figure(2)
semilogx(we,MAGanf)
grid on
```

APPENDIX III. SIMULATION GRAPHS

This appendix shows other simulation results. Its notable a convergence of the estimation error to that one presented on chapter *8*, that was optimized.

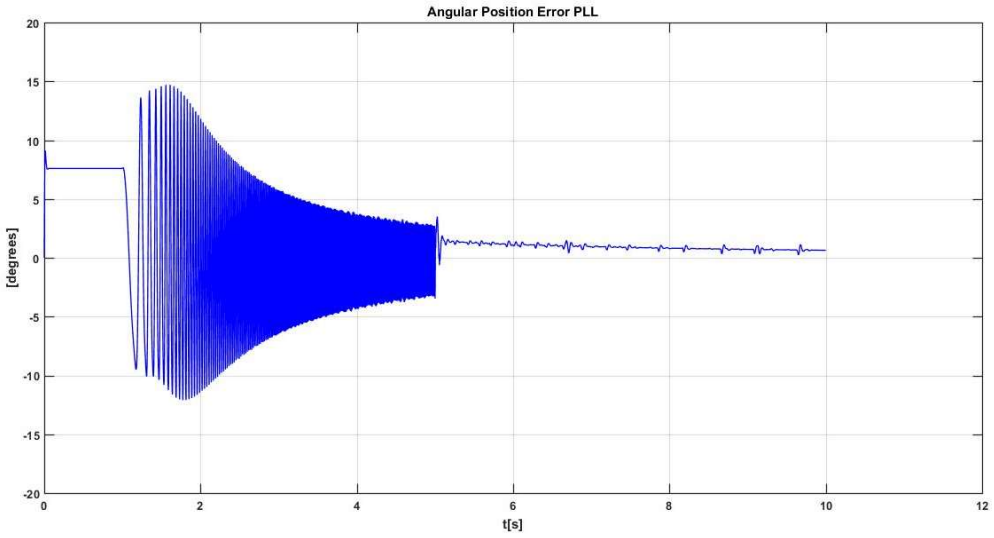

Figure 76 - PLL error estimation with activation of the ANF at 5 seconds

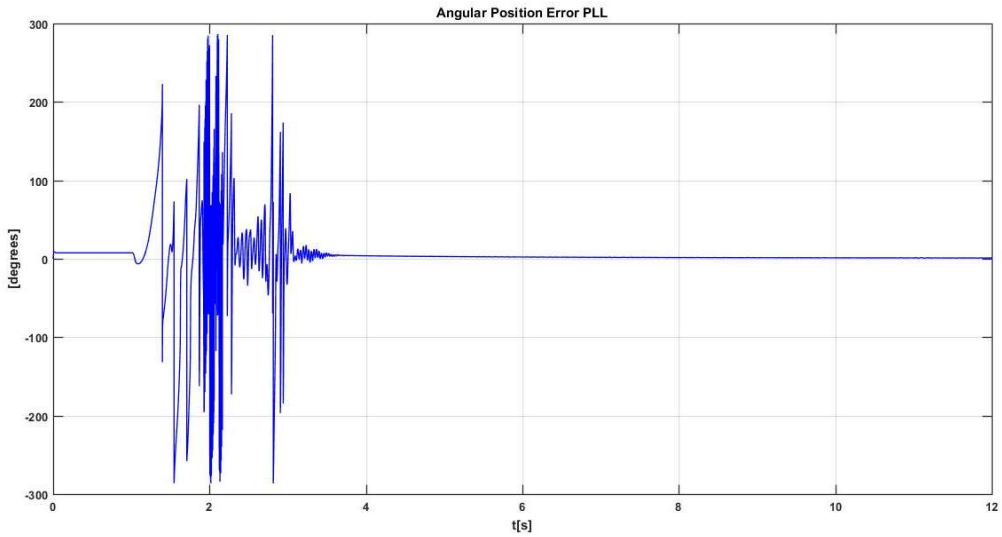

Figure 77 - PLL estimation error with ANF and without phase conpensation

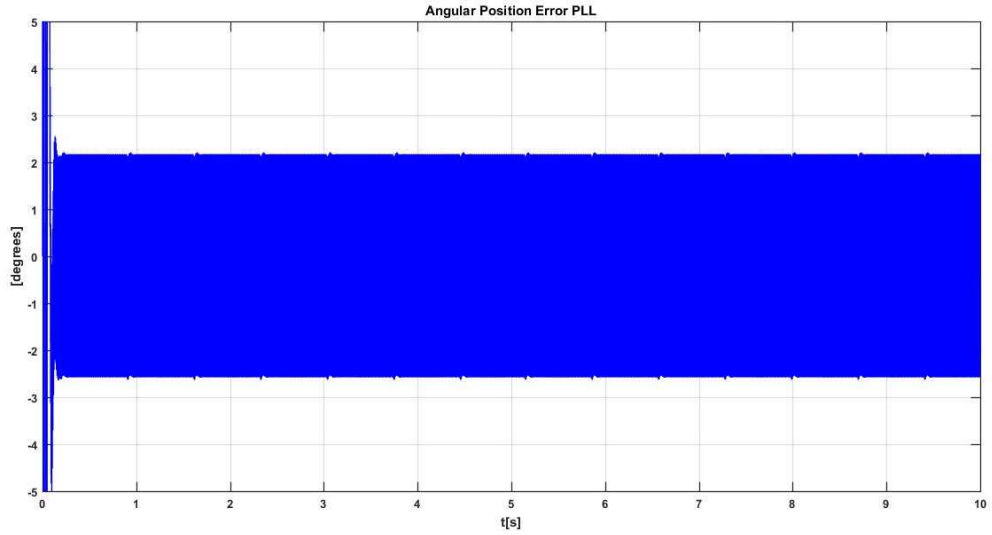

Figure 78 - PLL estimation error at a constant speed input

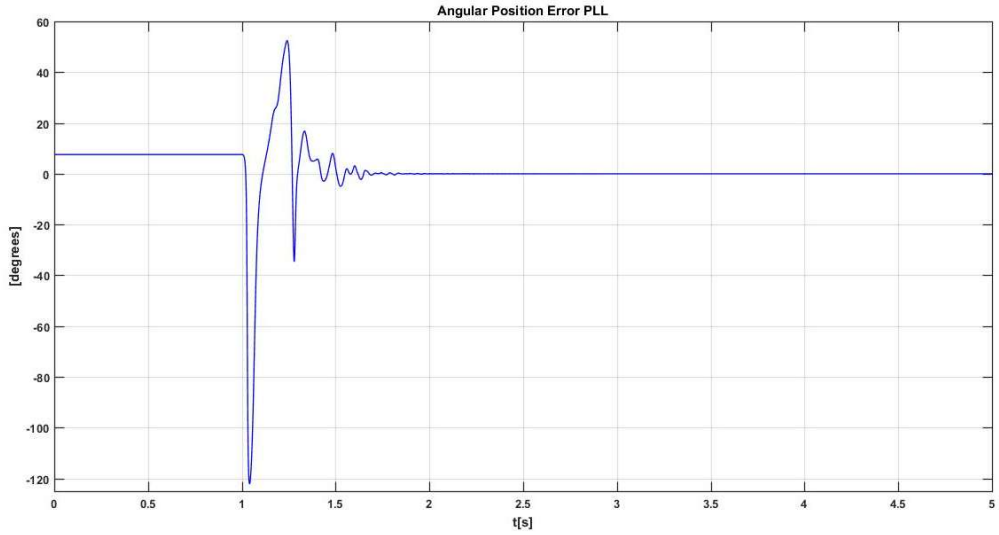

Figure 79 - Arctangent estimation error using reference speed

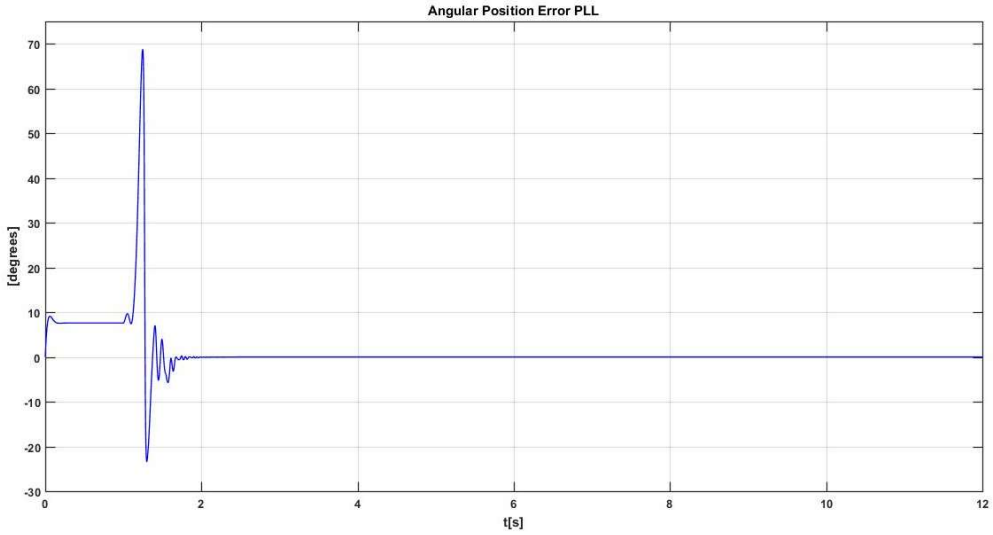

Figure 80 - PLL estimation error using reference speed