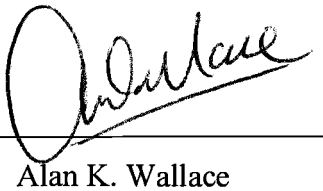
AN ABSTRACT OF THE DISSERTATION OF

Ernesto J. Wiedenbrüg for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on September 24, 1998. Title: Measurement, Analysis and Efficiency Estimation of Three Phase Induction Machines Using Instantaneous Electrical Quantities.

Abstract approved: 

Alan K. Wallace

Estimating the operating efficiency of an induction machine is of prime importance for modern plant management. An extensive testing program of present efficiency estimation techniques was performed at the Motor Systems Resource Facility (MSRF), during which serious shortcomings of the existing techniques were identified.

The objective of the work of this dissertation was to develop and investigate an alternative method of efficiency estimation, which imposes a substantially lower level of intrusion than the previous existing methods, without compromising typically achieved accuracies. A two-axis model of a three-phase induction machine was developed, in which components of current, power and impedance due to rotor asymmetries are investigated. The predicted current sidebands were verified in a laboratory setting and subsequently used for speed prediction of the operating load point. Operating torque was estimated via two axis analysis of the induction machine. With the knowledge of torque and operating speed, output power can be calculated. Input power is measured with the current and voltage sensor array used for output power estimation. Efficiency is obtained as the ratio of output to input power.

A proof of concept laboratory implementation of the suggested method is presented. Seven induction machines were tested on a dynamometer at four load points, verifying predicted accuracies of the implemented technique.

Measurement Analysis and Efficiency Estimation of Three Phase Induction Machines Using Instantaneous Electrical Quantities.

By Ernesto J. Wiedenbrüg

A DISSERTATION

Submitted to

Oregon State University

In partial fulfillment of

the requirements for

the degree of

Doctor of Philosophy

Presented September 24, 1998

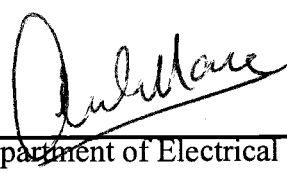
Commencement June 1999

Doctor of Philosophy dissertation of Ernesto J. Wiedenbrüg

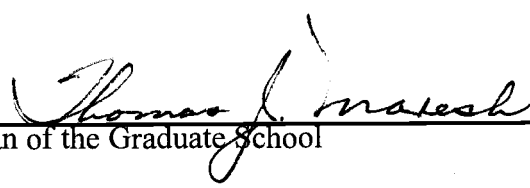
presented on September 24, 1998.

APPROVED:

Major Professor, representing Electrical and Computer Engineering



Head of the Department of Electrical and Computer Engineering



Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Ernesto J. Wiedenbrüg, Author

Table of Contents

Table of Contents (continued)

List of Figures

List of Figures (continued)

List of Tables

Appendix Figures

Appendix Figures (continued)

List of Appendix Tables

Measurement, Analysis and Efficiency Estimation of Three Phase Induction Machines Using Instantaneous Electrical Quantities

1. Introduction

1.1. Relevance of operating efficiencies

Until recent years, the National Electric Manufacturers Association (NEMA) standards on efficiencies of three phase Induction Machines (IM) were mere guidelines for the industry of the United States, generally providing a distinction between normal and high efficiency IMs. More recent legislation, including the National Energy Policy Act of 1992 (EPACT'92), obliges the industry to adopt higher efficiency standards for IMs, leading to the specification and development of "Design E" motors. In addition to legislation conformance, a knowledge of an IM's operating efficiency is of prime importance for cost effective plant management. An IM operating at low efficiency can point at either imminent failure of the machine, permitting preventive maintenance, or at a mismatched machine-load condition. Efficiency testing can also provide a means for 'sanity checking' the nameplate data of IMs.

1.2. Standardized efficiency measurements

In this work it is generally assumed that the IM is in the motoring mode of operation, and it will be stated explicitly if the generating mode is referred to.

The conventional definition of efficiency for all machines and electrical machinery in particular is by equation (1.1):

$$\eta = \frac{P_{out}}{P_{in}} \tag{1.1}$$

where P_{in} is the electrical input power, and P_{out} is the mechanical (shaft) output power of the IM.

However, depending on the situation, there are different numbers of efficiency that are of interest, and which must be clearly distinguished from each other. *Operating* efficiency is defined as the efficiency at which a particular machine is running at a certain point in time. The Institute of Electrical and Electronic Engineers (IEEE) specifies the procedures for measuring steady state operational efficiency in the standard IEEE 112a.

Rated, nameplate or *nominal efficiency* is the efficiency given on the nameplate of the IM. This number is specified by the manufacturer of the machine. It is the result of efficiency tests carried out, usually by the manufacturer's laboratory, according to the IEEE 112b standard on a statistically relevant sample of that particular model and rating. The obtained data is then evaluated, after which the producer specifies the nameplate efficiency of that model.

Conceptually, there is yet an other difference between standards IEEE 112a and IEEE 112b. The former standard, as described above, intends to provide an accurate snapshot of the instantaneous efficiency. For that reason the standard belongs to the class of input-output efficiency testing methods.

Conversely, standard 112b is much more involved and attempts to provide with two sets of information: Firstly, a load curve, specifying currents, power factors, efficiencies, operating temperatures and operating speeds for 25%, 50%, 75%, 100%, and typically 125% and 150% of mechanical loading at rated supply voltage conditions. Secondly, the IEEE 112b standard, being a loss segregation method, splits the occurring losses into different categories:

- Stator I^2R (conductive) losses
- Rotor I^2R (conductive) losses
- Core (hysteresis and eddy current) losses

- Friction and windage (mechanical) losses
- Stray load (other and combined) losses

From a systems point of view, standard IEEE 112a intends to describe the specific operating performance of a particular IM, while standard IEEE 112b attempts to describe the capabilities of a particular IM.

NEMA has introduced yet an other efficiency number, the *minimum* or *guaranteed efficiency*. According to NEMA, the stated nameplate data should represent a mean efficiency of IMs of that particular model, operated at rated conditions. Minimum efficiency represents a lower bound specified in [1] that not one healthy IM of a certain nameplate efficiency rating should fall below when run under rated conditions.

1.3. Approaches for efficiency and load estimation

As stated previously, plant operators and plant managers have good reasons for investigating the operating efficiencies of their IMs. But the standards IEEE 112a, and particularly IEEE 112b represent very intrusive procedures for test methods. Torque measurements and speed measurements are needed even in the simplest implementation of the standardized testing procedures, apart from also requiring more balanced voltage conditions than the ones commonly available at industrial sites. This high level of intrusiveness usually prohibits efficiency testing according to the IEEE specifications, leaving standardized testing to a laboratory environment option only.

Many different efficiency estimation methods have been developed with field suitability as a goal. These estimation techniques also share a lower level of intrusiveness in performing the required measurements, as well as a lesser accuracy in their predicted efficiencies.

A thorough study on efficiency estimation techniques has been performed by J. S. Hsu et al [2], which divides efficiency estimation techniques into the following classes:

- Nameplate method
- Slip method
- Current method
- Statistical method
- Equivalent circuit method
- Segregated loss method
- Air gap torque method
- Shaft torque method

The principles of each of these will now be described in the following sections.

1.3.1 Nameplate method

This method is the most trivial, least intrusive, and in consequence usually also the least accurate method. It is based on the assumption, that the machine will always operate at nominal efficiency, regardless of loading, voltage condition or health condition of the machine. The rated efficiency is obtained from the nameplate. Obviously, it is unlikely that the machine will perform at the nominal efficiency. Load variations, as well as source imbalances and harmonic components change the efficiency. Moreover, the nameplate efficiency is the result of a statistical evaluation of nominally identical machines' efficiencies and does not necessarily coincide with the evaluated machine's nominal efficiency. Another source of possibly large inaccuracies is that the machine could have been rewound, which could have a large impact on the operating efficiency, if the repair was not done properly.

1.3.2 Slip method

Fig. 1.1 shows the typical torque/speed profile for a 50 hp IM

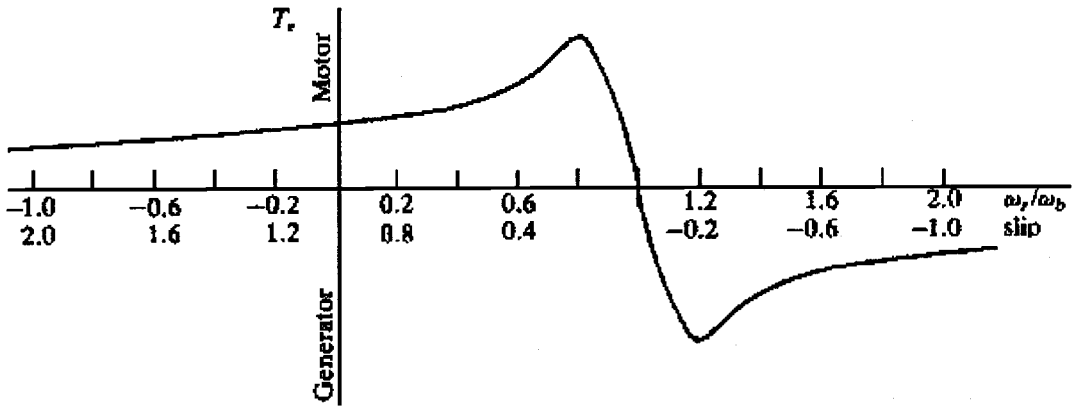


Fig. 1.1: Typical Torque-Speed profile of an IM

Approximating the normal operating range part of torque-speed curve of an IM by a straight line is a common approach in power engineering. This is a valid approach, since the steady state operating point is usually in the 0 to 0.1 slip range, for a machine of good efficiency, for which the profile is close to linear. It is simple to calculate the operating power output from the IM, knowing the rotor speed and the rated slip, which is obtained from the nameplate. The operating efficiency can then easily be calculated by dividing the estimated mechanical output power by the measured electrical input power.

The main disadvantage of this method is that the rated operating speed is seldom an accurate figure. NEMA specifies up to 20% of error for the nameplate slip. This error carries through to the efficiency calculation, which then becomes very unreliable.

Another disadvantage is also given by the fact that possible rotor damage will cause a significant drop in the operating speed for a given torque. Conversely, a drop in speed points to a larger mechanical output, if applying the slip method. This would cause a significant increase in the estimated efficiency figure, opposing what is physically happening and being counter-productive to preventive maintenance measures.

1.3.3 Current method

Another method of obtaining an efficiency estimate is achieved by using a rms current measurement as a scale for the mechanical loading. This approach errs significantly by not taking the perpendicular orientation of the magnetizing current phasor with respect to the torque-producing component into account. It also neglects the influence of I^2R losses in rotor and stator, which are comparatively large in lower horsepower IMs. Figure 1.2 shows from a circle diagram, that the error committed by the misalignment of the complex current with respect to the real load current is substantial.

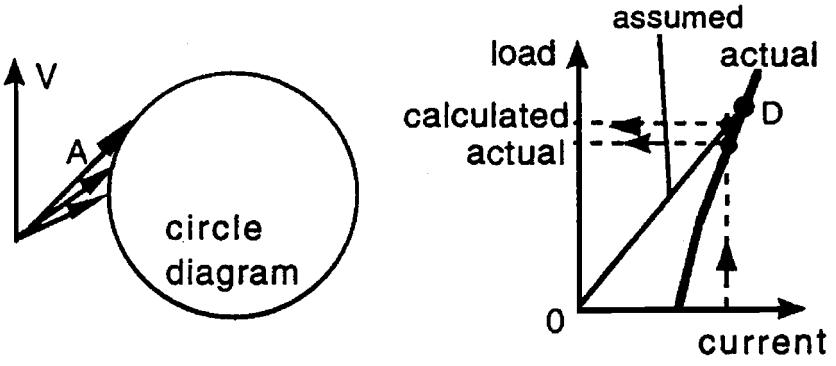


Figure 1.2: Circle diagram for the current efficiency estimation method.

In addition to these definite theoretical errors, there is also the problem of typical nameplate inaccuracy. The error of rated current may be as large as 10% on a healthy IM, according to NEMA.

After adding all the mentioned error sources it is clear that the current method can only roughly estimate the loading of the machine, and consequently cannot lead to reliable efficiency estimations.

1.3.4 Statistical method

Statistical methods combine measured data of IMs with empirical equations derived for groups of motors. An example of such a method is entering the measured stator current, the horsepower and the pole number into a program which has knowledge of typical stator and rotor resistances, friction and windage losses, and also stray load losses. Out of the combination of expected typical losses for a load point derived from the stator current, the program estimates the operating efficiency.

Inaccuracies similar to those identified with the previous methods also apply with this approach, since deviations of nameplate information compared with the actual performance of the machine are to be expected.

Additionally, statistically obtained information is obviously only as valuable, as the investigated machine approaches the knowledge based mean values. This fact causes also this approach to be particularly flawed for predictive maintenance purposes, since deviations of standardized values become non observable with this method.

1.3.5 Equivalent circuit method

Figure 1.3 shows the basic per phase equivalent circuit for an IM.

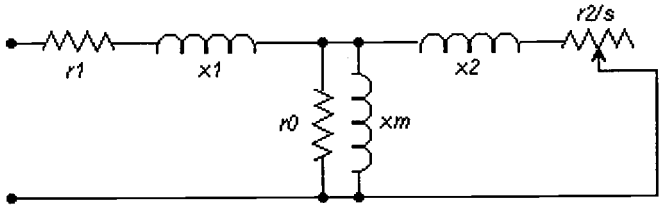


Figure 1.3: Equivalent circuit of an IM.

r_s and r_r respectively represent the stator and equivalent rotor resistances, x_s and x_r the leakage inductances for stator and referred rotor, x_m the mutual or magnetization

inductance, r_0 models the core losses, and s represents the slip of the given operating condition.

Obtaining the values of the equivalent circuit components is done differently depending on the particular implementation. Commonly, an off-line stator resistance measurement, and a no load measurement are performed to obtain the values of r_s and an estimate of r_r. Additional nameplate information, particularly the design letter, is commonly used to obtain an x_s/x_r ratio. Power factor measurements at no load and at a load point are further values commonly used to segregate the inductive circuit components.

This approach has the advantage of predicting the operating efficiency for different load points. The equivalent circuit neglects friction and windage as well as stray load losses, which can be part of an incorporated knowledge base, thus reducing the inaccuracies.

This model has the disadvantages that harmonics and imbalances are not accounted for, and that the friction and windage values must be estimated because they are non observable. Another disadvantage is the relatively high level of intrusion, since the machine requires no load and off-line measurements, additional to the operating load point data. It also assumes that the circuit parameters are constants, which is in error due to thermal and flux level effects in the IM.

1.3.6 Segregated loss method

This method is similar to the equivalent circuit method and is based on the equation (1.2).

$$\eta = 1 - \frac{P_{loss}}{P_{in}} \tag{1.2}$$

and attempts to characterize the total power loss P_{loss} component by component, and measure P_{in} to obtain η. The stator loss is estimated by utilizing the off-line measured

stator resistance r_s, and calculating $I^2 \cdot r_s$. Core losses, friction and windage as well as stray load losses are commonly estimated in a knowledge base. The rotor resistance is usually obtained by a no load measurement. The patent from Vogelsang and Benning is an implementation of this method [3].

The segregated loss approach has similar advantages as the equivalent circuit method, but does not share the advantage of predicting the performance of the IM for loading other than the current operating point.

1.3.7 Airgap torque method

This method utilizes equation (1.3), which has been utilized widely for torque control of IMs.

$$T_e = \frac{3}{2}\frac{P}{2}\left(\lambda_{ds} \cdot i_{qs} - \lambda_{qs} \cdot i_{ds}\right) \tag{1.3}$$

where $\lambda_{d,q\ s}$ represents the flux in direct and quadrature axis in the stator reference frame, and $i_{d,q\ s}$ represents the stator currents in the direct and quadrature axis, respectively. T_e is the airgap torque, which equals output plus friction and windage, and stray load loss torque. P is the pole number of the IM. The flux is commonly obtained by integrating the stator voltages after the stator resistive voltage drop, which means that the stator resistances have to be measured off-line, or they have to be estimated by different means.

In order to estimate the efficiency with this method it is also necessary to measure the speed, in order to obtain the mechanical output power using

$$P_{mech} = 2 \cdot \pi \cdot \omega_r \cdot T_e \tag{1.4}$$

Current and voltage need to be measured to obtain T_e, which makes P_{in} available for the η calculation.

This method has the advantages that it has a relatively low level of intrusiveness and a high accuracy, as proven many times in the drive industry. The disadvantages lie in the requirement of r_s measurement or estimation, and that friction and windage, as well as stray load losses and core losses have to be estimated as well, leading to expected inaccuracies.

1.3.8 Shaft torque method

This method is probably the most accurate, since it measures directly the electrical input power and the mechanical output power. The latter is measured by a torque transducer, which has to be mounted solidly on the shaft of the IM between motor and load. The serious disadvantage of this method is the extremely high level of intrusiveness it encompasses. It is extremely uncommon that an IM will be operated with a calibrated torque transducer in an industrial application. Retrofitting a torque transducer in a plant means serious downtime, with large losses of productivity. Additionally, accuracies of torque transducers are relatively low, if they are not properly calibrated, for which laboratory setups may be necessary. Torque transducers are also known to be relatively weak mechanically, and frequent startup processes may damage them, or make re-calibrations necessary.

Undoubtedly, shaft torque measurements are the most accurate for small and medium sized machinery, loosing accuracy for larger sized machines. But the extreme level of intrusion caused by mounting the transducers make this method an option only in the rarest occasions at an industrial site. Additionally, larger IMs commonly operate with efficiencies above 95%. This causes the required accuracies of mechanical torque measurement to be so high that they can commonly be only granted in laboratory environments.

2. Speed Estimation

2.1 Proposed current signature method

The goal of estimating the efficiency of an IM causing the least possible intrusion, while maintaining a high accuracy, leads to investigations on possible ways to estimate the operating speed of the IM. In an industrial environment this is desirable as speed estimation removes the need of connecting speed sensors, for which accessibility and physical proximity to the shaft of the machine under test is required. Numerous sources are available which investigate possible ways to extract the operating speed from stator current signatures [4].

The rotor and stator slot counts of the investigated IM, although known to the designer and manufacturer, are commonly not readily available at industrial sites. Thus the use of slot harmonics for speed prediction is not generally viable. Nonetheless, there are other components in the current spectra, like the ones caused by rotor eccentricities, which can be used to observe the speed from the electrical terminals of the machine.

The operating efficiency of IMs decays rapidly as the size of their airgaps increases. This causes IMs to be designed and built with the airgap bounded by realizable manufacturing tolerances. The resulting airgaps are not of a constant size around the rotor, but have inherent inaccuracies. These inaccuracies are commonly called eccentricities, and can be divided into static and dynamic forms. Static eccentricities can be caused by a misalignment of the rotor rotational and symmetry axis with respect to the bore axis, or by out-of-round components of the stator bore. Dynamic eccentricities are those components which either describe the misalignment of a round rotor's symmetry axis with respect to its own rotational axis, or the components that describe the out-of-roundness of the rotor.

These four cases of eccentricity are represented schematically in Fig. 2.1. The out of round eccentricities that are shown in Fig. 2.1. are simpler than will be encountered in real life, since the circles and ellipses are mathematical shapes, not possible to manufacture with perfect accuracy.

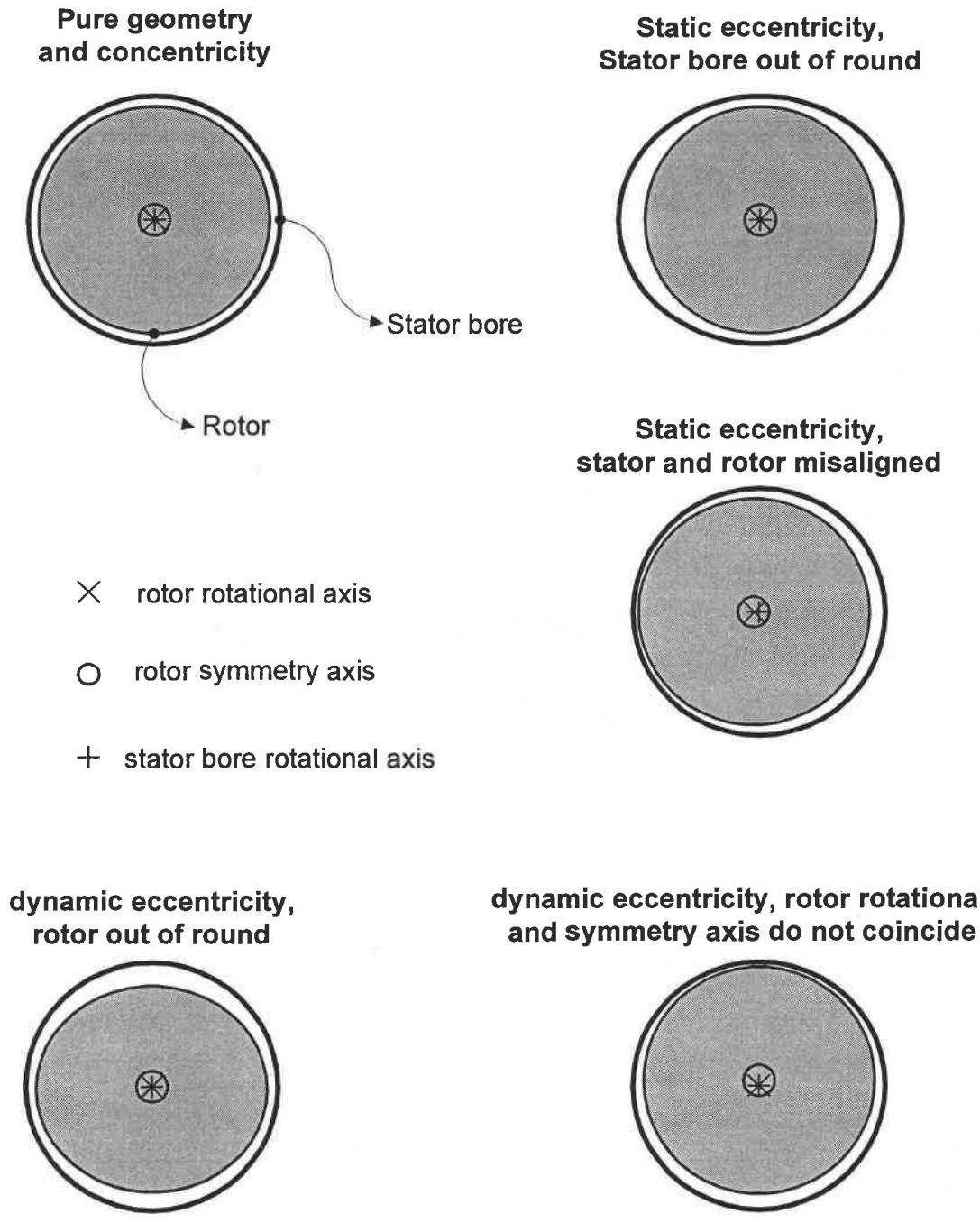


Figure 2.1: Static and dynamic eccentricity cases

A physical machine will have much more complex deviations from a circle than the ones represented by an ellipse. In practice all forms of eccentricity can be present in a practical machine.

The airgap in the "out-of-round" cases is the resulting difference between the circular and elliptic shapes. This can also be formulated as two concentric circles, one of which has its radius modulated by a sinusoid of low amplitude and of a rotational frequency equal to twice the angle in polar coordinates. As such it represents a "mild" form of saliency. With this type of description it becomes possible to reduce dynamic eccentricities to one case only.

2.2 Model of dynamic eccentricities

An infinite series of spatial Fourier components describes the irregular rotor or stator geometries with respect to their axes. In consequence, the only difference between the static and dynamic eccentricities becomes that the latter Fourier series is stationary in the stator reference frame, while the former rotates with rotor speed.

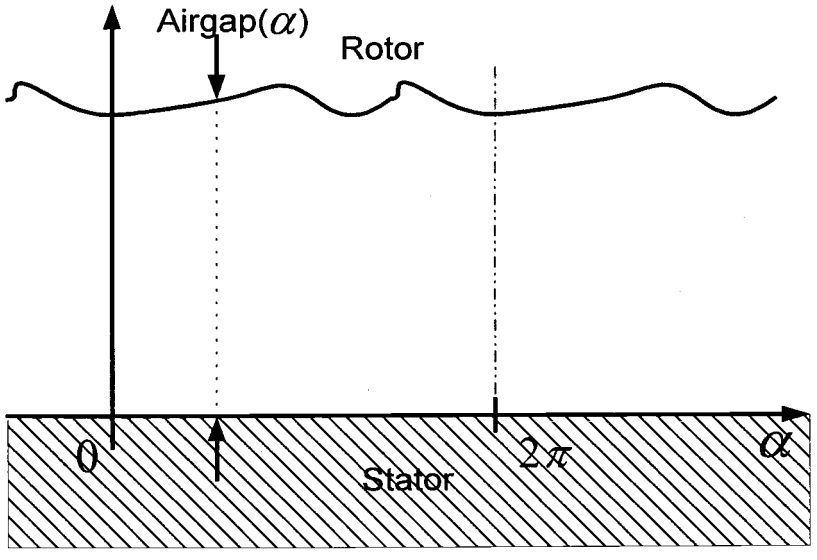


Figure 2.2: Linearized representation of rotor eccentricity.

This work will focus on dynamic eccentricity, which means that the stator is assumed to be a perfect circle, hence only the rotor imperfections are analyzed. Such a case is schematically shown in Fig. 2.2 in linear coordinates, where the airgap of the machine is shown as the difference from the perfectly flat stator to the imperfect rotor. It is obviously possible to expand the presented analysis by including static eccentricities. This can be achieved by creating a flat reference line, parallel to the developed rotor rotational axis line, and then describing deviations from that line by a static infinite Fourier series for the stator eccentricities. Adding another infinite Fourier Series, modulated by the rotor speed, would describe both forms of eccentricities.

Fig. 2.2 shows the pure rotor eccentricity case, where the angle around the stator is α, represented by the x-axis. The rotor structure obviously repeats itself every 2π.

The repetition of the airgap length every 2π is the reason why the Fourier series decomposition was chosen over other series decomposition methods. The airgap length is, by definition, always positive and unequal to zero, otherwise the rotor would contact the stator laminations. Thus it can be described in two different ways utilizing Fourier decomposition; the standard form shown in Eq. 2.1.a, and its inverse representation in Eq. 2.1.b:

$$airgap(\alpha,t) = \sum_{j=0}^{\infty} F_j \cdot \cos\left(j \cdot \alpha + j \cdot \varphi_j + j \cdot \omega_r \cdot t\right) \qquad (2.1.a)$$

$$airgap(\alpha,t) = \sum_{i=0}^{\infty} \frac{const}{F_i \cdot \cos\left(i \cdot \alpha + i \cdot \varphi_i + i \cdot \omega_r \cdot t\right)} \qquad (2.1.b)$$

In equation (2.1) ω_r represents the rotational speed of the rotor, and φ_i and φ_j the geometrical displacement of the i-th and j-th Fourier component. (2.1.b) is chosen for the further analysis, since it simplifies the superposition in the final model.

The variation of the airgap as a function of time and rotational speed cause the spectral components in the stator currents. These components can be analyzed and used for very accurate speed extraction.

The mutual inductance of stator and rotor is inversly proportional to the airgap.

$$L_m = \frac{k_m}{airgap} \tag{2.2}$$

Figure 2.3. shows schematically the influence of the varying airgap size on both the mutual and the leakage inductances.

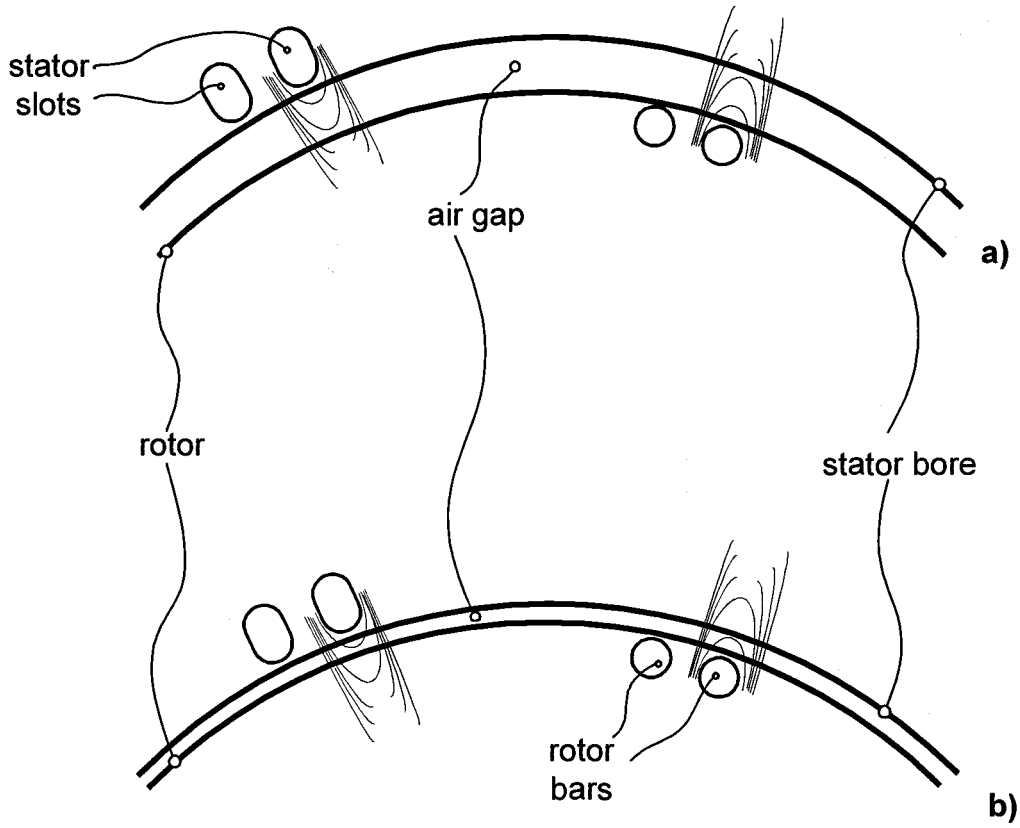


Fig. 2.3.: Mutual inductance and leakage inductances for large (a) and small (b) airgap sizes.

It becomes apparent that, while the mutual inductance decreases with increasing airgap sizes, the effect is opposite for the leakage inductances, particularly for such components as 'tooth-top' or 'zigzag' leakage. Lacking a more precise knowledge of the function between airgap size and leakage inductance, a linear relationship will be assumed.

$$L_l = k_l \cdot airgap \tag{2.3}$$

The relative permeability of non-saturated iron is assumed to be much higher than the relative permeability of the airgap. Skin effect, hystheresis and saturation effects are neglected as well. These simplifications allow viewing the iron paths of the IM as perfect magnetic short circuits, as a result of which the analysis can be solely focused on the airgap irregularities.

Applying Fourier decompositions, the inductances of phase *a* of the machine can be written as

$$L_{lsa} = L_{ls} \cdot \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + i \cdot \varphi_i\right)} \tag{2.4.a}$$

$$L_{lra} = L_{lr} \cdot \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + i \cdot \varphi_i\right)} \tag{2.4.b}$$

$$L_{ma} = L_m \cdot \sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + i \cdot \varphi_i\right) \tag{2.4.c}$$

where L_m, L_{ls} and L_{lr} are constants representing the median mutual, stator leakage and rotor leakage inductances, respectively.

Allowing for a polepair number P, (2.4) is more accurately expressed as

$$L_{lsa}(t) = \frac{L_{ls}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + \frac{u \cdot i \cdot 2\pi}{P}\right)} \tag{2.5.a}$$

$$L_{lra}(t) = \frac{L_{rs}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + \frac{u \cdot i \cdot 2\pi}{P}\right)} \tag{2.5.b}$$

$$L_{ma}(t) = \frac{L_m}{P} \cdot \sum_{u=1}^{P} \sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + \frac{u \cdot i \cdot 2\pi}{P}\right) \tag{2.5.b}$$

because the influence of the i-th Fourier component is different facing every pole due to the different location around the stator bore. This effect is addressed by the added superposition over the variable u and the second term in the cosine. Similarly, the effect of the airgap facing phases b and c must be calculated from the known coefficients F_i, φ_i and their geometrical displacement by 120° and 240° with respect to phase a.

$$L_{lsb}(t) = \frac{L_{ls}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} - i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right)}$$

$$L_{lsc}(t) = \frac{L_{ls}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right)} \qquad (2.6.a)$$

$$L_{lrb}(t) = \frac{L_{rs}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} - i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right)}$$

$$L_{lrc}(t) = \frac{L_{rs}}{P} \cdot \sum_{u=1}^{P} \frac{1}{\sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right)} \qquad (2.6.b)$$

$$L_{mb}(t) = \frac{L_m}{P} \cdot \sum_{u=0}^{P} \sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} - i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right)$$

$$L_{mc}(t) = \frac{L_m}{P} \cdot \sum_{u=0}^{P} \sum_{i=0}^{\infty} F_i \cdot \cos\left(i \cdot \omega_r \cdot t + \frac{i \cdot \varphi_i}{P} + i \cdot \frac{2\pi}{3} + \frac{u \cdot i \cdot 2\pi}{P} \right) \qquad (2.6.c)$$

Obviously, L_{ma}, L_{mb} and L_{mc} are functions of t. Only for perfect rotor and stator geometries would L_{ma} equal L_{mb} and L_{mc} for each t.

When describing the machine mutual inductances in matrix form, the three mutual inductances $L_{ma}(t)$, $L_{mb}(t)$ and $L_{mc}(t)$ are the diagonal elements of the inductance matrix. The off-diagonal elements represent the mutual couplings of the phases with respect to each other. These mutual inductances are described by

$$
\begin{aligned}
M_{ab}(t) &= M_{ba}(t) = k_{mph}\sqrt{L_{ma}(t)\cdot L_{mb}(t)} \\
M_{ac}(t) &= M_{ca}(t) = k_{mph}\sqrt{L_{ma}(t)\cdot L_{mc}(t)} \\
M_{bc}(t) &= M_{cb}(t) = k_{mph}\sqrt{L_{mb}(t)\cdot L_{mc}(t)}
\end{aligned}
\tag{2.7}
$$

The factor k_{mph} is a constant and represents the coupling coefficient of the different phases. The k_{mph} coupling coefficient is a function of the phase spread τ of the windings. k_{mph} is a measure of how much of the flux created by winding t coincides, including polarity, with winding u. If winding t and u shared the same phase spread, with position and turns ratio, while neglecting slot leakage, then the coupling factor would be 1. If winding t and winding u were physically orthogonal, then the mutual coupling becomes zero, which is the cause of decoupling of the d and q components when analyzing a symmetric machine in dq domain.

Two examples of stators with the pole pair number of one are drawn in Figure 2.4.. In the first case, the phase spread of $\tau = 60°$ causes no overlapping of adjacent windings. In the second case, with a phase spread of $\tau = 80°$, an overlap of $20°$ is formed. Note, however, that for the most common form of winding, two layer, the winding distributions are sinusoidal and the overlapping is total, with an overlapping of $60°$.

For calculating k_m, the concept of winding density distribution (*wdd*) is introduced. Each phase has a winding density distribution, which helps describing the physical winding distribution. If each phase has the same number of turns, then the maximum of the winding density distributions of every phase is set to one. The winding density distribution represents the mmf and hence the flux as a function of electrical degrees created for a positive current of the same magnitude through each winding. k_{mab} results from the integration of the multiplication of the winding density distributions of phases a and b, and dividing the result by π.

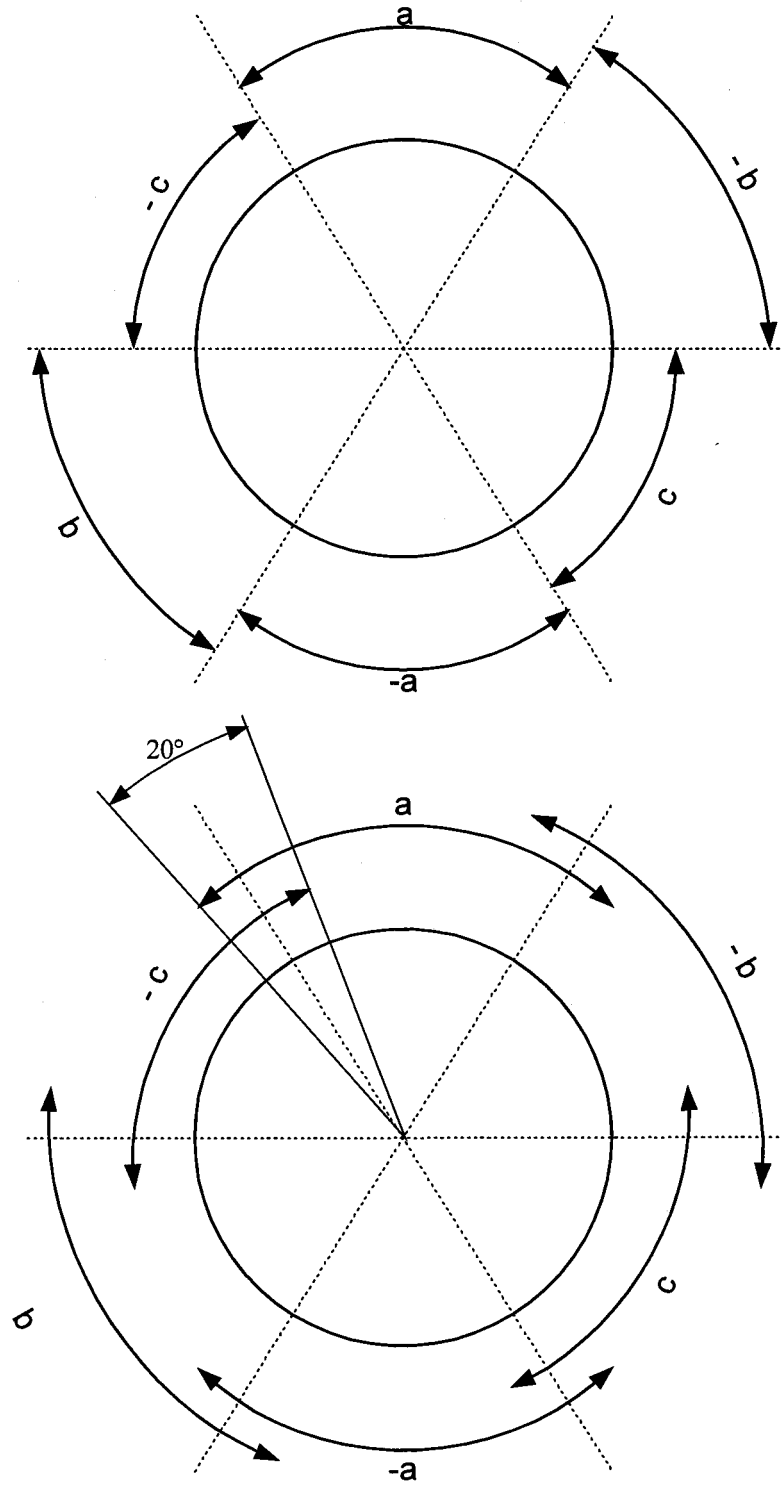


Figure 2.4.a: Coupling coefficients k_m.

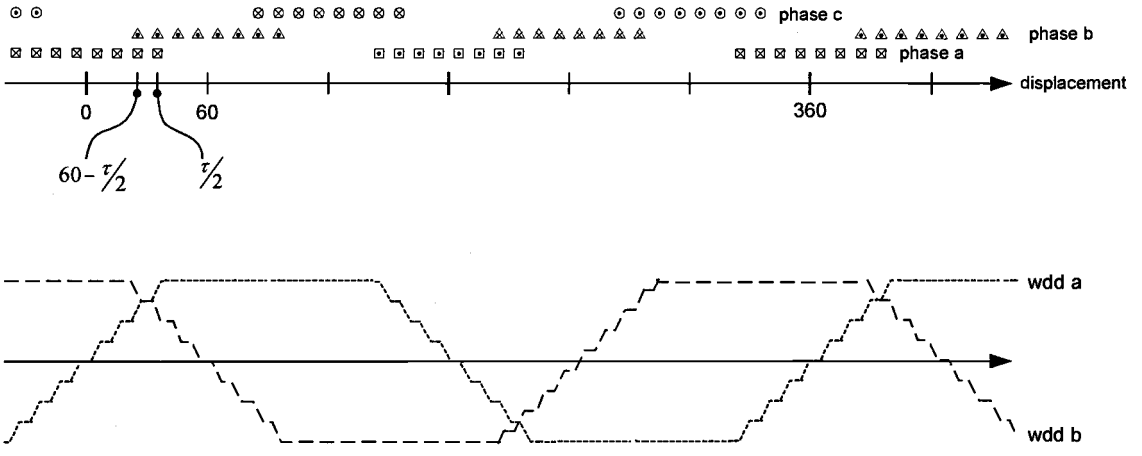


Figure 2.4.b: Stator winding distribution and winding density distributions.

Fig. 2.4. shows an example of winding density distributions for a phase spread of 80◦. The upper part of the figure shows the stator winding distribution for the three phases, while the lower part of the figure shows the winding density distributions for phases a and b. With this k_{mab} calculates as follows:

$$k_{mab} = \frac{1}{180°} \int_0^{180°} (wdda \cdot wddb) \cdot d\alpha \qquad (2.8.a)$$

$$turnslinea = \begin{cases} \dfrac{\alpha}{\tau/2} & 0 \le \alpha \le \tau/2 \\ 1 & \tau/2 \le \alpha \le 180° - \tau/2 \\ 1 - \dfrac{\alpha - (180° - \tau/2)}{\tau/2} & 180° - \tau/2 \le \alpha \le 180° \end{cases} \qquad (2.8.b)$$

$$turnslineb = \begin{cases} 1 & 0 \le \alpha \le 60° - \tau/2 \\ 1 - \dfrac{\alpha - (60° - \tau/2)}{\tau/2} & 60° - \tau/2 \le \alpha \le 60° + \tau/2 \\ -1 + \dfrac{\alpha - (240° - \tau/2)}{\tau/2} & 60° + \tau/2 \le \alpha \le 180° \end{cases} \qquad (2.8.c)$$

For a symmetrically distributed winding, $k_{mab} = k_{mbc} = k_{mca} = k_m$.

The two-axis theory allows to reduce a three phase (*abc*) machine to a two phase (*qd*) machine, with an additional *0* (zero) sequence component, current of which is zero, unless ground faults occur. Protection relays monitor ground currents closely for operator safety reasons. As soon as there is a ground fault current, the breakers will open virtually immediately. Hence there is no reason to include the *0* sequence component of the *dq* model for the current steady state investigation. The transformation of currents and voltages from the *abc* domain to the *qd0* domain can be performed with the following set of equations.

$$\begin{pmatrix} f_q \\ f_d \\ f_0 \end{pmatrix} = [K_s] \begin{pmatrix} f_a \\ f_b \\ f_c \end{pmatrix} \tag{2.9}$$

$$[K_s] = \frac{2}{3} \begin{pmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ ½ & ½ & ½ \end{pmatrix} \tag{2.10}$$

$$[K_s]^{-1} = \begin{pmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{pmatrix} \tag{2.11}$$

The impedance matrix has to be transformed into the *qd0* reference frame as well with the use of

$$\left(Z_{qd0}\right) = \left(K_s\right)\left(Z_{abc}\right)\left(K_s\right)^{-1} \tag{2.12}$$

Since the present analysis is performed in the stator reference frame, θ is zero and the *0* component is not evaluated for previously mentioned reasons. The resulting (L_{mqd}) which solely describes the stator-rotor and stator-stator inductances is

$$\left(L_{mqd}\right)=\begin{pmatrix} L_{m11} & L_{m12} \\ L_{m21} & L_{m22} \end{pmatrix} \tag{2.13}$$

$$\begin{aligned} L_{m11} &= \frac{1}{3}\left(2L_{ma} + \tfrac{1}{2}L_{mb} + \tfrac{1}{2}L_{mc} - 2M_{ab} - 2M_{ac} + M_{bc}\right) \\ L_{m22} &= \frac{1}{2}\left(L_{mb} + L_{mc} - 2M_{bc}\right) \\ L_{m12} &= \frac{1}{2\sqrt{3}}\left(L_{mb} - L_{mc} + 2M_{ac} - 2M_{ab}\right) \\ L_{m21} &= \frac{1}{2\sqrt{3}}\left(L_{mb} - L_{mc} + 2M_{ac} - 2M_{ab}\right) \end{aligned} \tag{2.14}$$

If the IM were symmetrical, which means that the three L_m were equal and the three M were equal, then $L_{m11} = L_{m22} = L_m - M$ and $L_{m12} = L_{m21} = 0$, which coincides with the symmetrical *qd* machine components. However, due to the eccentricities the off-diagonal elements of Z_{abc} are unequal. Consequently, *0* sequence components are present and all elements are time-varying in the investigated real world case. This makes it necessary to add some components to the commonly used *qd* machine model for non-symmetric machines, presented in [5] pp. 430.

The proposed non-symmetric model of an IM, according to Krause, can be modeled using the following set of equations:

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} r_s + p\cdot L_{ss} & 0 & p\cdot L_{ms} & 0 \\ 0 & r_S + p\cdot L_{SS} & 0 & p\cdot L_{mS} \\ p\cdot L_{ms} & -\frac{1}{n}\omega_r L_{ms} & r_r + p\cdot L_{rr} & -\frac{1}{n}\omega_r L_{RR} \\ n\omega_r L_{ms} & p\cdot L_{mS} & n\omega_r L_{rr} & r_R + p\cdot L_{RR} \end{bmatrix} \cdot \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \tag{2.15}$$

$$
\begin{aligned}
L_{ss} &= L_{ls} + L_{ms} \\
L_{SS} &= L_{lS} + L_{mS} \\
L_{rr} &= L_{lr} + L_{ms} \\
L_{RR} &= L_{lR} + L_{mS}
\end{aligned}
\tag{2.16}
$$

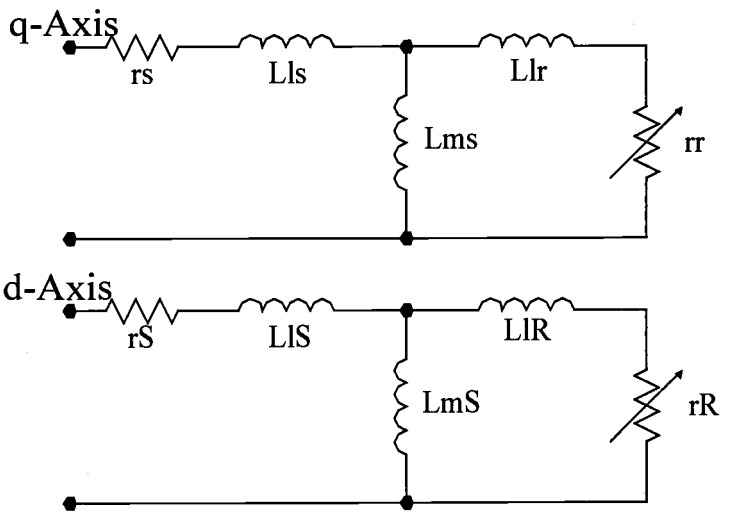


Figure 2.5: Non symmetric machine equivalent circuit.

$r_s = r_S$ and $r_r = r_R$ are fair assumptions, since only a variation in the airgap as a function of rotor position is evaluated. The turns ratio from stator to rotor n is assumed to be equal to *1* for matters of simplicity. The analysis is focused on the current spectra, and the relative amplitude of the peaks with respect to each other, and not on the absolute value of the rotor current.

The values for the stator and rotor leakage inductances in *dq* components are obtained using (2.12) and dropping the zero component:

$$
\left(L_s\right) = \left(K_s\right) \cdot \begin{pmatrix} L_{lsa} & 0 & 0 \\ 0 & L_{lsb} & 0 \\ 0 & 0 & L_{lsc} \end{pmatrix} \cdot \left(K_s\right)^{-1} \tag{2.17.a}
$$

$$(L_r)=(K_s)\cdot\begin{pmatrix} L_{lra} & 0 & 0 \\ 0 & L_{lrb} & 0 \\ 0 & 0 & L_{lrc} \end{pmatrix}\cdot(K_s)^{-1} \tag{2.17.b}$$

$$\begin{aligned} L_{s11} &= \tfrac{1}{3}\cdot\left(2L_{lsa}+\tfrac{1}{2}L_{lsb}+\tfrac{1}{2}L_{lsc}\right) & L_{r11} &= \tfrac{1}{3}\cdot\left(2L_{lra}+\tfrac{1}{2}L_{lrb}+\tfrac{1}{2}L_{lrc}\right) \\ L_{s22} &= \tfrac{1}{2}\cdot\left(L_{lsb}+L_{lsc}\right) & L_{r22} &= \tfrac{1}{2}\cdot\left(L_{lrb}+L_{lrc}\right) \\ L_{s12} &= \tfrac{1}{2\sqrt{3}}\cdot\left(L_{lsb}-L_{lsc}\right) & L_{r12} &= \tfrac{1}{2\sqrt{3}}\cdot\left(L_{lrb}-L_{lrc}\right) \\ L_{s21} &= \tfrac{1}{2\sqrt{3}}\cdot\left(L_{lsb}-L_{lsc}\right) & L_{r21} &= \tfrac{1}{2\sqrt{3}}\cdot\left(L_{lrb}-L_{lrc}\right) \end{aligned} \tag{2.17.c}$$

Furthermore, as mentioned previously, the off-diagonal elements of the L_{mqd} matrix are not equal zero, with which equation (2.14) has to be changed to

$$\begin{aligned} (v) &= (Z)\cdot(i) \\ (v) &= \begin{bmatrix} v_{qs} \\ v_{ds} \\ 0 \\ 0 \end{bmatrix} \\ (i) &= \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \\ [Z] &= \begin{bmatrix} r_s+p\cdot\left(L_{m11}+L_{s11}\right) & p\cdot\left(L_{m12}+L_{s12}\right) & p\cdot L_{m11} & p\cdot L_{m12} \\ p\cdot\left(L_{m21}+L_{s21}\right) & r_s+p\cdot\left(L_{m22}+L_{s22}\right) & p\cdot L_{m21} & p\cdot L_{m22} \\ p\cdot L_{m11} & p\cdot L_{m12}-\omega_r L_{m11} & r_r+p\cdot\left(L_{m11}+L_{r11}\right) & p\cdot\left(L_{m12}+L_{r12}\right)-\omega_r\left(L_{m22}+L_{r22}\right) \\ p\cdot L_{m21}+\omega_r L_{m11} & p\cdot L_{m22} & p\cdot\left(L_{m21}+L_{r21}\right)+\omega_r\left(L_{m11}+L_{r11}\right) & r_r+p\cdot\left(L_{m22}+L_{r22}\right) \end{bmatrix} \end{aligned} \tag{2.18}$$

This system of equations can be drawn in the equivalent circuit of Fig. 2.6.

The time variance of the self inductances, mutual inductances and currents make it necessary to pay closer attention to the differential terms of equation (2.18).

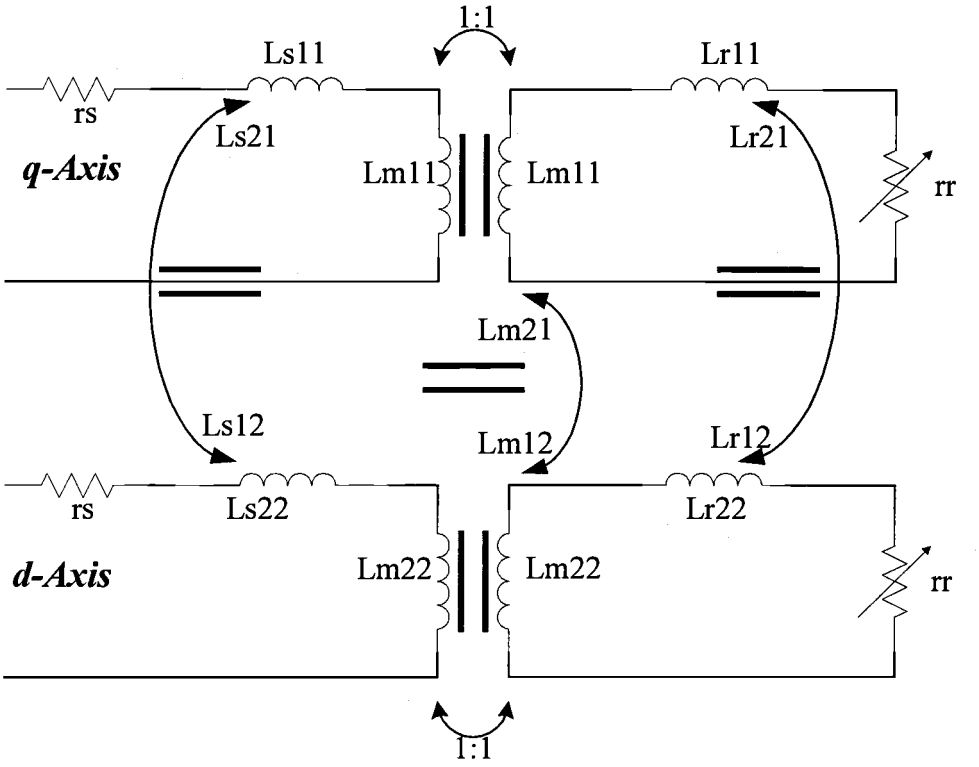


Figure 2.6: Equivalent circuit for airgap irregularities.

$$p\cdot\left(\left(L_{m11}+L_{s11}\right)\cdot i_{qs}\right)=\left(L_{m11}+L_{s11}\right)\cdot p\cdot i_{qs}+i_{qs}\cdot\left(pL_{m11}+pL_{s11}\right) \tag{2.19.a}$$

$$p\cdot\left(\left(L_{m12}+L_{s12}\right)\cdot i_{ds}\right)=\left(L_{m12}+L_{s12}\right)\cdot p\cdot i_{ds}+i_{ds}\cdot\left(pL_{m12}+pL_{s12}\right) \tag{2.19.b}$$

$$p\cdot L_{m11}\cdot i_{qr}=L_{m11}\cdot p\cdot i_{qr}+i_{qr}\cdot pL_{m11} \tag{2.19.c}$$

$$p\cdot L_{m12}\cdot i_{dr}=L_{m12}\cdot p\cdot i_{dr}+i_{dr}\cdot pL_{m12} \tag{2.19.d}$$

$$p\cdot\left(\left(L_{m21}+L_{s21}\right)\cdot i_{qs}\right)=\left(L_{m21}+L_{s21}\right)\cdot p\cdot i_{qs}+i_{qs}\cdot\left(pL_{m21}+pL_{s21}\right) \tag{2.19.e}$$

$$p\cdot\left(\left(L_{m22}+L_{s22}\right)\cdot i_{ds}\right)=\left(L_{m22}+L_{s22}\right)\cdot p\cdot i_{ds}+i_{ds}\cdot\left(pL_{m22}+pL_{s22}\right) \tag{2.19.f}$$

$$p\cdot L_{m21}\cdot i_{qr}=L_{m21}\cdot p\cdot i_{qr}+i_{qr}\cdot pL_{m21} \tag{2.19.g}$$

$$p\cdot L_{m22}\cdot i_{dr}=L_{m22}\cdot p\cdot i_{dr}+i_{dr}\cdot pL_{m22} \tag{2.19.h}$$

$$p\cdot L_{m11}\cdot i_{qs}=L_{m11}\cdot p\cdot i_{qs}+i_{qs}\cdot pL_{m11} \tag{2.19.i}$$

$$p\cdot L_{m12}\cdot i_{ds}=L_{m12}\cdot p\cdot i_{ds}+i_{ds}\cdot pL_{m12} \tag{2.19.j}$$

$$p\cdot\left(\left(L_{m11}+L_{s11}\right)\cdot i_{qr}\right)=\left(L_{m11}+L_{s11}\right)\cdot p\cdot i_{qr}+i_{qr}\cdot\left(pL_{m11}+pL_{s11}\right) \tag{2.19.k}$$

$$p\cdot\left(\left(L_{m12}+L_{s12}\right)\cdot i_{dr}\right)=\left(L_{m12}+L_{s12}\right)\cdot p\cdot i_{dr}+i_{dr}\cdot\left(pL_{m12}+pL_{s12}\right) \tag{2.19.l}$$

$$p\cdot L_{m21}\cdot i_{qs}=L_{m21}\cdot p\cdot i_{qs}+i_{qs}\cdot pL_{m21} \tag{2.19.m}$$

$$p\cdot L_{m22}\cdot i_{ds}=L_{m22}\cdot p\cdot i_{ds}+i_{ds}\cdot pL_{m22} \tag{2.19.n}$$

$$p\cdot\left(\left(L_{m21}+L_{s21}\right)\cdot i_{qr}\right)=\left(L_{m21}+L_{s21}\right)\cdot p\cdot i_{qr}+i_{qr}\cdot\left(pL_{m21}+pL_{s21}\right) \tag{2.19.o}$$

$$p\cdot\left(\left(L_{m22}+L_{s22}\right)\cdot i_{dr}\right)=\left(L_{m22}+L_{s22}\right)\cdot p\cdot i_{dr}+i_{dr}\cdot\left(pL_{m22}+pL_{s22}\right) \tag{2.19.p}$$

The terms pL_{m11}, pL_{m12}, pL_{m21}, pL_{m22}, pL_{s11}, pL_{s12}, pL_{s21}, pL_{s22}, pL_{r11}, pL_{r12}, pL_{r21} and pL_{r22} are calculated analytically, forming state independent time varying values. L_{m11}, L_{m12}, L_{m21} and L_{m22} are linear combinations of L_{ma}, L_{mb}, L_{mc}, M_{ab}, M_{bc} and M_{ac}. Similarly pL_{s11}, pL_{s12}, pL_{s21} and pL_{s22} are functions of L_{sa}, L_{sb} and L_{sc}, as are pL_{r11}, pL_{r12}, pL_{r21} and pL_{r22} functions of L_{ra}, L_{rb} and L_{rc}.

For this reason calculating pL_{ma}, pL_{mb}, pL_{mc}, pM_{ab}, pM_{bc}, pM_{ac}, pL_{sa}, pL_{sb}, pL_{sc}, pL_{ra}, pL_{rb} and pL_{rc} is necessary. Matrix *(Z)* holds linear combinations of these derivatives, which can be substituted by the derivatives of equations (2.14) and (2.17.c).

$$pL_{lsa}(t)=\frac{L_{ls}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.a}$$

$$pL_{lsb}(t)=\frac{L_{ls}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t-\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t-\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.b}$$

$$pL_{lsc}(t)=\frac{L_{ls}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t+\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t+\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.c}$$

$$pL_{lra}(t)=\frac{L_{lr}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.d}$$

$$pL_{lrb}(t)=\frac{L_{lr}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t-\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t-\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.e}$$

$$pL_{lrc}(t)=\frac{L_{lr}}{P}\sum_{u=1}^{P}\frac{\sum_{i=0}^{\infty}F_i\cdot i\cdot\omega_r\cdot\sin\left(i\cdot\omega_r\cdot t+\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)}{\left(\sum_{i=0}^{\infty}F_i\cdot\cos\left(i\cdot\omega_r\cdot t+\frac{i\cdot 2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right)\right)^2} \tag{2.20.f}$$

$$pL_{ma}(t)=-\frac{L_m}{P}\sum_{u=1}^{P}\sum_{i=0}^{\infty}F_i\cdot i\cdot\sin\left(i\cdot\omega_r\cdot t+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right) \tag{2.20.g}$$

$$pL_{mb}(t)=-\frac{L_m}{P}\sum_{u=1}^{P}\sum_{i=0}^{\infty}F_i\cdot i\cdot\sin\left(i\cdot\omega_r\cdot t-i\cdot\frac{2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right) \tag{2.20.h}$$

$$pL_{mc}(t)=-\frac{L_m}{P}\sum_{u=1}^{P}\sum_{i=0}^{\infty}F_i\cdot i\cdot\sin\left(i\cdot\omega_r\cdot t+i\cdot\frac{2\pi}{3}+\frac{i\cdot\varphi_i}{P}+\frac{u\cdot i\cdot 2\pi}{P}\right) \tag{2.20.i}$$

$$pM_{ab}=p\left(-k_{mph}\cdot\sqrt{L_{ma}\cdot L_{mb}}\right)=-\frac{k_{mph}}{2}\cdot\frac{L_{ma}\cdot pL_{mb}+L_{mb}\cdot pL_{ma}}{\sqrt{L_{ma}\cdot L_{mb}}} \tag{2.20.j}$$

$$pM_{ac}=p\left(-k_{mph}\cdot\sqrt{L_{ma}\cdot L_{mc}}\right)=-\frac{k_{mph}}{2}\cdot\frac{L_{ma}\cdot pL_{mc}+L_{mc}\cdot pL_{ma}}{\sqrt{L_{ma}\cdot L_{mc}}} \tag{2.20.k}$$

$$pM_{bc}=p\left(-k_{mph}\cdot\sqrt{L_{mc}\cdot L_{mb}}\right)=-\frac{k_{mph}}{2}\cdot\frac{L_{mc}\cdot pL_{mb}+L_{mb}\cdot pL_{mc}}{\sqrt{L_{mc}\cdot L_{mb}}} \tag{2.20.l}$$

The equations (2.19) and (2.20) permit rewriting (2.18) into the following simplified form.

$$v=[A]\cdot\dot{x}+[B]\cdot x \tag{2.21.a}$$

$$v = \begin{bmatrix} v_{qs} \\ v_{ds} \\ 0 \\ 0 \end{bmatrix} \tag{2.21.b}$$

$$x = \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{dr} \end{bmatrix} \tag{2.21.c}$$

$$[A] = \begin{bmatrix} L_{m11} + L_{s11} & L_{m12} + L_{s12} & L_{m11} & L_{m12} \\ L_{m21} + L_{s21} & L_{m22} + L_{s22} & L_{m21} & L_{m22} \\ L_{m11} & L_{m12} & L_{m11} + L_{r11} & L_{m12} + L_{r12} \\ L_{m21} & L_{m22} & L_{m21} + L_{r21} & L_{m22} + L_{r22} \end{bmatrix} \tag{2.21.d}$$

$$[B] = \begin{bmatrix} r_s + pL_{m11} + pL_{s11} & pL_{m12} + pL_{s12} & pL_{m11} & pL_{m12} \\ pL_{m21} + pL_{s21} & r_s + pL_{m22} + pL_{s22} & pL_{m21} & pL_{m22} \\ pL_{m11} & pL_{m12} - \omega_r L_{m11} & r_r + pL_{m11} + pL_{r11} & pL_{m12} + pL_{r12} - \omega_r (L_{m22} + L_{r22}) \\ pL_{m21} + \omega_r L_{m11} & pL_{m22} & pL_{m21} + pL_{r21} + \omega_r (L_{m11} + L_{r11}) & r_r + pL_{m22} + pL_{r22} \end{bmatrix} \tag{2.21.e}$$

The system of equations will be simulated using the ODE suite, which is a part of Matlab. The ODE suite requires the set of differential equations to obey

$$\dot{x} = E \cdot x + u \tag{2.22}$$

This form is obtained by rearranging (2.21) to

$$\dot{x} = [A]^{-1} \cdot (-[B] \cdot x + v) \tag{2.23}$$

2.3 Results of simulation

The model that has been presented was in section 2.2 implemented successfully in Matlab, and the Matlab code is given seen in Appendix B. The simulation has been run for a 4-pole machine, with the parameters equal to the 50hp machine of page 190 in [5], with $r_s = 0.087\Omega$, $r_r = 0.228\Omega$, $X_{ls} = X_{lr} = 0.302\Omega$ and $X_M = 13.08\Omega$. The simulation

shows currents that are close to sinusoidal, which is to be expected for an operating IM. But the currents are not purely sinusoidal, which is intuitive looking at the modulation of the inductances, which even occurs with a frequency different from a multiple of 60Hz. For that reason it is of interest to examine the non sinusoidal components, which can be done best in the frequency domain, employing Matlab's Fast Fourier Transform on the calculated currents.

The imperfections that have been modeled first are only of first order, which describes a misalignment of a perfectly round rotor with it's rotational axis. The severity of this model was of $\pm 3\%$ with respect of a mean airgap length of l. The first graphs shown here are from the currents I_{ds} and I_{qs}. The program was set to simulate 30s of operation, for which took roughly 10Mflops.

Since the time steps that were requested from the Matlab ODE34 modeling suite are of 0.001s, and the available FFT function does not offer a time step input, the resulting magnitudes of the components are not displayed in physical Amps. The time domain spectra given in Figs. 2.7, 2.8, 2.9.a and 2.9.b are all presented to the same logarithmic scale. Consequently, the magnitudes are valid for comparison purposes of their relative size.

The significance of this particular extract of the frequency domain is, that the location of the peak can be used to calculate the operating steady state speed (29.1262Hz in this case). This peak in the frequency domain is located at synchronous speed minus rotor speed. Similarly, the side band located to the right of the fundamental as seen in Figure 2.7., is located at synchronous speed plus rotor speed. Obviously, the peaks for phase d are located in the same positions, since the rotor irregularities are rotating and will be facing phase d 90° later.

Using the currents and the voltages it becomes possible to generate a variety of electrical and mechanical instantaneous signals, like the current phasor, real and imaginary power, complex impedance and airgap torque. All these quantities are of instantaneous nature, which creates their distinct signatures.

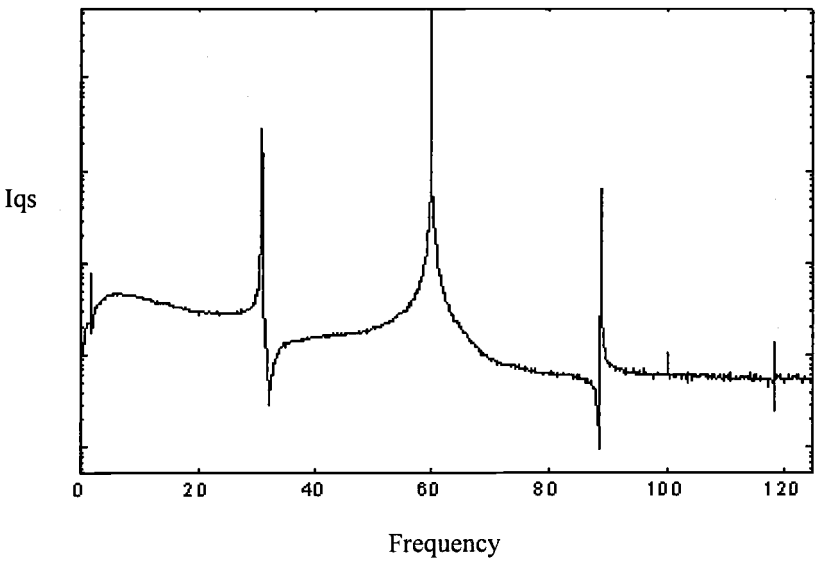


Figure 2.7: I_{qs} in the frequency domain, 0-125Hz.

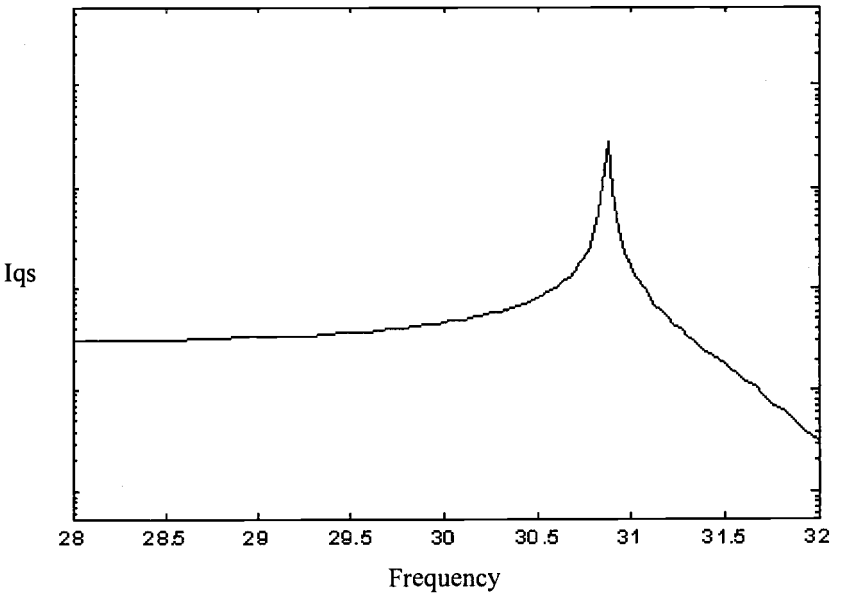


Figure 2.8: I_{qs} in the frequency domain, 28-32Hz.

The current phasor is defined in the *abc* as well as in the *dq* domain. Here its amplitude was calculated as

$$I = \sqrt{i_d^{\,2} + i_q^{\,2}} \tag{2.24}$$

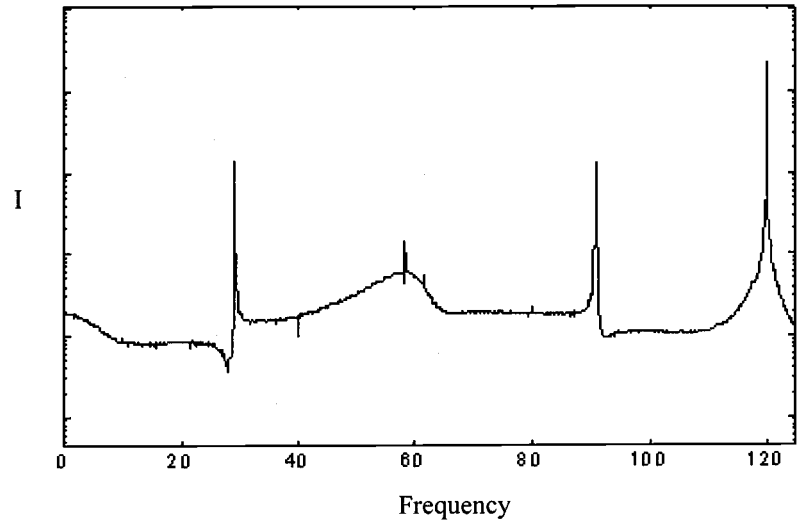


Figure 2.9.a: Frequency content of the current phasor, 0-125Hz.

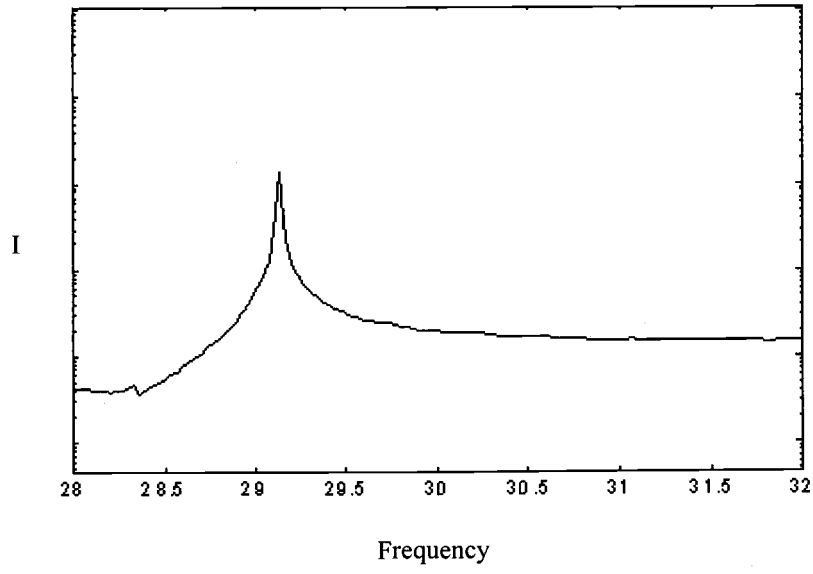


Figure 2.9.b: Frequency content of the current phasor, 28-32Hz.

As in Fig. 2.7., Fig. 2.9.b. shows that the speed information is present in the current phasor signal as well. In this case, rather than being located at synchronous speed minus rotor speed, it is located at rotor speed. The second side band can be found at 60Hz plus rotor speed.

Similarly, real and imaginary power are defined in the *dq* domain as

$$\begin{aligned} p_q &= i_q \cdot v_q \\ p_d &= i_d \cdot v_d \\ p &= p_q + p_d \end{aligned} \qquad (2.25.a)$$

$$\begin{aligned} q_d &= v_d \cdot i_q \\ q_q &= -v_q \cdot i_d \\ q &= q_d + q_q \end{aligned} \qquad (2.25.b)$$

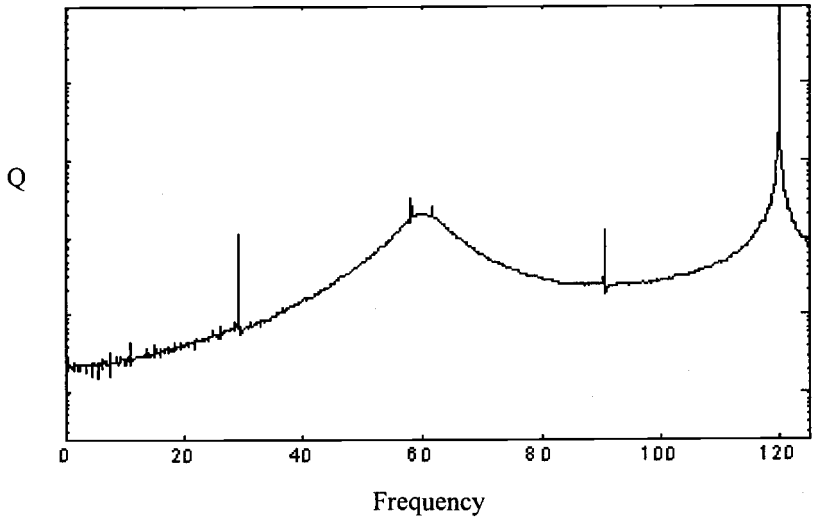


Figure 2.10.a: Frequency content of the imaginary power, 0-125Hz.

Once more the representation in Figure 2.10.b. shows that the rotor speed can be obtained by looking at the side bands of an instantaneous signal, in this case the instantaneous imaginary power. In this case the speed equals the location of the peak.

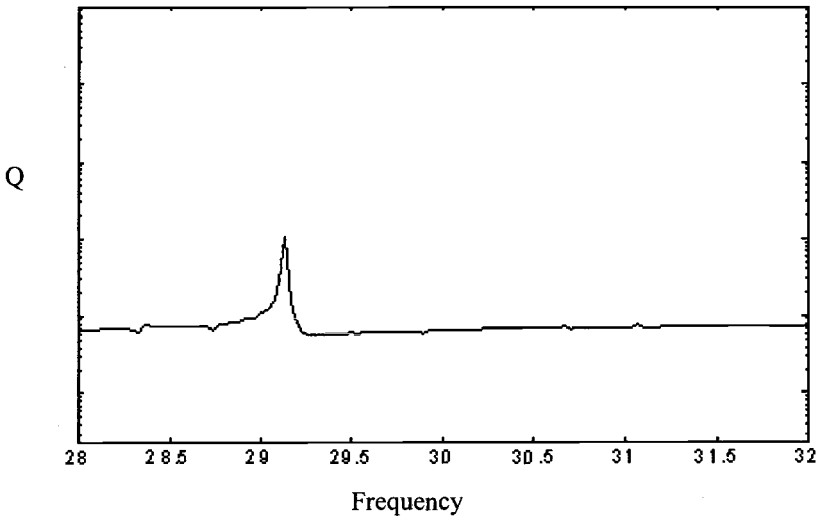


Figure 2.10.b: Frequency content of the imaginary power, 28-32Hz.

Similar results are obtained for instantaneous real powers, absolute values of the instantaneous impedance, instantaneous phase angle between the current and the voltage phasor, and the calculated (or with analog circuitry generated) instantaneous torque signal.

2.4 Digital frequency lock loop filter

2.4.1 Limits of the Spectrum function

The frequency spectra that were shown in section 2.3 have been generated from the simulated current using the FFT function, which is part of the Matlab environment. In laboratory tests it is necessary to perform similar analyses on practical data obtained by measurement. The experimental system runs with LabWindows/CVI as the software environment. This software packet also permits the calculation of the frequency components of a data string by using the Spectrum function. The implementation of the Spectrum function includes a sample rate input, with which the relative sizes of the

current components are intact (other than with the FFT function in the Matlab environment, where the amplitude scale changed). The algorithm which is employed, however, has restrictions in terms of the resolution and the maximum frequency that can be extracted.

The maximal frequency that can be extracted from an acquired data string with a constant sample rate of *rate*, is bounded by the Nyquist criterion [6] to

$$f_{\max} = \frac{rate}{2} \tag{2.26}$$

The maximal resolution Δf in the frequency domain is limited by the algorithm which has been employed, the Spectrum algorithm, to:

$$\Delta f = \frac{1}{T} \tag{2.27}$$

where T is the total time of data acquisition, equal to the number of samples acquired times the sample rate. This limit is imposed by the algorithm itself, and is not created by a fundamental law established in signal processing theory.

Since the prototype analysis system that is developed in this work seeks simplicity, ease of use and accuracy, a closer look has to be given to the constraints imposed by the mentioned algorithm. If the user would want to estimate the operating speed of an IM within ± 1 *rpm*, then the side band of the signal signatures has to be obtained within a sixtieth of a Hertz. Obviously, an error of a sixtieth of a Hertz translates into a *1rpm* error. While using the Spectrum algorithm of LabWindows/CVI, the minimum data acquisition time required, while maintaining the mentioned accuracy of ± 1 *rpm*, translates to at least one minute of data acquisition (2.27). During this period of time the operating point of the machine should not vary significantly; otherwise the speed prediction would be seriously compromised.

Since the limitations are imposed by the algorithm that was employed, and not by laws of physics, other methods of accurately finding the location of the side bands were

researched and developed. These methods intend to minimize length of the data acquisition period, while finding the side bands with an increased analytical effort.

2.4.2 Digital frequency locked loop

The first method that was analyzed was a cross correlation based approach. A family of numerically generated sine waves, varying in frequency, were cross correlated with the acquired data. The frequencies that were generated lay within the 'window of interest', the 4 Hertz window from 28Hz to 32Hz for a 4-pole machine. The result of the cross correlation was plotted against the frequency of the generated sine waves. The plot that was generated shows how much of each particular frequency exists in the acquired signal. The resulting plot resembles closely the results obtained by the Spectrum function. Wavelet analysis, a modern digital signal processing technique, employs a similar methodology.

Wavelet analysis usually also multiplies a window function (commonly Hanning, Hamming, Flattop or Gaussian) times the generated family of sine waves. This forms shapes like Fig. 2.11, where the wave is a sine wave, with an envelope window function.

The advantage of including the window function is that it simplifies non linearities caused by a varying inner product. Additionally, it avoids the requirement of FFT and Spectrum functions, which imply that the signal is of infinite duration in time. The wavelet has certain frequency components, while bundling its energy to a short duration in time. It is very important to point out, that it is not a requirement in wavelet theory to have a sinusoidal component. For the presented analysis, the sinusoidal component, however, proves very useful, since it can be used to filter out a certain frequency from the spectrum.

An other difference between modern wavelet theory and the classical fft and Spectrum analysis resides in the fact that the latter bases on orthogonal functions (a set of sine waves of varying frequency), while the former can handle sets of linearly independent functions of varying shapes (wavelets).

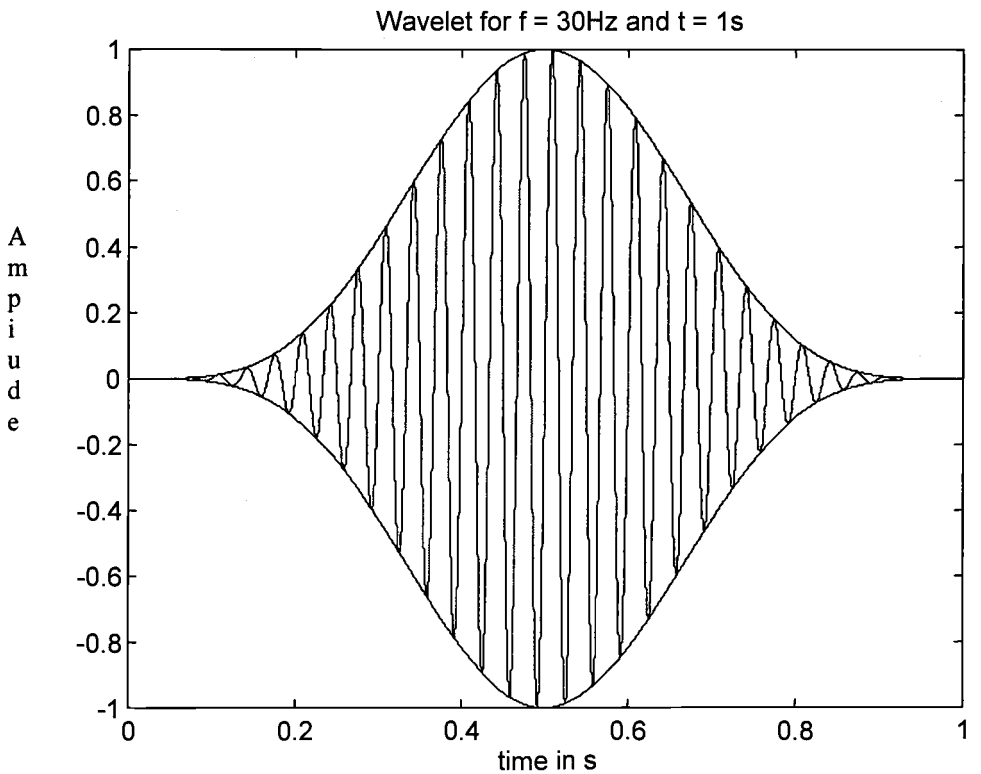


Figure 2.11: Wavelet for 30Hz, with 1s of duration.

A stronghold of wavelet theory is that it allows joint time-frequency analysis (JTFA). This approach enables observation of the frequency domain, while simultaneously seeing the time variations occurring there. This is achieved by creating the inner product of the family of wavelets, using inner frequency f_i as the parameter that filters out the frequency. With this variable, a family of wavelets is created with (2.28)

$$\varphi_i(t) = w(t) \cdot \sin(2\pi \cdot f_i \cdot t) \tag{2.28.a}$$

$$w(z) = \tfrac{1}{2}\left(1 - \cos(2\pi \cdot f_i \cdot t)\right)\Big|_{z=0}^{z=1} \tag{2.28.b}$$

In (2.28), φ_i is the wavelet function that has been employed, and w is the window function of this case. The variable z is defined proportional to time, and ranging from *0*

to *1* for the width of the window. In Fig. 2.11., the width of the window function was set to 1 second, with which $z=t$ for this case only.

According to [7], the next step is to decompose the acquired signal *f(t)* into a set of wavelets, which can be linearly superimposed to generate back the signal (2.29)

$$f(t) = \sum_i a_i \cdot \varphi_i \tag{2.29}$$

At this point, no other requisite is made to φ_i other than that the set is complete, which means that any signal *f(t)* could be reconstructed of the set of φ_i. Since the following approach is a very simplified approach of wavelet analysis, there is no need to deepen further into orthogonal sets of φ_i, minimum bases and similar issues.

While the set of φ is complete, reconstruction of *f(t)* is possible, which means that all information of *f(t)* is available in the φ domain. As stated, one of the strengths of wavelet analysis is that it is the best known transformation aiding JTFA. Commonly, for every t, a set of inner products is made between *f(t)* and φ, according to

$$A(t,i) = \sum \varphi_i(t-k) \cdot f(t) \tag{2.30}$$

The variable k is a time-shift, which has an analogous function to the τ time-variable in the convolution integral. The set of A represents how big of a φ_i component is present in the signal f at the time t, for this particular choice of φ_i.

Looking at equation (2.30) it is clear why φ_i ought to be time-limited, since the inner product is only different from zero for φ_i unequal zero. The main disadvantage of this scheme is that it is too computational intensive to be implemented by on-line solutions. On the other hand, the need for JTFA is not really the reason for the search for other mechanisms of spectrum analysis. Avoiding the time-shifting, which is required in (2.30), would decrease the computational effort to the largest extent.

The building of an inner product according to (2.30) shows how much of a certain frequency is present in a signal *f(t)*, while that signal does not have a phase shift

component. Hence it is possible, using vector algebra approach, to calculate how much of a singular frequency component is present in a signal, by calculating the inner product of that signal with a sine and with a cosine function, as shown in (2.31).

Two families of signals are numerically created: a sine and a cosine function. Each one of these functions is then multiplied, point by point, with the acquired data string, and the results of the multiplications are added, simulating the following equations:

$$c = \int_{t=0}^{t=T} f(t) \cdot \cos(2\pi \cdot f_i \cdot t) \cdot w(t) \cdot dt \tag{2.31.a}$$

$$s = \int_{t=0}^{t=T} f(t) \cdot \sin(2\pi \cdot f_i \cdot t) \cdot w(t) \cdot dt \tag{2.31.b}$$

$$A_i = \sqrt{s_i^2 + c_i^2} \tag{2.31.c}$$

Again, the use of the window function w permits the limitation of T to comparatively small times, without imposing restrictions on the inner products. The variable i is varied, as stated, over the frequencies of interest. The amplitudes A_i reflect the amount of the particular frequency f_i that is present. If there is a distinct peak present within the window of interest, then it shows clearly in the plot of A_i versus the frequencies f_i. Supposing that the *signal(t)* contains a peak at f_k, and other than that peak the rest of the signal is comparatively small, then (2.31) becomes

$$c = \int_{t=0}^{t=T} B_i \sin(2\pi \cdot f_k \cdot t + \varphi) \cdot \cos(2\pi \cdot f_i \cdot t) \cdot w(t) \cdot dt \tag{2.32.a}$$

$$s = \int_{t=0}^{t=T} B_i \cdot \sin(2\pi \cdot f_k \cdot t + \varphi) \cdot \sin(2\pi \cdot f_i \cdot t) \cdot w(t) \cdot dt \tag{2.32.b}$$

$$A_i = \sqrt{s_i^2 + c_i^2} \tag{2.32.c}$$

Obviously, s and c show the orthogonal components of the signal, which accomodates for the phase shift φ, which is an approach commonly seen in phasor calculations.

These equations have been implemented, using the Matlab script shown in Appendix F.

This very simple Matlab script shows the implementation of the digital filter locked loop strategy. The resulting plot, generated with the particular variables of the script in Appendix F, is shown in Figure 2.12.

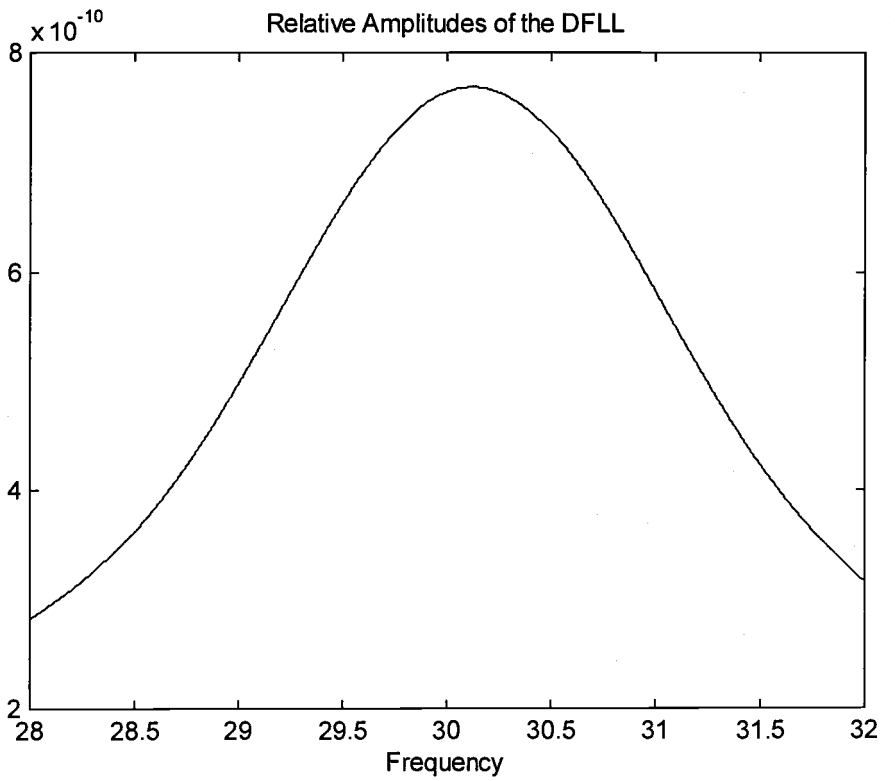


Figure 2.12: Result of the DFLL simulation.

The plot of Fig. 2.12. does not show, however, what accuracy can be achieved with the DFLL as implemented. The calculated maximal value of the result vector lay at the frequency of 30.124Hz, which has an error of only 0.001Hz.

The results obtained with the DFLL algorithm are compared with the results that would have been obtained with the fft algorithm employed by LabWindows/CVI. For only 10s of data acquisition, an accuracy of 0.1Hz could have been achieved, with which it is clear that the wavelet approach can be used to substantially reduce acquisition time.

3. Torque Calculation

3.1 Equation:

Torque calculations for IM have been used for decades for drive applications. The dq theory supplies with a set of equations, with which it is possible to calculate the torque on the rotor of an operating IM. The same set of equations has also been utilized successfully in the drive industry for providing torque control of IMs.

According to [5], pp. 178, the air gap torque T_e equation in the stationary reference-frame variables translates to

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)\left(\psi'_{qs} i'_{ds} - \psi'_{ds} i'_{qs}\right) \tag{3.1}$$

where P is the pole number, $\psi_{qs,ds}$ are the linked fluxes for the q and d axis respectively for the stationary reference frame. $i_{ds,qs}$ represent the qd currents, also for the stationary reference frame

The linked fluxes ψ can be calculated as the integral of the voltages over time, with

$$\psi_{q,d}(t) = \int v_{q,d}(t) \cdot dt \tag{3.2}$$

The stator voltages are readily available, and can be obtained in the dq reference frame using the set of transformations presented in Chapter 2. But the required voltages for the stator referenced airgap torque is not equal to the stator voltages, as seen in the equivalent circuit of Fig. 3.1.. Following the current path from the terminals, there is a voltage drop, caused by the stator resistance of the windings, R_s. This voltage drop causes the stator flux linkage ψ_s to be smaller than if it was calculated right at the terminals, where the erroneous value of ψ_t would have been obtained. The stator flux oriented equation (3.1) requires the use of ψ_s. The difference between ψ_s and ψ_t can clearly seen in Fig. 3.1.

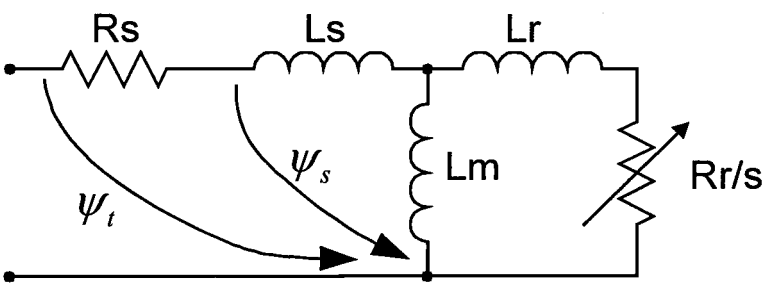


Figure 3.1: Single phase equivalent circuit of IM showing stator flux linkage.

A minimum amount of disruption is intended in the proposed solution of this work for an efficiency estimator, which means that having to stop the IM from operating for off-line measurements is avoided. This means that it is not a requirement to connect a $\mu\Omega$-meter to the terminals of the machine for obtaining the stator resistance value accurately. For this reason an estimation is performed, which is based on typical values of R_s in pu basis.

3.2 Stator resistance estimation:

Smaller motors usually have an R_s of 0.1pu, while large machines typically have an R_s of 0.01pu. A sliding scale is implemented based on a similar knowledge based scale, which uses the user input of rated power of the machine as a guide. The program also contains the option of entering the stator resistance by hand in case it is known.

Obviously, errors committed with this assumption will translate into errors on the torque calculations. Nevertheless it has been consistently observed in the implementation of the torque equation in the practical setup (by varying the inputted stator resistance for the same set of acquired data), that the errors resulting from this coarse estimation have a very small effect looking at the calculated torque.

The reason for this is the following: Looking at the main current path in the equivalent circuit of Fig. 3.1., one sees that it circulates in the outer mesh of the equivalent circuit. It is known that the value of R_r is commonly of the same order of magnitude as R_s. After dividing R_r by s, the result is a very large equivalent load resistance when compared with R_s, while assuming that the machine is operating in a standard operating condition, and connected to a 60Hz or 50Hz source. That large resistance models both the I^2R losses and the mechanical power given to the rotor. Since the resistive voltage drop of the outer mesh is caused to the very largest extent by the rotor circuit equivalent resistive element, any error of reasonable size in the stator resistance estimation will not skew largely the calculated torque.

However it is important to point out, that the stator resistive element has a much larger influence in the torque equations if the machine was operated, not at line frequencies, but at much lower frequencies from an adjustable frequency drive. If the efficiency estimator presented here was to be implemented for such an environment, then the stator resistance estimation would have to be re-evaluated.

3.3 Calculation of flux linkage:

(3.2) shows the equation which has been used for calculating the time varying flux linkage. Nevertheless, it has to be adapted slightly for the practical implementation within the data acquisition system. Due to the discrete nature of the data acquisition system, the integral is replaced by a sum. The flux linkage resembles closely a sinusoidal function even for varying load points (the voltage regulation caused by the varying load current when calculating the airgap flux linkage will not affect largely the shape of the curve). The sine wave that describes the flux linkage is oscillating around zero, having no net flux linkage component over time.

Translating (3.2) into a time-discrete system,

$$\psi_{q,d} = \sum_{t=0}^{t=T} v_{q,d}(t) \tag{3.3}$$

If the data acquisition is not synchronized perfectly to a zero crossing of the q and the d voltage components, a dc-offset will result from (3.3). That offset can be avoided in a variety of ways, one of them by subtracting the mean of ψ(t). An other method, which is well described in [8] has been employed, since it offers further advantages. A forgetting factor ff is introduced, which is a constant slightly smaller than one, and multiplied every step where ψ(t) is calculated, as described in (3.4)

$$\psi_{q,d}(n+1) = ff \cdot \psi_{q,d}(n) + v_{q,d}(n+1) \tag{3.4}$$

In (3.4), n is the sample number. If, by the particular start in time of the data acquisition process, a constant dc offset would have been created according to (3.3), that value would be diminished by the sample n as described by (3.5).

$$offset(n) = offset(0) \cdot (ff)^n \tag{3.5}$$

(3.5) also describes another and important advantage of this particular implementation of the flux linkage calculation. The influence of noise has to be evaluated for every physical data acquisition and processing setup. If the forgetting factor would not have been included, the noise in the measurement would be present in every calculated ψ(n) value posterior to the sample with the noise. The influence of noise is attenuated using (3.4) exponentially with the same time constant as the dc offset is in (3.5). That time constant is a function of ff and the sampling rate.

The value ff has to be chosen with relative care. If ff is too close to one, there would not be sufficient attenuation of the dc offset and the influence of noise. If ff is too small, the rms value of ψ would be unacceptable small, and a leading phase shift would be introduced to the calculated ψ(t), skewing the resulting torque calculation.

4. Implementation

4.1 Hardware environment

The methods that have been presented in chapters 1 through 3 were implemented in a PC based data acquisition system. The currents were measured using NANA Hall-effect sensors, which have an output of 2mV/A and which need a DC power supply of $\pm$15V referenced to ground. A good signal to noise ratio of the current signal was ensured by feeding multiple turns of the current conductor through the current sensors.

The voltage signals were obtained using Tektronix differential voltage probes, which require a 110V voltage supply. The array of three voltage sensors was wye-connected, using the grounded motor case as reference. The choice of connecting the sensors in wye assured proper alignment of phase currents and phase voltages, making power factor calculations simple.

The six signals, three currents and three voltages, were fed into a sample and hold card from National Instruments, which resides in a SCXI chassis. This sample and hold card has the function of ensuring time-synchronicity of the six data channels at every sample, so that a displacement power factor error is avoided. Figure 4.1 shows schematically the hardware configuration.

The concatenation of sensor arrays and data acquisition board was calibrated to the operating range using a Fluke 5500A calibrating unit, which secured a good quality of the measured quantities. The use of the calibrating unit, which allows phase-locking of a voltage source to a current source, generating variable phase shifts, permitted ensuring as well that the implemented RMS calculation algorithms for power and power factor ran properly. The RMS calculations were also checked by introducing a harmonic component into the voltage created, while the current remained sinusoid. These two tests warranted an accurate power factor calculation, for any combination of displacement and harmonic components.

Since the present work intends to reach as far as to be a proof of concept and not a marketable prototype, no additional investigation into acquisition-based errors was performed.

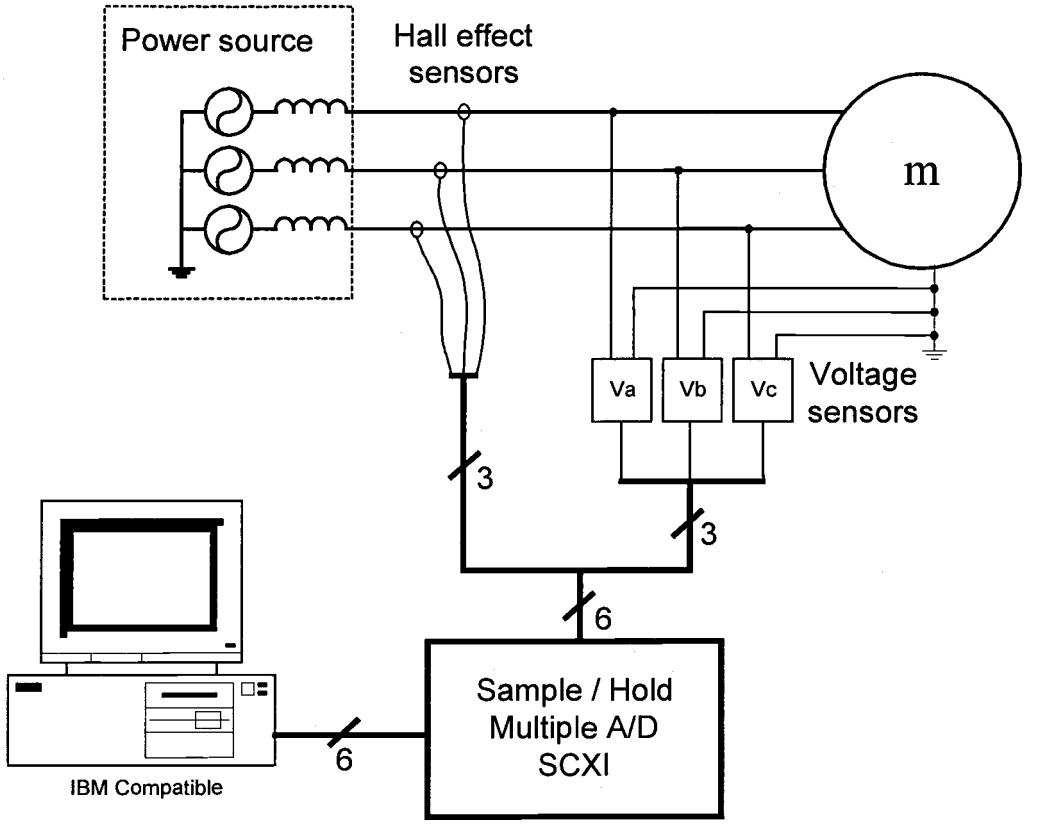


Figure 4.1: Hardware configuration of the experimental setup

A commercial unit would not rely on these selected off-the-shelf sensor arrays and signal conditioning, but rather on a custom designed signal generation and conditioning system. Opto-isolating the signals would increase operator's safety, and low-passing the generated signals would avoid possible aliasing problems. Aliasing problems were avoided in this implementation largely by acquiring data with high sampling frequencies, commonly above 2 kSamples/sec.

4.2 Software environment:

The software environment that was used is LabWindows/CVI (CVI), from National Instruments. This software packet interacts easily with the data acquisition hardware, and

includes a large library of powerful data processing functions, which made the data processing easier to implement.

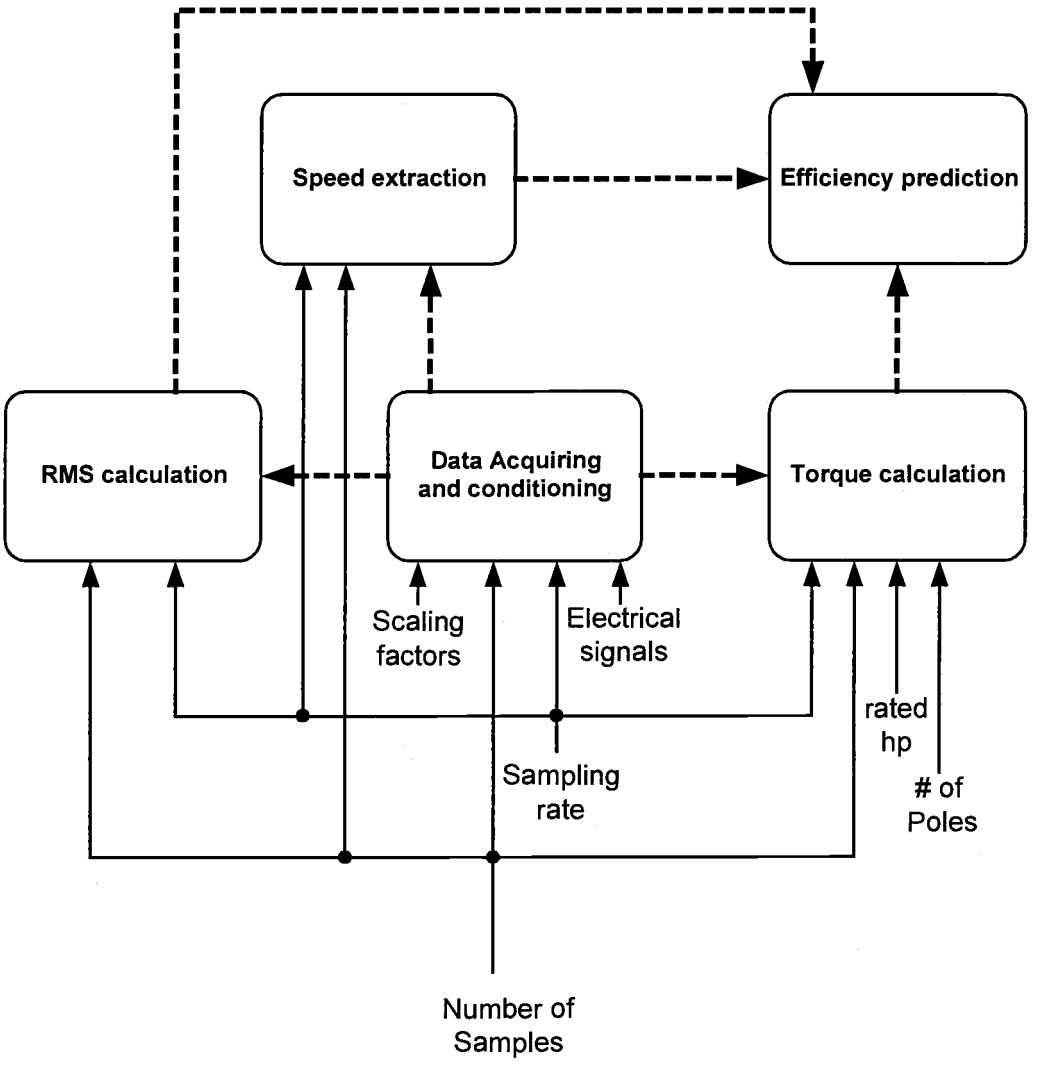


Figure 4.2: Block diagram of the software implementation of the Efficiency Estimation prototype.

The highest level schematic, which shows the process of efficiency estimation and the required inputs for each stage, is shown in Figure 4.2.

4.2.1 Data acquiring and conditioning:

As can be seen in Figure 4.2., the required inputs for this block are the number of samples, the sampling rate and the scaling factors. In addition, of course, the electrical signals to be measured must also be connected to the data acquisition board.

The CVI function "AIAcquireWaveforms" is used for the process of data acquiring. The CVI code can be seen in Appendix D. AIAcquireWaveforms acquires the specified number of samples at the specified sampling rate for the six input channels: the three currents and the three voltages. The SCXI based sampling and hold board insures that the acquired data is synchronous in the time domain, so that no phase shift is introduced during acquisition.

The output of the AIAcquireWaveforms function is a string of six times the length of the number of scans, where the results for each channel are consequently stored. The values are equal to the signal level voltage fed into each channel. The resulting string is then divided into its six components, Ia, Ib, Ic, Va, Vb and Vc. At the same time, the strings are scaled by multiplying the acquired voltage level with the appropriate constant, which changes the signal level voltage value to the true physical quantity in Volts or Amps. The scaling factors were obtained by calibrating the unit using a Fluke 5500A unit, as previously described.

At this point the operator has the chance to run a digitally implemented lowpass function if desired, where cutoff frequency, order and type can be entered. The types of lowpass filters that are available are Butterworth, Chebyshev, elliptic and inverse elliptic filters. All these filters are implemented using standard functions available in the CVI packet.

4.2.2 RMS calculation:

The inputs to this part of the program are the scaled data strings, the number of samples and the sampling rate, as can be seen in Fig. 4.2.. The current and voltage strings are multiplied with each other, resulting in instantaneous per channel Volt-Amps. Next, the three voltage and the three current strings are each inputted into the rms function which is provided in the CVI environment. The rms function is based on the strict definition of root mean square quantity for discrete signals as follows:

$$rms(f) = \frac{\sqrt{\sum_i (f(i))^2}}{n} \tag{4.1}$$

Where n represents the number of samples of the function $f(i)$. The resulting Iarms, Ibrms, Icrms, Varms, Vbrms and Vcrms are printed on the main panel of the efficiency estimation software. The three voltage rms values, as well as the three current rms values are averaged, and these averages are displayed on the main panel as well for an easy first assessment of the operating load condition.

The next calculation is the numeric integration of the volt-amps channels, which is used to calculate the real power per channel, P_j. Real power is defined in (4.2) for the time-continuous case:

$$P_j = \frac{1}{T} \int_{t_0}^{t_0+T} \left(v_j(t) \cdot i_j(t)\right) dt \tag{4.2}$$

A numeric approximation is required for this integral, since we deal with time-discrete signals here. The algorithm which is proposed by the CVI environment and which was implemented for this case is

$$\begin{aligned} F(0) &= 0 \\ F(n+1) &= F(n) + \frac{1}{6}\left(f(n-1) + 4 \cdot f(n) + f(n+1)\right) \end{aligned} \tag{4.3}$$

The heavy weighing of the central integration sample, while still permitting some smaller influence of the neighboring samples makes this integral algorithm slightly more suited for comparatively rapidly changing signals. In this case this positive feature is of no crucial advantage however, since, as stated previously, care is taken to over sample thus avoiding aliasing problems. The resulting integrating operation is performed in place, so that the former string is overwritten by its own integral.

The resulting individual-phase power is equal to the last value of the resulting integrated Volt-Amps string, divided by the sampling time. The three individual-phase powers are displayed on the main panel, as is the resulting total power, which equals the sum of the three individual-phase powers.

Next the individual-phase power factors and the three phase power factor are calculated. The multiplication of the rms value of each channel's voltage and current equals to the individual-phase complex power, and power factor is defined as

$$pf = \frac{P}{S} \tag{4.4}$$

Where S is the complex power. The total power factor is not obtained as an average of the three individual-phase power factors, but as the division of the sum of the real powers divided by the sum of the complex powers:

$$pf_{tot} = \frac{\sum_i P_i}{\sum_i S_i} \tag{4.5}$$

The three individual-phase and the total power factor are displayed on the main panel.

4.2.3 Torque calculation:

The method by which the torque is estimated has been presented in Chapter 3. In this part of the work only the procedure itself will be presented. The schematic of the procedure is shown in Fig. 4.3.

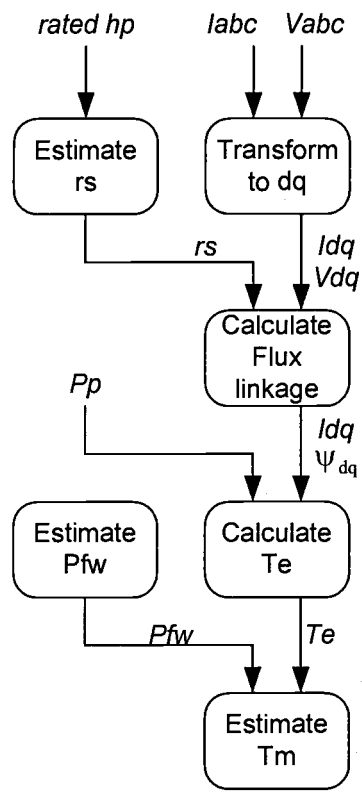


Figure 4.3: Block diagram of Torque estimation

Estimation rs:

The stator resistance rs is required for calculation of airgap torque in the dq domain. In the previous chapter it has been stated that the torque calculation is not very sensitive to comparatively small errors made in estimating rs. For that reason the estimation procedure of the stator resistance is limited to a rule of thumb, which states that smaller machines have a pu stator resistance of about 0.1pu, while large machines tend towards

0.01pu. A knowledge based lookup table is used, and then the pu stator resistance is transferred into Ω, using the measured operating voltage and entered machine rating as the equivalents of the voltage and machine power.

dq Transformation:

The required matrices to transform currents and voltages from the abc domain into the dq domain were presented in chapter 2:

$$\begin{pmatrix} f_q \\ f_d \\ f_0 \end{pmatrix} = [K_s] \begin{pmatrix} f_a \\ f_b \\ f_c \end{pmatrix} \tag{4.6}$$

$$[K_s] = \frac{2}{3} \begin{pmatrix} \cos\theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \sin\theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ ½ & ½ & ½ \end{pmatrix} \tag{4.7}$$

$$[K_s]^{-1} = \begin{pmatrix} \cos\theta & \sin\theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) & 1 \end{pmatrix} \tag{4.8}$$

Since the analysis is performed in the stationary reference frame, θ is constant, and can be chosen at the author's discretion, 0 being the most simple value. The transformation is of an instantaneous nature, which translates into a point-by-point procedure in the time-discrete case. The required inputs are the current and voltage strings in the abc domain, and the output is the current and voltage strings in the dq domain.

Calculation of Flux linkage:

The equation which defines the stator flux linkage has been presented in Chapter 3 as

$$\psi_{q,d}(t) = \int v_{q,d}(t) \cdot dt \tag{4.9}$$

where the $v_{q,d}$ are the terminal voltages minus stator resistance drops. For this reason, the estimated stator resistance r_S, as well as the dq domain voltages and currents are required inputs for this part of the function.

$$v_{q,d} = v_{\text{terminal}_{q,d}} - r_s \cdot i_{q,d} \tag{4.10}$$

The integration is performed with the use of the forgetting factor ff, for reasons which have been explained in the previous chapter.

$$\psi_{q,d}(n+1) = ff \cdot \psi_{q,d}(n) + v_{q,d}(n+1) \tag{4.11}$$

A sensible dimension for the forgetting factor has been found by requiring the influence of a noisy sample to drop to 10% of its value within a sixth of the total number of samples. This translates to

$$0.1 = ff^{N/6} \tag{4.12}$$

Which can be solved for the forgetting factor ff as a function of the total number of samples N. The resulting string of flux linkage is defined as a function of time and the dq axes, which means that it has the same length as the current and voltage strings.

Calculation of Electrical Torque, Te:

The electrical torque, or airgap torque, is defined in chapter 3 for the stator reference frame and the dq axes as

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P_p}{2}\right)\left(\psi'_{qs} i'_{ds} - \psi'_{ds} i'_{qs}\right) \tag{4.13}$$

The dq currents and the dq domain flux linkages have previously been calculated on a sample by sample basis, and the pole pair P_p is the other required input for this part of the function. It is possible to calculate the torque that is acting from the stator to the rotor with these inputs. The result of (4.13) is a function of time, which means that a single torque value results for every data acquired sample.

Estimation of Friction and windage Power and Mechanical Torque:

The difference between the operating torque of a machine, and the electrical, or airgap torque, is caused by the friction and windage and stray load losses. These losses will be estimated on a knowledge based table in future work. This table will be a function of pole pair number and rating of the machine, perhaps being also a function of the design letter of the machine and must be based on statistical relevant data. The data can be obtained by performing a series of laboratory tests of machines of different pole pair numbers and hp ratings, but the scope of such a testing procedure clearly exceeds the scope of the presented work.

Since the proof of concept here presented will be evaluated on a set of motors of 0.75hp and 1hp, with 4 and 6 numbers of poles, it will not be possible to gather data spanning over the different ratings and pole pair numbers as required for a full implementation of the estimation of mechanical output torque T_m.

The required friction and windage torque estimation is based on the results obtained by measuring the first part of the set of motors on a dynamometer, and subtracting the measured mechanical output torque T_m from the calculated electrical output torque T_e. The result of this subtraction is the friction and windage and stray load loss torque T_{fw}. This T_{fw} is used for the latter set of machines evaluated on the same dynamometer setup. This second series of tests will be able to show how closely the total estimation of mechanical output torque really is to the measured mechanical output torque.

4.2.4 Speed extraction:

The speed extraction procedure is based on signatures of instantaneous signals. It has been found during the research, that basing the extraction of speed solely on frequency side bands of the current frequency spectrum is not an extremely reliable approach. The three predicted speeds, one for each current signal, disagree sometimes, or point to another speed than the actually operating one.

To avoid erroneous estimations, many different instantaneous signals are evaluated, which increases the reliability of the speed prediction. An instantaneous signal is here referred to as the calculation of a physical quantity. This quantity can be extracted from the instantaneous currents and voltages at the machine terminals by analytical or other signal processing means, including analogue computing.

Table 4.1. shows a list of the instantaneous signals currently being used, and the location of the frequency side bands of interest:

Table 4.1: Instantaneous signals used, and locations of the side bands of interest.

Signal name	Upper side band	Lower side band	Lowpass
Individual-phase current	fs+fr	fs-fr	No
Current vector	2fs-fr	Fr	Yes
Individual-phase power	2fs-fr	Fr	No
Sum of powers	2fs-fr	Fr	No
Individual-phase imaginary power	2fs-fr	Fr	No
Sum of imaginary powers	2fs-fr	Fr	No
Absolute value of impedance	2fs-fr	Fr	Yes
Angle of impedance	2fs-fr	Fr	Yes
Instantaneous torque	2fs-fr	Fr	Yes

The low-pass field is marked with Yes for all the signals which are utilized two times for the speed estimation, one time before the low-passing procedure, and one time after low-passing the current and voltage signals, as will be explained in detail later this chapter. The implemented speed estimating procedure can be represented as shown in the block diagram of Figure 4.4.

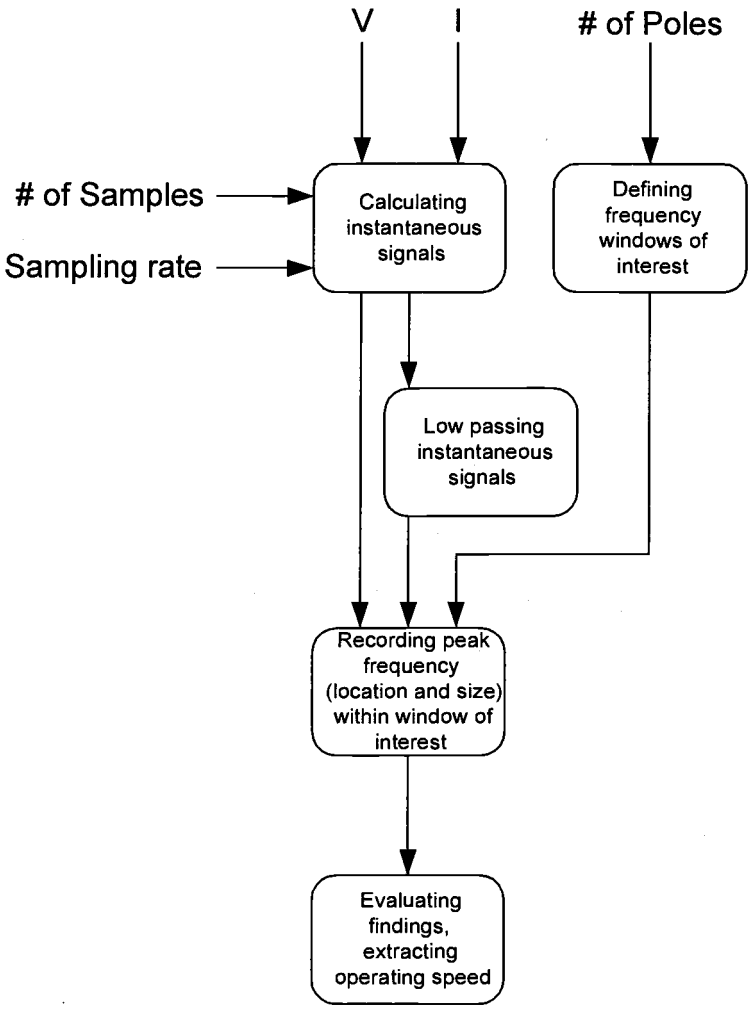


Figure 4.4: Speed estimation procedure.

Calculation of instantaneous signals:

The required inputs for calculating the instantaneous signals are the data acquired current and voltage strings, as well as the number of samples and the sampling rate. The method with which the instantaneous signals enumerated in Table 4.1. are obtained, is described in the following paragraphs.

Individual-phase current: There are three individual-phase currents, and each one will be used independently for speed estimation purposes. No further signal processing is required for this case.

Current vector: The current vector, or current phasor, is obtained in this particular implementation from the dq domain currents, and only the amplitude of the current phasor is used for speed extraction purposes. The amplitude of the current phasor is an instantaneous quantity, which is defined in the dq domain as described in equation (4.14)

$$I_{phasor}(t) = \sqrt{\left(i_d(t)\right)^2 + \left(i_q(t)\right)^2} \tag{4.14}$$

Individual-phase powers: The instantaneous signal called individual-phase powers, is not equal to the strict definition of the individual-phase power, given in equation (4.15) for any phase i.

$$P_i(t_0) = \frac{1}{T}\int_{t_0}^{t_0+T}\left(i_i(t)\cdot v(t)\right)dt \tag{4.15}$$

What is used in this work as the instantaneous individual-phase power is the multiplication of the instantaneous current and the instantaneous voltage signals:

$$p_i(t) = v_i(t)\cdot i_i(t) \tag{4.16}$$

Equation (4.16) is easier to implement than the strict definition of (4.15), where considerations for the choice of the integration period, T, have to be made, which might influence strongly the shape of the spectra to be calculated. In the general case it is not possible to describe frequencies of interest as either integer multiples or as integer divisions of the fundamental. The reason for this is that the operating speed is a variable that can vary on a continuous scale, and not only on states that are rational functions of

the harmonics. Example 4.1 shows how a particular choice of T could mask the frequency of interest, if it were calculated by equation (4.15).

Example 4.1: If a 4 pole machine were given, which was truly operating at synchronous speed, then the signatures of the instantaneous power signals would be expected at 30Hz and at 90Hz. If T was chosen to be equal to two cycles $\left(\frac{2}{60}s\right)$ of the fundamental frequency of 60Hz, then the result of the spectrum of $P(t)$ would not show any component at either the 30Hz, or the 90Hz location, since

$$\sin(2\cdot\pi\cdot 30\cdot t)=\sin\left(2\cdot\pi\cdot 30\cdot\left(t+\tfrac{2}{60}\right)\right) \tag{4.17}$$

for all t.

Sum of powers: The instantaneous signal called sum of powers equals to the sum of the three instantaneous individual-phase power signals. The sum is performed on a sample by sample basis.

Individual-phase imaginary power: The individual-phase power signal can be interpreted as the scalar product of the instantaneous current vector with the instantaneous voltage vector for each phase. The resulting equation of the vector multiplication, in the abc domain, is equal to (4.16).

It is well known that the imaginary power is orthogonal to the real power, which can be verified by looking at the power triangle in Figure 4.5. The significance of this fact can be used in defining the instantaneous imaginary power as the cross product of the instantaneous current vector times the instantaneous voltage vector. The relative frequency components of the resulting signal will be proportional to

$$q_i(t)=\left(v_j(t)-v_k(t)\right)\cdot i_i(t) \tag{4.18}$$

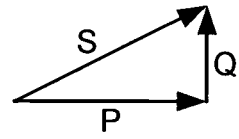


Figure 4.5: Power triangle.

where i, j, k is any permutation of the phases a, b, c. (4.18) is a very simple implementation, on a sample by sample basis, which can be obtained from the abc domain.

Sum of imaginary powers: The instantaneous signal called sum of imaginary powers equals to the sum of the three instantaneous individual-phase imaginary power signals. The sum is performed on a sample by sample basis.

Absolute value of impedance: Instantaneous impedance signals can be defined either on a per phase basis, or as a complex impedance (in the phasor domain). The complex definition of the instantaneous impedance signal has been chosen here. The instantaneous impedance is given in equation (4.19), and the phasors are shown in Figure 4.6.

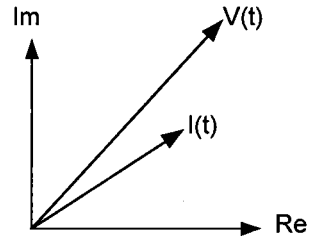


Figure 4.6: I(t) and V(t) phasor diagram

$$Z(t) = \frac{V(t)}{I(t)} \tag{4.19}$$

It is important to keep in mind that the phasors here depicted are not constant quantities, since they are modulated by harmonics, sub-harmonics and other frequencies, which are the key to speed estimation procedures. Also, even if the current phasor is a function of time, it will not be zero at any given time during operation of the IM, hence Z will not be singular at any point in time.

The instantaneous impedance, which can be represented as a complex variable, has two components, its absolute value, and its phase. The latter will now be addressed below.

Angle of impedance: As stated in the last paragraph, the instantaneous signal of the impedance can be split in a variety of forms, two of which are the polar (absolute value and angle) or the Cartesian forms (d- and q-axes). In the present case the polar form has been chosen. Both, the amplitude and the angle translate to quantities which are mainly constants, having only small time-variations, which carry the frequency information to be extracted.

Instantaneous torque: Extracting the operating torque is a requirement for the efficiency estimation procedure shown here. The dq analysis employed results in an instantaneous torque signal, which is also used for speed extraction.

Low passing of instantaneous signals:

In the previous paragraphs it has been described how instantaneous signals of physically meaningful quantities can be obtained. These instantaneous signals can be obtained with analogue circuitry, and then processed digitally, or they can also be calculated from the digitized current and voltage signals. In either case they represent quantities which are frequently utilized to describe a motor in operation.

Many of the instantaneous signals can be conceptualized in the phasor domain, which aides the understanding of occurring phenomena. A phasor is represented by a vector of

a constant magnitude and a constant orientation. The relative orientation of phasors to each other represent the constant phase shifts, or time-lags between the sinusoidal quantities represented. The phasor domain is commonly defined for one particular frequency, which obviously can not be done if other frequencies are present.

An extended understanding of the phasor domain, including instantaneous and time-varying phasors, is utilized in the vector based control of IMs. This concept is also necessary to illustrate the usefulness of the low-passed signals for detection of the speed.

The instantaneous impedance is a phasor, resulting from the division of two instantaneous phasors, and as such, it has an absolute amplitude, and an angle. The voltage and the current phasors are varying with time caused by (Dynamic eccentricities, harmonics, sub-harmonics, electro-mechanical resonances on the power grid, etc.). The time variation of the current and the voltage phasors are not equal. The frequency components present in the voltage spectrum are modulated by the complex impedance.

Lower frequency components will cause larger currents than will higher frequencies, since IMs are inductive. Additionally, if there were no other frequencies present in the voltage spectrum than the fundamental, then the inherent dynamic eccentricities would induce current components of other frequencies than solely in the fundamental, as has been shown in Chapter 2. Yet an other source producing differences between the voltage phasor spectrum and the current phasor spectrum is the inherent non-linearities of the magnetic material of the IM. The result is that the division represented in (4.19) is not a constant. This means that the impedance vector will be modulated as a function of time, constantly changing slightly its amplitude and its phase.

Lets assume that there is only one frequency present which modulates the impedance phasor in a frequency band of interest. That frequency could manifest itself in three different forms in the phasor domain, as seen in Fig.4.7. Obviously, the circular trajectory and the linear trajectory case can be described as special cases of the elliptic case. In Fig. 4.7, however, it is important to point out that the size of the instantaneous phasor is exaggerated when compared with the motors that were tested. The side bands on the instantaneous signals were always less than 30dB below the size of the fundamental (60Hz).

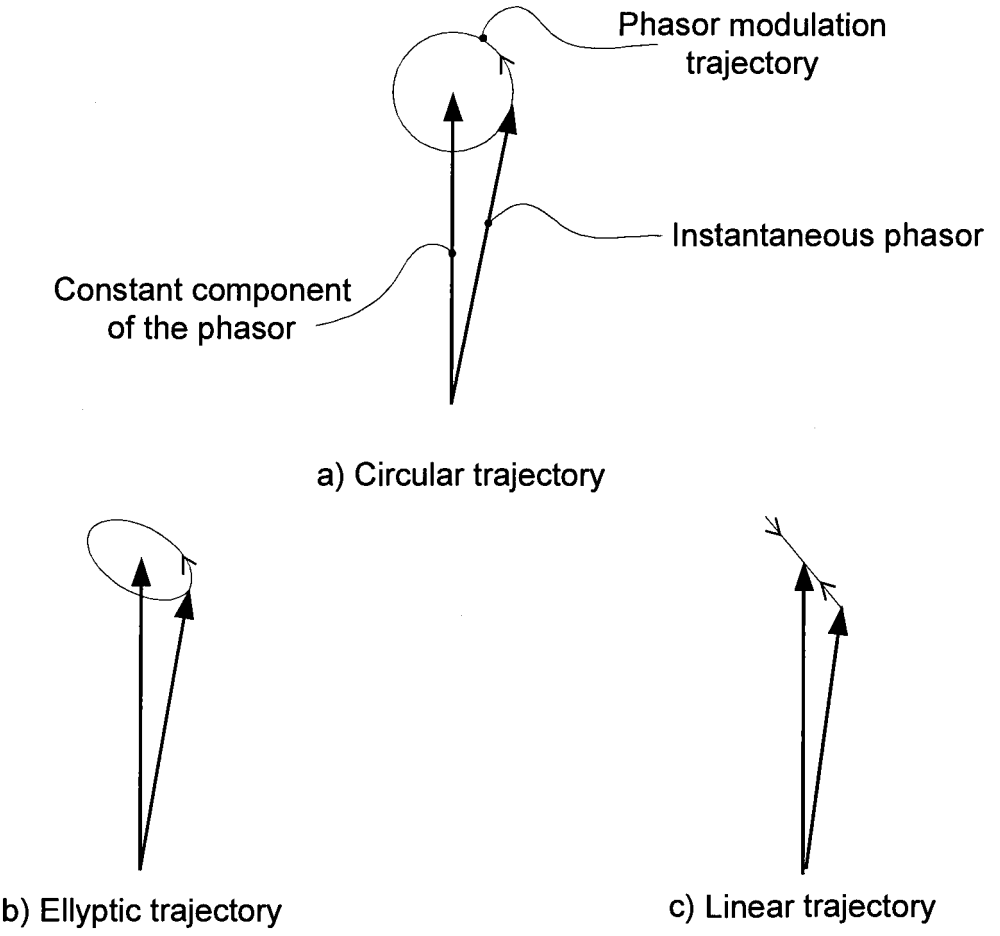


Figure 4.7: Phasor domain representation of instantaneous signals

A more realistic, yet still strongly exaggerated representation of the circular trajectory case can be seen in Fig. 4.8.a. Here, only a small portion of each end of the phasors is drawn. The constant component of the phasor is that part of the signal, which can be extracted as the DC component in the phasor domain. The phasor domain, however, is referenced to the line frequencies, 60Hz in our case.

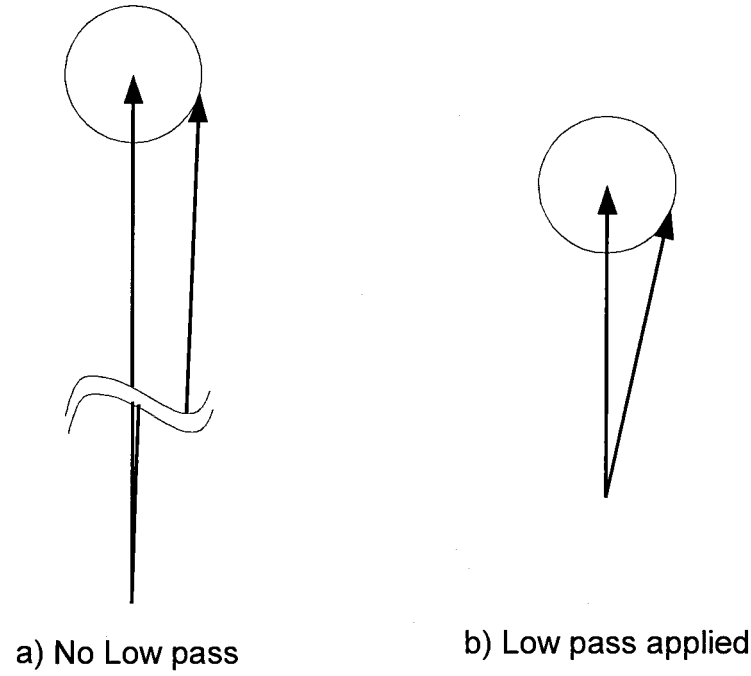


Figure 4.8: Result of applying the low pass in the phasor domain.

The results of the simulations of Chapter 2 showed that the largest component which carries the speed information in the instantaneous signal's spectrum, is the band below the 60Hz line frequency. Fig. 4.8.b shows the result in the phasor domain of implementing a low pass filter with a very sharp stop band and a cutoff frequency laying between the low frequency side band and the fundamental. The results of this sharp lowpass filter would ideally strongly dampen the constant component of the phasor, while not dampening much the size of the trajectory of the instantaneous phasor. Any phase shift introduced by this approach has no influence on the calculated speed, since only the frequency and not the phase shift is information of importance for speed extraction.

Defining frequency windows of interest:

Table 4.1 shows the locations at which speed information can be found for the investigated instantaneous signals. The presented implementation makes only use of the

lower side band, since its size has always been found to be greater than the upper side band frequency's.

For the individual-phase currents, the lower side band is located at fs-fr, while for all the other signals it is located at fr. It has been assumed that the operating condition of the IM will not have a slip larger than 180 r/min, which translates into 3Hz in the frequency domain. Hence, the frequency windows of interest are

$$f_{wdw} = 60\text{Hz} - f_s \quad \ldots \quad \left(60\text{Hz} - f_s + 3\text{Hz}\right) \qquad \text{individual-phase currents} \qquad (4.20.a)$$

$$f_{wdw} = f_s - 3\text{Hz} \quad \ldots \quad f_s \qquad \text{other signals} \qquad (4.20.b)$$

Extracting operating speed:

It is a simple task to calculate the frequency components of any acquired instantaneous signal. Among other methods, it is possible to transform an acquired signal to the frequency domain via FFT methods, or via the previously presented DFLL algorithm. The latter method is very computationally intensive, particularly if one needs frequency information over a large bandwidth. But regardless of the particular method chosen, it is simple to obtain a frequency domain mapping of an instantaneous signal over a known frequency window, as the one specified in the previous paragraph.

Once the frequency spectrum is obtained for every instantaneous signal, the location of the peak frequency within that window is recorded. Additionally, a number, which describes the relative strength of that signal is recorded. That number, called signal strength, is obtained according to (4.21).

$$SigStrenght = \frac{\sum_{i=i_{peak}-N/20}^{i_{peak}+N/20} \left(signalFreq(i)\right)^2}{\sum_{i=i_{\min}}^{i_{\max}} \left(signalFreq(i)\right)^2} \qquad (4.21)$$

Where $signalFreq$ is the array with the frequency representation of the signal. i_{min} and i_{max} are the boundary samples of the frequency window of interest. N is the number of samples within that window, and i_{peak} is the sample with the peak frequency within the window of interest. (4.21) effectively calculates the relative power of the peak frequency within the window, including the surrounding 10% of the frequencies surrounding the peak, with respect to the total energy present in that frequency band.

The next step that has to be taken is to extract the true operating speed from the wealth of partly contradicting information. The count of evaluated instantaneous signals (channels) of Table 4.1. is equal to 19. Very rarely does the speed predicted by any one of these channels coincide with that predicted by one all the other channels. Three different methods of extracting the most probable speed were implemented. These will now be described:

Average method: The first method that was implemented was a plain average of all the predicted speeds. The result was not a very reliable speed prediction, since the few channels that did not coincide with the other channels, were predicting speeds that were not close to the actual operating point. This skewed the predicted resulting speed prediction unacceptably.

Average of the majority method: This method also predicts the speed from the average of many single channel predictions. The difference resides in the fact that only a group of channels is accepted to input their predictions into the mean building process. These channels are the largest group of channels for which predictions coincides, within a certainty of 2r/min.

The average of the majority method has been found to be very reliable, and extremely accurate. The accuracy of the steady state speed prediction is bound by the resolution obtained from the frequency mapping procedure, which is generally limited by the size of the data acquisition period that was chosen.

The reliability of this method was found to be far exceeding 95%. Additionally reliability could be achieved by increasing the period of data acquisition further.

The average of the majority of the stronger signals: It was noticed, however, that the rare cases could be avoided in which a wrong speed was predicted, if the speed extraction was done by an operator. The reason resided in the fact that the operator additionally evaluates the relative strength of the peak frequency, dropping the channels which showed suspiciously weak peak frequencies.

This process was implemented by first analyzing the strength of all the predicted 19 channels, and dropping the 6 channels with the weakest signals. After this is done, the speed prediction is performed according to the average of the majority of the channels.

This method has been found to be extremely reliable, only returning the wrong speed predictions, in cases where excessive compromises are made by shortening the data acquisition time. The data acquisition times that are currently being used are either 15 seconds for the FFT frequency mapping, or 6 seconds for the DFLL method.

For Pentium computers faster or equal to 166MHz, it has been found that the total time used for the speed prediction is shorter with the DFLL method, even including the substantially larger number processing effort. For slower microprocessors, like the also evaluated 486 66MHz computer, the time spent data processing the DFLL case exceeded 3 minutes, compared to 30 seconds for the FFT case.

4.2.5 Efficiency prediction:

Fig. 4.2. shows that only three inputs are required for this step. First, form the RMS calculation block, the sum of powers is used as input power. Output power is calculated using (4.24)

$$P_{out} = 2 \cdot \pi \cdot T_m \cdot f_r \tag{4.24}$$

based on the results of T_m and f_r from the torque calculation and speed extraction

procedures respectively. Efficiency results, as stated in Chapter 1, are calculated from (4.25)

$$\eta = \frac{P_{out}}{P_{in}} \tag{4.25}$$

5. Results

5.1 Experimental Equipment

The theory and implementation presented in previous chapters, were tested on an experimental configuration. A series of 6 small IMs were tested on a Baker Instrument Company AMT 1200 automated motor tester dynamometer. The equipment is designed for baseline testing in production environments. This automated full load dynamometer consists of the following components:

A compressed air operated ram holds the motor under test in place for the duration of the test; a compressed air operated clutch holds the shaft of the motor under test, and is connected via a coupling to the torque and speed transducer; the transducer is connected via a second coupling to the dc machine, which is operated via a bi-directional drive, to provide for speed and torque control.

Fast cycle time for testing of IMs is optimized, frequently achieving full load cycle testing times of only 6 seconds in an 'automated' test mode. During this test time, pull-up torque, break-down torque and a torque-speed curve are calculated, as well as voltage, current and power conditions for the whole operating range. Obviously, the production environment does not permit long enough test times for each produced motor to allow for temperature stabilization.

Conversely, the 'manual' test mode, permits setting of both test time and load condition. This was the mode of operation utilized in all measurements presented in this thesis. Temperature stabilization is indeed of significant importance in gauging the efficiency estimator's knowledge based friction and windage estimation lookup table, and was carefully ensured.

The fast cycle times required for production line testing of small IMs, however, result in a different hardware configuration for the dynamometer, when compared with laboratory-quality dynamometers. Set up and load testing of each motor has to be achieved within a total of 10 seconds for small motors in a production environment. This makes the use of standard rigid couplings between the motor under test and the torque transducer not an option. Instead, a pressured air operated clutch is utilized.

5.2 Dynamometer torque correction factor

The clutch system described above introduces unmeasured losses between the motor under test and the torque transducer. Hence, the losses occurring in the clutch are part of the output power of the motor, while not observable on the torque transducer. This loss is comparable to the dynamometer correction factor present in the IEEE 112B testing procedure when true shaft torque is not measured. Dynamometer correction factors have as premise that friction and windage elements present in the test bed can be measured prior to measuring the motor under test.

It was assumed that friction and windage losses present in the AMT 1200 setup are only a function of speed. For this reason, torque measured on the torque transducer, was mapped against speed, with no motor physically in place. The torque measured on the transducer, which represents the losses in the clutch mechanism, was obviously of opposite direction to the torque measured during loaded motoring operation.

Tables 5.1 and 5.2 show the measured friction and windage torque as a function of speed for the speed ranges of interest for 4- and 6 pole motors.

Table 5.1: Friction and Windage torque of AMT for 4-pole speed range

Speed [r/min]	F&W Torque [Nm]
1800	0.3534
1780	0.347
1760	0.3458
1740	0.34
1720	0.335

The tables 5.1 and 5.2 were used for building a linear regression of friction and windage torque (dynamometer correction torque) against the operating speed. The dynamometer correction torque was added to the torque measured at the torque transducer for each particular operating torque, resulting in the true output torque.

Table 5.2: Friction and Windage torque of AMT for 6-pole speed range

Speed [r/min]	F&W Torque [Nm]
1200	0.3958
1180	0.3789
1160	0.3698
1140	0.3626
1120	0.3604

5.3 Differences in estimated torque caused by stator resistance estimation

Chapter 3 presented both the chosen method for estimating the operating torque of IMs, also the method used for estimating stator resistances was shown. Differences between the true stator resistance and the estimated value will result in differences between estimated operating torque and true operating torque. It was stated previously that small differences in stator resistances will not influence resulting torque prediction with a similar order of magnitude than the deviation between estimated stator resistance and true stator resistance. This statement will be validated by the following study.

Using measurements performed on one particular motor at 4 different load points; 25%, 50%, 75% and 100% of rated load, a sensitivity study was completed investigating the influence of misestimations of stator resistances. Torque predictions were calculated for a range of stator resistances. The span of stator resistances, normalized to the true stator resistance (4.2 Ω for the investigated 1hp 4 pole Design B motor), extended from 50% up to 150%. The predicted torque was calculated for each of the 4 load points for a totality of 21 estimated stator resistances. In order to allow comparison of the predicted torques at the different load points, it was also necessary to normalize the predicted torques. This was accomplished using the true estimated torque, defined as the predicted torque calculated with the true stator resistance of 4.2 Ω. The results of this study are shown in Fig. 5.1.

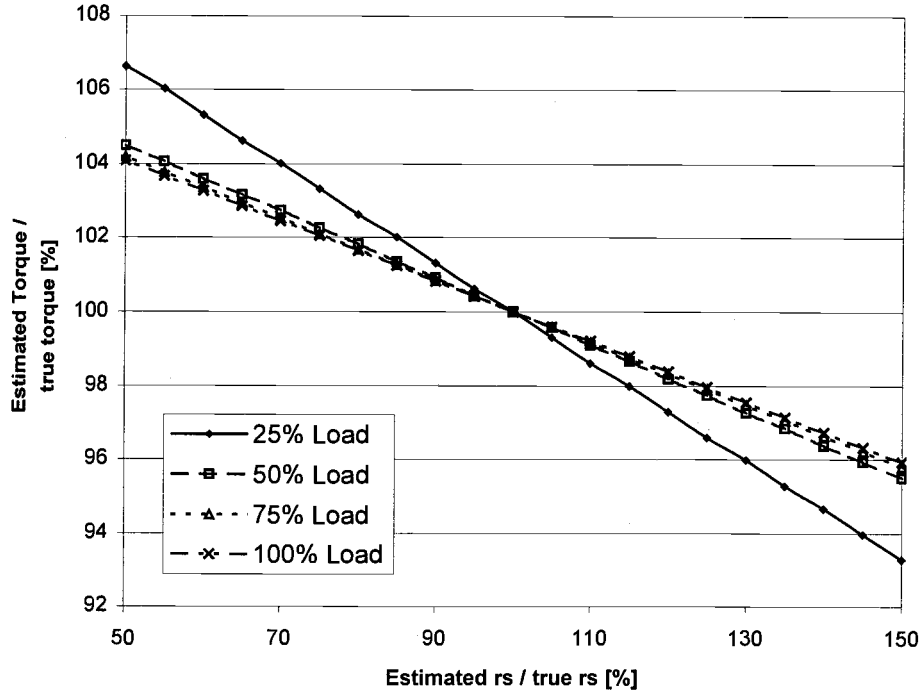


Figure 5.1: Influence of stator resistance estimation on predicted torque.

It is clear from Fig. 5.1, that a very severe misestimation of the stator resistance will result in a comparatively mild error in the predicted torque, for the analysis of line operated motors. For example, the error due to misestimation of the stator resistance by 50%, on a 25% load point, will be lower than 7%. This is a worst case scenario, since larger loading decreases the error. Additionally, larger motors will have a much lower stator resistance in pu, which will lead to lower $I^2 r_s$ errors distorting the torque prediction even less.

5.4 Test series finding friction, windage and stray load losses

Chapter 3 also stated that values for friction, windage and stray load loss are required for predicting the operating efficiency with the airgap torque method. Values for these

losses should be obtained from a knowledge based lookup table. In order to find typical values for low horse power motors, six IMs were tested on the AMT 1200 platform. The torque was measured as a dc signal obtained from a Himmelstein model MCRT torque transducer. The dc signal was calibrated according to the manufacturer's instructions. The signal was also fed directly into the same data acquisition system, that measured the electrical quantities, ensuring synchronicity for the recorded electrical and mechanical data.

The measured electrical quantities were connected to the efficiency estimation system described in Chapter 4. Since this set of measurements and calculations is used to obtain the knowledge based values for expected friction, windage and stray load losses, the estimated values of these losses were set to zero. The difference between the measured (dynamometer corrected) and the predicted (zero friction, windage and load losses) torques will reflect the true loss torques. In order to minimize other possible sources of errors, the stator resistance was not estimated but measured and entered into the calculations for each motor.

Ten measurements were performed and the acquired instantaneous signals stored for each of the four load points for each of the six motors tested. The multiplicity of measurements can be used as a check for reproducibility of the estimated quantities, particularly for the estimated friction, windage and stray load losses.

It is not possible to maintain the exact operating condition at any particular load point over time. The reason is that the dynamometer drive has a hysteresis based feedback loop for the load control, hence the operating torque can freely fluctuate within the hysteresis band. For this reason it is not advisable to check reproducibility on strongly load dependent values, as input power or output torque.

Another advantage of the multiplicity of measurements is that small errors committed during measurement can be decreased by averaging the ten measurements.

The tables on the next pages show an extract record of the results obtained by testing the six motors on the four load conditions. Temperature stabilization was ensured by operating each motor for at least 45 minutes, whilst checking the temperature of the casing with an IR sensor for temperature fluctuations. The load conditions were set in

ascending order in an attempt to minimize the temperature step between former operating condition and the current one.

A set of two tables is required for each motor. The motors were numbered, and their nameplate information has been copied to the upper part of the first table for each motor. The first three motors are of the same brand, rating and model: Design B, General Electric 1hp, 4-pole motors. The other three motors are of the same manufacturer, Reliance, and of the same model line, differing in rating and pole pairs, however. Motors 4 and 5 are 0.75 hp machines, the former a 4-pole and the latter a 6-pole, while the last motor tested is a 1.5 hp 4-pole machine. The Reliance motors specified neither their design nor their nominal efficiency.

The rest of the tables shows: the results of the test; the compound average line to line voltage; the average line current; the total electrical input power; the average power factor. The operating speed shown was obtained by the speed extraction procedure previously presented and checked with a handheld digital counter numerous times. The measured torque was gained directly from the electrical output signal that was fed into the data acquisition system. The dynamometer correction factor results from a regression of the measured no load friction and windage losses of the AMT 1200 system for the operating speed range, as previously presented. The operating torque is the addition of the measured torque plus the dynamometer correction factor. The estimated torque is the result of the torque calculations presented in Chapter 3, excluding any friction, windage and stray load losses and also incorporating the true (rather than the estimation of the) stator resistance, which was measured off line. The column titled F&W Trq shows the subtraction of the estimated torque minus the operating torque. The mechanical output power is the multiplication of operating speed in radians, times the operating torque. Dividing mechanical by electrical power leads to the results entered into the efficiency column.

The units of the following tables are: voltages in V, currents in A, electrical power in W, speed in r/min and dynamometer correction and torques in Nm.

Table 5.3: F&W results motor 1 part 1.

Motor 1

Name plate Data:							
Hp:	1	Volts:	208-230/460	rpm:	1745	ph:	3
Hz:	60	code:	M	Frame:	143TC	Design:	B
Insul:	F	duty:	const.	s.f.:	1.25		
Eff.:	80%	Model:	F029	Type:	UT		

Note: F&W estimate set to 0W L-L Stator resistance = 4.2 ohms

25% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im1_25_0	215	2.2	243	0.30	1789.2	0.549	0.351	0.900	1.140	0.240	168.5	69.3
Im1_25_1	215	2.2	240	0.29	1789.1	0.505	0.351	0.856	1.120	0.264	160.2	66.8
Im1_25_2	215	2.2	241	0.30	1789.1	0.528	0.351	0.879	1.100	0.221	164.5	68.3
Im1_25_3	215	2.2	240	0.29	1789.2	0.521	0.351	0.872	1.120	0.248	163.2	68.0
Im1_25_4	215	2.2	233	0.29	1789.3	0.492	0.351	0.843	1.090	0.247	157.8	67.7
Im1_25_5	215	2.2	240	0.29	1789.0	0.532	0.351	0.883	1.100	0.217	165.3	68.9
Im1_25_6	215	2.2	235	0.29	1789.9	0.496	0.351	0.847	1.080	0.233	158.6	67.5
Im1_25_7	215	2.2	227	0.28	1790.1	0.445	0.351	0.796	1.060	0.264	149.1	65.7
Im1_25_8	215	2.2	231	0.28	1790.0	0.475	0.351	0.826	1.080	0.254	154.7	67.0
Im1_25_9	215	2.2	226	0.28	1789.8	0.464	0.351	0.815	1.060	0.245	152.6	67.5
								0.851		**0.244**		**67.7**

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im1_50_0	214	2.4	432	0.48	1776.0	1.506	0.348	1.854	2.090	0.236	344.6	79.8
Im1_50_1	214	2.4	431	0.48	1776.6	1.5	0.348	1.848	2.070	0.222	343.6	79.7
Im1_50_2	214	2.4	428	0.48	1776.7	1.481	0.348	1.829	2.060	0.231	340.1	79.5
Im1_50_3	214	2.4	430	0.48	1776.7	1.491	0.348	1.839	2.070	0.231	342.0	79.5
Im1_50_4	214	2.4	434	0.48	1776.0	1.517	0.348	1.865	2.070	0.205	346.6	79.9
Im1_50_5	214	2.4	435	0.48	1776.3	1.515	0.348	1.863	2.100	0.237	346.3	79.6
Im1_50_6	214	2.4	427	0.48	1776.7	1.485	0.348	1.833	2.040	0.207	340.8	79.8
Im1_50_7	214	2.4	425	0.48	1776.7	1.484	0.348	1.832	2.030	0.198	340.7	80.2
Im1_50_8	214	2.4	426	0.48	1777.0	1.482	0.348	1.830	2.040	0.210	340.4	79.9
Im1_50_9	214	2.4	432	0.48	1776.8	1.508	0.348	1.856	2.050	0.194	345.1	79.9
								1.845		**0.217**		**79.8**

Table 5.4: F&W results motor 1 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im1_75_0	214	2.8	642	0.62	1762.0	2.489	0.345	2.834	3.100	0.266	522.6	81.4
Im1_75_1	214	2.8	643	0.62	1761.9	2.497	0.345	2.842	3.100	0.258	524.0	81.5
Im1_75_2	214	2.8	641	0.61	1762.0	2.481	0.345	2.826	3.060	0.234	521.1	81.3
Im1_75_3	214	2.8	640	0.61	1762.0	2.475	0.345	2.820	3.060	0.240	520.0	81.3
Im1_75_4	214	2.8	642	0.61	1761.9	2.489	0.345	2.834	3.070	0.236	522.6	81.4
Im1_75_5	214	2.8	644	0.62	1761.7	2.496	0.345	2.841	3.080	0.239	523.8	81.3
Im1_75_6	214	2.8	643	0.62	1762.0	2.492	0.345	2.837	3.100	0.263	523.1	81.4
Im1_75_7	214	2.8	642	0.62	1762.0	2.479	0.345	2.824	3.090	0.266	520.8	81.1
Im1_75_8	214	2.8	643	0.62	1761.5	2.496	0.345	2.841	3.110	0.269	523.7	81.4
Im1_75_9	2.14	2.8	641	0.61	1761.5	2.493	0.345	2.838	3.080	0.242	523.2	81.6
								2.833		**0.252**		**81.4**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im1_100_0	214	3.2	859	0.72	1745.0	3.536	0.341	3.877	4.130	0.253	708.1	82.4
Im1_100_1	214	3.2	856	0.72	1745.7	3.516	0.341	3.857	4.100	0.243	704.8	82.3
Im1_100_2	214	3.2	855	0.72	1745.9	3.508	0.341	3.849	4.100	0.251	703.4	82.3
Im1_100_3	214	3.2	854	0.72	1745.8	3.502	0.341	3.843	4.110	0.267	702.2	82.2
Im1_100_4	214	3.2	854	0.72	1745.7	3.488	0.341	3.829	4.080	0.251	699.6	81.9
Im1_100_5	214	3.2	855	0.72	1745.6	3.506	0.341	3.847	4.080	0.233	702.9	82.2
Im1_100_6	214	3.2	855	0.72	1745.6	3.506	0.341	3.847	4.090	0.243	702.9	82.2
Im1_100_7	214	3.2	850	0.72	1745.9	3.503	0.341	3.844	4.080	0.236	702.5	82.6
Im1_100_8	214	3.2	846	0.71	1746.0	3.488	0.341	3.829	4.060	0.231	699.8	82.7
Im1_100_9	214	3.2	851	0.71	1745.8	3.513	0.341	3.854	4.090	0.236	704.3	82.8
								3.848		**0.244**		**82.4**

Table 5.5: F&W results motor 2 part 1.

Motor 2

Name plate Data:							
Hp:	1	Volts:	208-230/460	rpm:	1745	ph:	3
Hz:	60	code:	M	Frame:	143TC	Design:	B
Insul:	F	duty:	const.	s.f.:	1.25		
Eff.:	80%	Model:	F029	Type:	UT		

Note: F&W estimate set to 0W L-L Stator resistance = 4.2 ohms

25% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im2_25_0	213	2.2	244	0.30	1788.9	0.507	0.351	0.858	1.040	0.182	160.6	65.8
Im2_25_1	214	2.2	243	0.30	1788.9	0.499	0.351	0.850	1.030	0.180	159.1	65.5
Im2_25_2	214	2.2	248	0.30	1788.2	0.504	0.350	0.854	1.050	0.196	159.9	64.5
Im2_25_3	214	2.2	240	0.30	1788.9	0.502	0.351	0.853	1.010	0.157	159.6	66.5
Im2_25_4	214	2.2	238	0.29	1789.0	0.484	0.351	0.835	1.020	0.185	156.3	65.7
Im2_25_5	214	2.2	235	0.29	1788.9	0.456	0.351	0.807	1.000	0.193	151.0	64.3
Im2_25_6	214	2.2	233	0.29	1789.0	0.47	0.351	0.821	0.990	0.169	153.7	65.9
Im2_25_7	214	2.2	232	0.29	1789.3	0.45	0.351	0.801	0.970	0.169	149.9	64.6
Im2_25_8	214	2.2	242	0.30	1788.6	0.508	0.351	0.859	1.020	0.161	160.7	66.4
Im2_25_9	214	2.2	239	0.29	1789.0	0.494	0.351	0.845	1.010	0.165	158.1	66.2
								0.838		**0.176**		**65.5**

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im2_50_0	213	2.4	425	0.48	1776.3	1.437	0.348	1.785	1.950	0.165	331.8	78.1
Im2_50_1	213	2.4	429	0.48	1776.1	1.466	0.348	1.814	1.980	0.166	337.2	78.6
Im2_50_2	213	2.4	430	0.48	1776.1	1.463	0.348	1.811	1.970	0.159	336.6	78.3
Im2_50_3	213	2.4	429	0.48	1775.9	1.466	0.348	1.814	1.980	0.166	337.1	78.6
Im2_50_4	213	2.4	429	0.48	1775.9	1.455	0.348	1.803	1.980	0.177	335.1	78.1
Im2_50_5	213	2.4	434	0.48	1776.0	1.478	0.348	1.826	1.990	0.164	339.4	78.2
Im2_50_6	213	2.4	433	0.48	1776.0	1.471	0.348	1.819	2.000	0.181	338.1	78.1
Im2_50_7	213	2.4	428	0.48	1776.0	1.461	0.348	1.809	1.970	0.161	336.2	78.6
Im2_50_8	213	2.4	426	0.48	1776.2	1.45	0.348	1.798	1.950	0.152	334.2	78.5
Im2_50_9	213	2.4	423	0.48	1776.0	1.429	0.348	1.777	1.950	0.173	330.3	78.1
								1.805		**0.167**		**78.3**

Table 5.6: F&W results motor 2 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im2_75_0	215	2.8	644	0.62	1761.0	2.495	0.344	2.839	3.040	0.201	523.4	81.3
Im2_75_1	215	2.8	646	0.62	1761.0	2.505	0.344	2.849	3.040	0.191	525.2	81.3
Im2_75_2	215	2.8	648	0.62	1760.9	2.519	0.344	2.863	3.070	0.207	527.8	81.4
Im2_75_3	215	2.8	648	0.62	1760.9	2.519	0.344	2.863	3.070	0.207	527.8	81.4
Im2_75_4	214	2.8	643	0.62	1761.2	2.486	0.345	2.831	3.030	0.199	521.8	81.1
Im2_75_5	215	2.8	642	0.62	1761.3	2.495	0.345	2.840	3.040	0.200	523.5	81.5
Im2_75_6	215	2.8	643	0.62	1761.1	2.495	0.344	2.839	3.040	0.201	523.4	81.4
Im2_75_7	215	2.8	648	0.62	1760.9	2.502	0.344	2.846	3.070	0.224	524.6	81.0
Im2_75_8	215	2.8	641	0.62	1761.0	2.484	0.344	2.828	3.050	0.222	521.3	81.3
Im2_75_9	214	2.8	639	0.61	1761.1	2.474	0.344	2.818	3.030	0.212	519.5	81.3
								2.842		**0.206**		**81.3**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im2_100_0	213	3.2	849	0.72	1744.0	3.5	0.341	3.841	4.040	0.199	701.1	82.6
Im2_100_1	213	3.2	848	0.72	1744.0	3.492	0.341	3.833	4.070	0.237	699.6	82.5
Im2_100_2	213	3.2	852	0.72	1743.9	3.52	0.341	3.861	4.080	0.219	704.7	82.7
Im2_100_3	213	3.2	846	0.72	1744.0	3.495	0.341	3.836	4.030	0.194	700.2	82.8
Im2_100_4	213	3.2	857	0.72	1743.9	3.538	0.341	3.879	4.090	0.211	708.0	82.6
Im2_100_5	213	3.2	854	0.72	1744.0	3.527	0.341	3.868	4.070	0.202	706.0	82.7
Im2_100_6	213	3.2	850	0.72	1744.0	3.505	0.341	3.846	4.070	0.224	702.0	82.6
Im2_100_7	213	3.2	853	0.72	1743.5	3.521	0.341	3.862	4.060	0.198	704.7	82.6
Im2_100_8	213	3.2	849	0.72	1744.0	3.504	0.341	3.845	4.060	0.215	701.8	82.7
Im2_100_9	213	3.2	848	0.72	1743.9	3.491	0.341	3.832	4.040	0.208	699.4	82.5
								3.850		**0.211**		**82.6**

Table 5.7: F&W results motor 3 part 1.

Motor 3

Name plate Data:							
Hp:	1	Volts:	208-230/460	rpm:	1745	ph:	3
Hz:	60	code:	M	Frame:	143T C	Design:	B
Insul:	F	duty:	const.	s.f.:	1.25		
Eff.:	80%	Model:	F029	Type:	UT		

Note: F&W estimate set to 0W L-L Stator resistance = 4.2 ohms

25% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im3_25_0	214	2.3	246	0.29	1789.0	0.553	0.351	0.904	1.140	0.236	169.2	68.8
Im3_25_1	214	2.3	233	0.28	1789.1	0.479	0.351	0.830	1.080	0.250	155.4	66.7
Im3_25_2	214	2.3	240	0.29	1788.9	0.518	0.351	0.869	1.110	0.241	162.6	67.8
Im3_25_3	214	2.3	241	0.29	1788.9	0.530	0.351	0.881	1.110	0.229	164.9	68.4
Im3_25_4	214	2.3	238	0.28	1789.1	0.511	0.351	0.862	1.130	0.268	161.3	67.8
Im3_25_5	214	2.3	235	0.28	1789.2	0.505	0.351	0.856	1.090	0.234	160.2	68.2
Im3_25_6	214	2.3	240	0.29	1789.0	0.522	0.351	0.873	1.120	0.247	163.4	68.1
Im3_25_7	214	2.3	243	0.29	1788.9	0.532	0.351	0.883	1.150	0.267	165.3	68.0
Im3_25_8	214	2.3	241	0.29	1788.9	0.522	0.351	0.873	1.120	0.247	163.4	67.8
Im3_25_9	214	2.3	246	0.29	1788.9	0.561	0.351	0.912	1.160	0.248	170.7	69.4
								0.874		**0.247**		**68.1**

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im3_50_0	213	2.5	432	0.47	1775.2	1.479	0.348	1.827	2.080	0.253	339.4	78.6
Im3_50_1	213	2.5	435	0.47	1775.0	1.487	0.348	1.835	2.080	0.245	340.8	78.4
Im3_50_2	213	2.5	436	0.47	1775.0	1.490	0.348	1.838	2.080	0.242	341.4	78.3
Im3_50_3	213	2.5	437	0.47	1775.2	1.487	0.348	1.835	2.090	0.255	340.9	78.0
Im3_50_4	213	2.5	432	0.47	1775.3	1.468	0.348	1.816	2.080	0.264	337.4	78.1
Im3_50_5	213	2.5	433	0.47	1775.3	1.471	0.348	1.819	2.100	0.281	337.9	78.0
Im3_50_6	213	2.5	433	0.47	1775.1	1.470	0.348	1.818	2.090	0.272	337.7	78.0
Im3_50_7	213	2.5	437	0.47	1775.2	1.483	0.348	1.831	2.100	0.269	340.1	77.8
Im3_50_8	213	2.5	433	0.47	1775.3	1.470	0.348	1.818	2.080	0.262	337.7	78.0
Im3_50_9	213	2.5	436	0.47	1775.0	1.482	0.348	1.830	2.070	0.240	339.9	78.0
								1.826		**0.259**		**78.1**

Table 5.8: F&W results motor 3 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im3_75_0	212	2.8	640	0.61	1760.0	2.479	0.344	2.823	3.090	0.267	520.1	81.3
Im3_75_1	212	2.9	651	0.62	1759.0	2.540	0.344	2.884	3.160	0.276	531.0	81.6
Im3_75_2	213	2.9	652	0.62	1759.0	2.549	0.344	2.893	3.160	0.267	532.6	81.7
Im3_75_3	213	2.9	652	0.62	1759.2	2.547	0.344	2.891	3.140	0.249	532.3	81.6
Im3_75_4	213	2.9	656	0.62	1759.0	2.557	0.344	2.901	3.150	0.249	534.1	81.4
Im3_75_5	213	2.9	652	0.62	1759.1	2.542	0.344	2.886	3.150	0.264	531.4	81.5
Im3_75_6	213	2.9	652	0.62	1759.0	2.556	0.344	2.900	3.170	0.270	533.9	81.9
Im3_75_7	213	2.9	653	0.62	1759.0	2.549	0.344	2.893	3.130	0.237	532.6	81.6
Im3_75_8	212	2.9	646	0.62	1759.3	2.522	0.344	2.866	3.150	0.284	527.8	81.7
Im3_75_9	212	2.8	642	0.61	1760.0	2.498	0.344	2.842	3.090	0.248	523.6	81.6
								2.878		**0.261**		**81.6**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im3_100_0	213	3.3	862	0.71	1741.3	3.550	0.340	3.890	4.150	0.260	709.0	82.3
Im3_100_1	212	3.3	859	0.71	1741.1	3.530	0.340	3.870	4.160	0.290	705.3	82.1
Im3_100_2	213	3.3	863	0.71	1741.1	3.544	0.340	3.884	4.150	0.266	707.8	82.0
Im3_100_3	213	3.3	864	0.71	1741.0	3.558	0.340	3.898	4.170	0.272	710.3	82.2
Im3_100_4	213	3.3	862	0.71	1741.3	3.549	0.340	3.889	4.140	0.251	708.8	82.2
Im3_100_5	213	3.3	864	0.71	1741.0	3.563	0.340	3.903	4.140	0.237	711.2	82.3
Im3_100_6	213	3.3	865	0.71	1741.0	3.553	0.340	3.893	4.150	0.257	709.4	82.0
Im3_100_7	213	3.3	865	0.71	1741.1	3.571	0.340	3.911	4.150	0.239	712.7	82.4
Im3_100_8	213	3.3	865	0.71	1741.0	3.567	0.340	3.907	4.160	0.253	712.0	82.3
Im3_100_9	213	3.3	863	0.71	1741.2	3.548	0.340	3.888	4.140	0.252	708.6	82.1
								3.893		**0.258**		**82.2**

Table 5.9: F&W results motor 4 part 1.

Motor 4

Name plate Data:							
Hp:	0.75	**Volts:**	208-230/460	**rpm:**	1725	**ph:**	3
Hz:	60	**code:**	J	**Frame:**	EC56	**Design:**	
Insulation:	F	**duty:**	const.	**s.f.:**	1.15	**Amps:**	2.9
Nom. Eff.:		**Model:**		**Type:**			
MFD Model:		**No:**	P56H1301VZW	L-L Stator resistance = 9.05 ohms			

Note: F&W estimate set to 0W

25% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im4_25_0	214	1.7	181	0.30	1787.0	0.207	0.350	0.557	0.810	0.253	104.2	57.6
Im4_25_1	214	1.9	179	0.29	1787.5	0.184	0.350	0.534	0.790	0.256	100.0	55.8
Im4_25_2	214	1.6	178	0.29	1787.6	0.182	0.350	0.532	0.810	0.278	99.6	56.0
Im4_25_3	214	1.7	182	0.30	1786.9	0.197	0.350	0.547	0.810	0.263	102.3	56.2
Im4_25_4	214	1.7	180	0.29	1787.3	0.180	0.350	0.530	0.790	0.260	99.2	55.1
Im4_25_5	214	1.7	180	0.29	1787.3	0.191	0.350	0.541	0.830	0.289	101.2	56.2
Im4_25_6	214	1.7	182	0.30	1787.1	0.210	0.350	0.560	0.820	0.260	104.8	57.6
Im4_25_7	214	1.6	179	0.29	1787.3	0.195	0.350	0.545	0.800	0.255	102.0	57.0
Im4_25_8	214	1.6	178	0.29	1787.5	0.185	0.350	0.535	0.800	0.265	100.1	56.3
Im4_25_9	214	1.6	178	0.29	1787.4	0.179	0.350	0.529	0.810	0.281	99.0	55.6
								0.541		**0.266**		**56.3**

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Speed	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im4_50_0	214	1.8	339	0.50	1769.3	0.981	0.346	1.327	1.590	0.263	245.8	72.5
Im4_50_1	214	1.8	327	0.49	1770.3	0.928	0.346	1.274	1.530	0.256	236.2	72.2
Im4_50_2	214	1.8	337	0.50	1769.0	0.978	0.346	1.324	1.570	0.246	245.2	72.8
Im4_50_3	214	1.8	340	0.51	1769.1	0.984	0.346	1.330	1.580	0.250	246.3	72.4
Im4_50_4	214	1.8	338	0.50	1769.2	0.978	0.346	1.324	1.610	0.286	245.2	72.6
Im4_50_5	214	1.8	341	0.51	1769.0	0.994	0.346	1.340	1.590	0.250	248.1	72.8
Im4_50_6	214	1.8	341	0.51	1769.0	0.990	0.346	1.336	1.590	0.254	247.4	72.6
Im4_50_7	214	1.8	340	0.51	1769.0	1.001	0.346	1.347	1.610	0.263	249.4	73.4
Im4_50_8	214	1.8	341	0.51	1769.0	0.998	0.346	1.344	1.610	0.266	248.9	73.0
Im4_50_9	214	1.8	344	0.51	1769.0	1.011	0.346	1.357	1.620	0.263	251.3	73.1
								1.331		**0.259**		**72.7**

Table 5.10: F&W results motor 4 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im4_75_0	214	2.1	504	0.65	1748.8	1.756	0.342	2.098	2.360	0.262	384.0	76.2
Im4_75_1	214	2.1	502	0.65	1748.1	1.745	0.342	2.087	2.370	0.283	381.8	76.1
Im4_75_2	214	2.1	503	0.65	1748.3	1.746	0.342	2.088	2.370	0.282	382.0	75.9
Im4_75_3	214	2.1	503	0.65	1748.1	1.752	0.342	2.094	2.380	0.286	383.1	76.2
Im4_75_4	214	2.1	503	0.65	1749.0	1.757	0.342	2.099	2.380	0.281	384.2	76.4
Im4_75_5	214	2.1	504	0.65	1749.0	1.760	0.342	2.102	2.380	0.278	384.8	76.3
Im4_75_6	214	2.1	504	0.65	1749.0	1.768	0.342	2.110	2.370	0.260	386.2	76.6
Im4_75_7	214	2.1	504	0.65	1748.0	1.763	0.342	2.105	2.380	0.275	385.1	76.4
Im4_75_8	214	2.1	504	0.65	1749.0	1.757	0.342	2.099	2.370	0.271	384.2	76.2
Im4_75_9	214	2.1	500	0.65	1748.9	1.744	0.342	2.086	2.360	0.274	381.8	76.4
								2.097		**0.275**		**76.3**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im4_100_0	214	2.4	685	0.76	1722.0	2.534	0.336	2.870	3.180	0.310	517.3	75.5
Im4_100_1	214	2.4	686	0.76	1722.0	2.531	0.336	2.867	3.200	0.333	516.7	75.3
Im4_100_2	214	2.4	686	0.76	1722.0	2.541	0.336	2.877	3.210	0.333	518.5	75.6
Im4_100_3	214	2.4	682	0.76	1722.5	2.523	0.336	2.859	3.180	0.321	515.4	75.6
Im4_100_4	214	2.4	681	0.76	1723.0	2.526	0.336	2.862	3.170	0.308	516.2	75.8
Im4_100_5	214	2.4	684	0.76	1722.0	2.529	0.336	2.865	3.180	0.315	516.4	75.5
Im4_100_6	214	2.4	683	0.76	1722.2	2.519	0.336	2.855	3.180	0.325	514.6	75.3
Im4_100_7	214	2.4	682	0.76	1722.0	2.521	0.336	2.857	3.200	0.343	514.9	75.5
Im4_100_8	214	2.4	680	0.76	1722.8	2.507	0.336	2.843	3.180	0.337	512.7	75.4
Im4_100_9	214	2.4	680	0.76	1722.7	2.514	0.336	2.850	3.180	0.330	513.9	75.6
								2.860		**0.326**		**75.5**

Table 5.11: F&W results motor 5 part 1.

Motor 5

Name plate Data:							
Hp:	0.75	**Volts:**	208-230/460	**rpm:**	1140	**ph:**	3
Hz:	60	**Code:**	J	**Frame:**	EC56	**Design:**	
Insulation:	F	**Duty:**	const.	**s.f.:**	1.15	**Amps:**	2.9
Eff.:		**Model:**		**Type:**			
MFD Model:		**No:**	P56H3020TZW	L-L Stator resistance = 8.4 ohms			

Note: F&W estimate set to 0W

25% Load												
File name	**Vave**	**Iave**	**Pel tot.**	**pf ave**	**Spd**	**Meas. Trq.**	**Dyn Corr.**	**Op. Trq.**	**Est. Trq.**	**F&W Trq**	**Pmech**	**Eff.**
Im5_25_0	215	1.9	210	0.29	1188.3	0.743	0.386	1.129	1.420	0.291	140.4	66.9
Im5_25_1	215	1.9	214	0.30	1188.0	0.772	0.386	1.158	1.430	0.272	144.0	67.3
Im5_25_2	215	1.9	213	0.30	1188.1	0.763	0.386	1.149	1.410	0.261	142.9	67.1
Im5_25_3	215	1.9	214	0.30	1188.0	0.772	0.386	1.158	1.430	0.272	144.0	67.3
Im5_25_4	215	1.9	214	0.30	1188.3	0.762	0.386	1.148	1.420	0.272	142.8	66.7
Im5_25_5	216	1.9	215	0.30	1188.0	0.770	0.386	1.156	1.420	0.264	143.7	66.9
Im5_25_6	215	1.9	213	0.30	1188.0	0.759	0.386	1.145	1.440	0.295	142.4	66.8
Im5_25_7	215	1.9	212	0.30	1188.0	0.757	0.386	1.143	1.380	0.237	142.1	67.0
Im5_25_8	215	1.9	212	0.30	1188.0	0.752	0.386	1.138	1.450	0.312	141.5	66.7
Im5_25_9	215	1.9	213	0.30	1188.1	0.743	0.386	1.129	1.430	0.301	140.4	65.9
								1.145		**0.278**		**66.9**

50% Load												
File name	**Vave**	**Iave**	**Pel tot.**	**pf ave**	**Spd**	**Meas. Trq.**	**Dyn Corr.**	**Op. Trq.**	**Est. Trq.**	**F&W Trq**	**Pmech**	**Eff.**
Im5_50_0	215	2.1	370	0.47	1174.1	1.878	0.380	2.258	2.540	0.282	277.5	75.0
Im5_50_1	215	2.1	371	0.47	1174.8	1.875	0.380	2.255	2.560	0.305	277.3	74.7
Im5_50_2	215	2.1	369	0.47	1174.6	1.868	0.380	2.248	2.540	0.292	276.4	74.9
Im5_50_3	215	2.1	371	0.47	1174.8	1.870	0.380	2.250	2.520	0.270	276.7	74.6
Im5_50_4	215	2.1	372	0.47	1174.2	1.885	0.380	2.265	2.550	0.285	278.4	74.8
Im5_50_5	215	2.1	373	0.47	1174.4	1.881	0.380	2.261	2.560	0.299	277.9	74.5
Im5_50_6	215	2.1	372	0.47	1174.4	1.880	0.380	2.260	2.520	0.260	277.8	74.7
Im5_50_7	215	2.1	374	0.47	1174.1	1.888	0.380	2.268	2.540	0.272	278.7	74.5
Im5_50_8	215	2.1	373	0.47	1174.1	1.894	0.380	2.274	2.530	0.256	279.4	74.9
Im5_50_9	215	2.1	369	0.47	1174.6	1.864	0.380	2.244	2.540	0.296	275.9	74.8
								2.258		**0.282**		**74.7**

Table 5.12: F&W results motor 5 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im5_75_0	215	2.4	544	0.60	1158.5	3.047	0.373	3.420	3.720	0.300	414.7	76.2
Im5_75_1	215	2.4	544	0.60	1158.2	3.053	0.373	3.426	3.730	0.304	415.3	76.3
Im5_75_2	215	2.4	546	0.60	1158.0	3.055	0.373	3.428	3.730	0.302	415.5	76.1
Im5_75_3	215	2.4	546	0.60	1158.2	3.060	0.373	3.433	3.730	0.297	416.2	76.2
Im5_75_4	215	2.4	545	0.60	1158.0	3.049	0.373	3.422	3.720	0.298	414.7	76.1
Im5_75_5	215	2.4	545	0.60	1158.3	3.051	0.373	3.424	3.730	0.306	415.1	76.2
Im5_75_6	215	2.4	543	0.60	1158.1	3.041	0.373	3.414	3.720	0.306	413.8	76.2
Im5_75_7	215	2.4	542	0.60	1158.2	3.036	0.373	3.409	3.700	0.291	413.2	76.2
Im5_75_8	215	2.4	543	0.60	1158.2	3.048	0.373	3.421	3.700	0.279	414.7	76.4
Im5_75_9	215	2.4	541	0.60	1158.3	3.045	0.373	3.418	3.720	0.302	414.4	76.6
								3.421		**0.299**		**76.3**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im5_100_0	214	2.9	738	0.69	1139.0	4.256	0.364	4.620	4.980	0.360	550.8	74.6
Im5_100_1	214	2.9	732	0.69	1139.0	4.232	0.364	4.596	4.920	0.324	548.0	74.9
Im5_100_2	214	2.9	727	0.69	1139.2	4.188	0.365	4.553	4.900	0.347	542.8	74.7
Im5_100_3	214	2.9	728	0.69	1139.1	4.214	0.364	4.578	4.880	0.302	545.9	75.0
Im5_100_4	214	2.9	728	0.69	1139.3	4.197	0.365	4.562	4.930	0.368	544.0	74.7
Im5_100_5	214	2.9	732	0.69	1139.0	4.219	0.364	4.583	4.980	0.397	546.4	74.6
Im5_100_6	214	2.9	728	0.69	1139.1	4.203	0.364	4.567	4.900	0.333	544.6	74.8
Im5_100_7	214	2.8	725	0.69	1139.3	4.177	0.365	4.542	4.930	0.388	541.6	74.7
Im5_100_8	214	2.8	726	0.69	1138.8	4.199	0.364	4.563	4.890	0.327	543.9	74.9
Im5_100_9	214	2.8	725	0.69	1139.1	4.186	0.364	4.550	4.890	0.340	542.5	74.8
								4.572		**0.348**		**74.8**

Table 5.13: F&W results motor 6 part 1.

Motor 6

Name plate Data:							
Hp:	1.5	**Volts:**	208-230/460	**rpm:**	1140	**ph:**	3
Hz:	60	**code:**	J	**Frame:**	EC56	**Design:**	
Insul:	F	**duty:**	const.	**s.f.:**	1.15	**Amps:**	2.9
Eff.:		**Model:**		**Type:**			
Model:		**No:**	P56H3020TZW	L-L Stator resistance = 3.4 ohms			

Note: F&W estimate set to 0W

25% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im6_25_0	211	3.2	365	0.31	1785.9	0.985	0.350	1.335	1.53	0.195	249.5	68.4
Im6_25_1	212	3.2	362	0.31	1786.0	0.987	0.350	1.337	1.53	0.193	249.9	69.0
Im6_25_2	212	3.2	357	0.31	1786.0	0.963	0.350	1.313	1.50	0.187	245.4	68.7
Im6_25_3	212	3.2	362	0.31	1785.8	0.996	0.350	1.346	1.52	0.174	251.6	69.5
Im6_25_4	211	3.2	357	0.31	1785.9	0.979	0.350	1.329	1.51	0.181	248.4	69.6
Im6_25_5	211	3.2	356	0.31	1786.0	0.972	0.350	1.322	1.50	0.178	247.1	69.4
Im6_25_6	211	3.2	360	0.31	1785.5	0.985	0.350	1.335	1.52	0.185	249.5	69.3
Im6_25_7	211	3.2	354	0.31	1785.9	0.950	0.350	1.300	1.48	0.180	243.0	68.6
Im6_25_8	211	3.2	354	0.31	1786.0	0.948	0.350	1.298	1.48	0.182	242.6	68.5
Im6_25_9	211	3.2	352	0.3	1785.9	0.941	0.350	1.291	1.48	0.189	241.3	68.6
								1.321		**0.184**		**69.0**

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im6_50_0	211	3.5	658	0.51	1769.1	2.490	0.346	2.836	3.01	0.174	525.2	79.8
Im6_50_1	211	3.6	667	0.51	1769.0	2.531	0.346	2.877	3.05	0.173	532.7	79.9
Im6_50_2	211	3.6	666	0.51	1769.0	2.531	0.346	2.877	3.05	0.173	532.7	80.0
Im6_50_3	211	3.6	661	0.51	1769.0	2.505	0.346	2.851	3.03	0.179	527.9	79.9
Im6_50_4	211	3.6	659	0.51	1769.0	2.499	0.346	2.845	3.01	0.165	526.8	79.9
Im6_50_5	211	3.5	660	0.51	1769.1	2.507	0.346	2.853	3.02	0.167	528.3	80.0
Im6_50_6	211	3.5	655	0.51	1769.0	2.481	0.346	2.827	3.01	0.183	523.5	79.9
Im6_50_7	211	3.6	670	0.52	1768.6	2.561	0.346	2.907	3.07	0.163	538.1	80.3
Im6_50_8	211	3.6	677	0.52	1768.0	2.584	0.346	2.930	3.11	0.180	542.2	80.1
Im6_50_9	210	3.6	671	0.52	1768.6	2.567	0.346	2.913	3.09	0.177	539.3	80.4
								2.872		**0.173**		**80.0**

Table 5.14: F&W results motor 6 part 2.

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im6_75_0	211	4.1	992	0.66	1748.4	4.088	0.342	4.430	4.64	0.210	810.6	81.7
Im6_75_1	211	4.2	996	0.66	1748.0	4.100	0.342	4.442	4.66	0.218	812.6	81.6
Im6_75_2	211	4.1	987	0.66	1748.4	4.066	0.342	4.408	4.62	0.212	806.6	81.7
Im6_75_3	211	4.1	983	0.65	1749.0	4.036	0.342	4.378	4.61	0.232	801.4	81.5
Im6_75_4	210	4.1	981	0.65	1749.0	4.038	0.342	4.380	4.59	0.210	801.8	81.7
Im6_75_5	211	4.1	983	0.65	1749.0	4.044	0.342	4.386	4.60	0.214	802.9	81.7
Im6_75_6	211	4.1	984	0.65	1749.0	4.044	0.342	4.386	4.62	0.234	802.9	81.6
Im6_75_7	211	4.1	981	0.65	1749.0	4.027	0.342	4.369	4.58	0.211	799.8	81.5
Im6_75_8	210	4.1	984	0.65	1748.9	4.053	0.342	4.395	4.61	0.215	804.5	81.8
Im6_75_9	211	4.1	982	0.65	1749.0	4.043	0.342	4.385	4.59	0.205	802.7	81.7
								4.396		**0.216**		**81.7**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	F&W Trq	Pmech	Eff.
Im6_100_0	210	4.9	1337	0.75	1724.0	5.624	0.336	5.960	6.24	0.280	1075.5	80.4
Im6_100_1	210	4.9	1335	0.75	1724.0	5.610	0.336	5.946	6.25	0.304	1073.0	80.4
Im6_100_2	210	4.9	1336	0.75	1724.0	5.608	0.336	5.944	6.23	0.286	1072.6	80.3
Im6_100_3	210	4.9	1335	0.75	1724.0	5.611	0.336	5.947	6.24	0.293	1073.2	80.4
Im6_100_4	210	4.9	1335	0.75	1723.9	5.603	0.336	5.939	6.22	0.281	1071.7	80.3
Im6_100_5	210	4.9	1330	0.75	1724.0	5.594	0.336	5.930	6.19	0.260	1070.1	80.5
Im6_100_6	210	4.9	1329	0.75	1724.0	5.591	0.336	5.927	6.21	0.283	1069.6	80.5
Im6_100_7	211	4.9	1337	0.75	1723.6	5.639	0.336	5.975	6.23	0.255	1078.0	80.6
Im6_100_8	211	4.9	1330	0.75	1724.0	5.605	0.336	5.941	6.22	0.279	1072.1	80.6
Im6_100_9	211	4.9	1334	0.75	1723.3	5.624	0.336	5.960	6.24	0.280	1075.1	80.6
								5.947		**0.280**		**80.5**

Tables 5.3 through 5.14 show an extract of the results obtained in the test series which was performed with the goal of finding a sensible estimation of friction and windage losses. The most important results of the previous tables are further compressed in Table 5.15.

Table 5.15: Selection of most relevant F&W data for the tested motors.

Motor	Make	Design	hp	Poles	Load	pf	Speed [r/min]	F&W Trq [Nm]	F&W Pwr [W]	Eff
Motor 1	**GE**	**B**	**1**	**4**	25%	0.29	1789	0.244	45.7	79.8
					50%	0.48	1777	0.217	40.4	86.5
					75%	0.62	1762	0.252	46.5	85.9
					100%	0.72	1746	0.244	44.6	85.8
									44.3	
Motor 2	**GE**	**B**	**1**	**4**	25%	0.29	1789	0.176	33.0	77.5
					50%	0.48	1776	0.167	31.1	85
					75%	0.62	1761	0.206	38.0	85.8
					100%	0.72	1744	0.211	38.5	86.1
									35.1	
Motor 3	**GE**	**B**	**1**	**4**	25%	0.29	1789	0.247	46.3	80
					50%	0.47	1775	0.259	48.1	84.8
					75%	0.62	1759	0.261	48.1	86.1
					100%	0.71	1741	0.258	47.0	85.6
									47.4	
Motor 4	**Reliance**		**0.75**	**4**	25%	0.29	1787	0.541	101.2	72.2
					50%	0.51	1769	0.259	48.0	81.3
					75%	0.65	1749	0.275	50.4	82.1
					100%	0.76	1722	0.326	58.8	79.9
									64.6	
Motor 5	**Reliance**		**0.75**	**6**	25%	0.3	1188	0.278	34.6	73.7
					50%	0.47	1174	0.282	34.7	78.8
					75%	0.6	1158	0.299	36.3	79.2
					100%	0.69	1139	0.348	41.5	77
									36.8	
Motor 6	**Reliance**		**1.5**	**4**	25%	0.31	1786	0.184	34.4	76.8
					50%	0.51	1769	0.173	32.0	84.4
					75%	0.65	1749	0.216	39.6	84.7
					100%	0.75	1724	0.28	50.5	82.7
									39.1	

The friction, windage and stray load loss powers of Table 5.15 are plotted against percent load for all six motors in Figure 5.2.

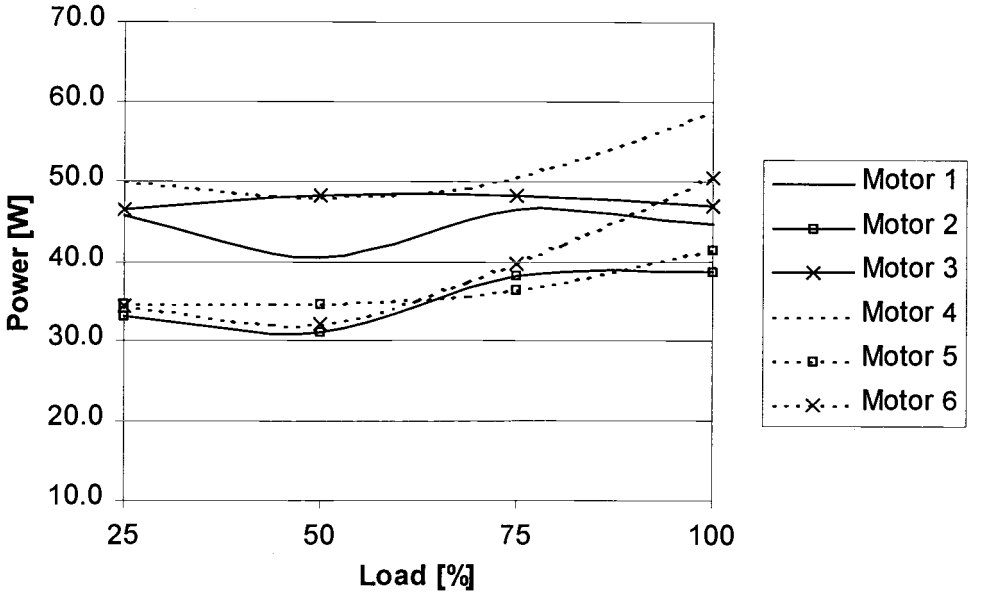


Figure 5.2: Friction, Windage and Stray Load torques for the tested motors.

The data represented in Fig. 5.2 shows a good correlation in relative amplitude, and also in shape. Based on the plot of Fig. 5.2 it has been decided that an adequate fitting of the six motor's data would be to run a linear regression on the average Friction, Windage and Stray Load Losses powers for each load point of the six motors. Then a linear regression of the resulting data can be performed, which can be used as the equation to look up estimation values for friction, windage and stray load losses. Table 5.16 shows the values for reaching the slope and y-intercept values of the friction, windage and stray load loss power estimation.

Table 5.16: Linear regression of Friction, Windage and Stray Load Loss Powers.

Load	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6	Average
25	45.7	33	46.3	49.8	34.6	34.4	**39.62**
50	40.4	31.1	48.1	48	34.7	32	**38.78**
75	46.5	38	48.1	50.4	36.3	39.6	**42.48**
100	44.6	38.5	47	58.8	41.5	50.5	**47.26**

slope	**0.10648**
y intercept	**35.38**

The results of Table 5.16 can be expressed in the following equation, which is implemented as the lookup formula for the estimated friction and windage losses for small motors:

$$Est.PowerLoss = 35.38 + 0.11 \cdot \frac{OperatingLoad}{\%Load} \quad \text{[Watts]} \qquad (5.1)$$

The tables 5.1 and 5.2 were used for building a linear regression of friction and windage torque (dynamometer correction torque) against the operating speed. The dynamometer correction torque was added to the torque measured at the torque transducer for each particular operating torque, resulting in the true output torque.

5.5 Resulting overall accuracies of prediction

A seventh motor was tested, using the friction and windage, as well as the stator resistance prediction methods that have been described. The seventh motor was tested after the prediction algorithms were incorporated into the efficiency estimation software, making bias impossible.

The seventh motor is of 1.5 hp rating, a 4-pole Reliance Design B motor, with a nameplate nominal efficiency of 78.5%. Tables 5.17 and 5.18 show the performance of motor 7, for an identical testing procedure.

Table 5.17: Results for Motor 7 part 1.

Motor 7

Name plate Data:							
Hp:	1.5	**Volts:**	208-230/460	**rpm:**	1730.00	**ph:**	3.00
Hz:	60	**code:**	J	**Frame:**	FC145T	**Design:**	B
Insulation:	F	**duty:**	const.	**s.f.:**	1.15	**Amps:**	5.6/2.8
Nom. Eff.:	78.5%	**Model:**		**Type:**	P		
MFD Model:		**MFG No:**	P14H1402PWZ	L-L Stator resistance = 3.8 ohms			

25% Load													Using rs estimator	
File name	Vave	Iave	Pel	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	%Err Trq	Pmech	Eff.	Est. Trq	Error Trq%
lm7_25_0	212	3.2	330	0.28	1787.1	0.91	0.35	1.26	1.20	4.92	236.3	71.6	1.23	2.86
lm7_25_1	212	3.2	341	0.29	1786.9	0.96	0.35	1.31	1.24	5.06	244.5	71.7	1.27	3.07
lm7_25_2	212	3.2	347	0.3	1786.8	0.98	0.35	1.33	1.29	3.09	249.1	71.8	1.32	1.13
lm7_25_3	212	3.2	346	0.3	1786.7	0.97	0.35	1.32	1.27	3.93	247.6	71.6	1.30	2.05
lm7_25_4	212	3.2	355	0.3	1786.0	1.02	0.35	1.37	1.33	2.99	256.1	72.1	1.35	1.16
lm7_25_5	212	3.2	350	0.3	1786.1	0.99	0.35	1.34	1.28	4.18	250.5	71.6	1.31	2.31
lm7_25_6	212	3.2	347	0.3	1786.4	0.99	0.35	1.34	1.27	4.65	249.6	71.9	1.30	2.70
lm7_25_7	212	3.2	349	0.3	1786.4	0.98	0.35	1.33	1.29	3.15	249.2	71.4	1.32	1.20
lm7_25_8	212	3.2	351	0.3	1786.0	1.03	0.35	1.38	1.30	5.43	257.8	73.4	1.33	3.55
lm7_25_9	212	3.2	342	0.29	1786.7	0.99	0.35	1.34	1.24	6.90	249.7	73.0	1.27	4.95
								1.33		**4.43**		**72.0**		**2.50**

Table 5.18: Results Motor 7 part 2.

50% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	%Err Trq	Pmech	Eff.	Est. Trq	Error Trq%
Im7_50_0	211	3.6	659	0.51	1770.0	2.56	0.35	2.91	2.84	2.32	538.3	81.7	2.87	1.12
Im7_50_1	211	3.8	643	0.5	1771.0	2.48	0.35	2.83	2.75	2.96	524.3	81.5	2.78	1.76
Im7_50_2	211	3.5	654	0.5	1770.3	2.53	0.35	2.88	2.78	3.46	533.5	81.6	2.82	2.24
Im7_50_3	211	3.5	649	0.5	1770.7	2.51	0.35	2.86	2.78	2.68	529.2	81.5	2.81	1.53
Im7_50_4	211	3.6	657	0.5	1770.0	2.54	0.35	2.89	2.81	2.68	535.3	81.5	2.85	1.50
Im7_50_5	211	3.5	643	0.5	1771.1	2.48	0.35	2.83	2.74	3.10	524.4	81.5	2.78	1.86
Im7_50_6	211	3.5	652	0.5	1770.3	2.52	0.35	2.87	2.78	3.22	531.9	81.6	2.81	2.04
Im7_50_7	211	3.6	662	0.51	1769.9	2.58	0.35	2.92	2.85	2.61	541.4	81.8	2.88	1.45
Im7_50_8	211	3.5	646	0.5	1771.0	2.50	0.35	2.84	2.75	3.43	527.1	81.6	2.78	2.20
Im7_50_9	211	3.6	667	0.51	1771.0	2.60	0.35	2.94	2.86	2.84	545.5	81.8	2.89	1.65
								2.88		**2.93**		**81.6**		**1.73**

75% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	%Err Trq	Pmech	Eff.	Est. Trq	Error Trq%
Im7_75_0	212	4.1	958	0.64	1752.0	3.97	0.34	4.31	4.26	1.26	791.0	82.6	4.30	0.24
Im7_75_1	212	4.1	971	0.64	1752.0	4.05	0.34	4.39	4.34	1.15	804.6	82.9	4.38	0.17
Im7_75_2	211	4.1	975	0.65	1751.0	4.08	0.34	4.42	4.33	2.06	810.3	83.1	4.37	1.07
Im7_75_3	211	4.1	955	0.64	1752.1	3.98	0.34	4.32	4.25	1.61	792.0	82.9	4.29	0.59
Im7_75_4	212	4.1	973	0.65	1751.7	4.06	0.34	4.40	4.33	1.76	807.2	83.0	4.37	0.78
Im7_75_5	212	4.1	967	0.64	1752.0	4.02	0.34	4.36	4.29	1.62	799.8	82.7	4.34	0.61
Im7_75_6	211	4.1	985	0.65	1750.0	4.07	0.34	4.42	4.37	0.95	808.9	82.1	4.42	0.07
Im7_75_7	211	4.1	977	0.65	1751.0	4.04	0.34	4.38	4.36	0.49	803.5	82.2	4.41	0.54
Im7_75_8	211	4.1	980	0.65	1750.7	4.07	0.34	4.41	4.34	1.50	807.9	82.4	4.39	0.48
Im7_75_9	211	4.1	977	0.65	1751.0	4.06	0.34	4.40	4.35	1.23	806.6	82.6	4.39	0.21
								4.38		**1.36**		**82.6**		**0.48**

100% Load

File name	Vave	Iave	Pel tot.	pf ave	Spd	Meas. Trq.	Dyn Corr.	Op. Trq.	Est. Trq.	%Err Trq	Pmech	Eff.	Est. Trq	Error Trq%
Im7_10_0	210	4.8	1323	0.75	1726.2	5.64	0.34	5.98	5.91	1.13	1080.4	81.7	5.98	0.02
Im7_10_1	210	4.8	1310	0.75	1727.7	5.56	0.34	5.90	5.84	0.94	1066.2	81.4	5.91	0.22
Im7_10_2	210	4.8	1313	0.75	1727.2	5.57	0.34	5.91	5.90	0.10	1067.9	81.3	5.97	1.05
Im7_10_3	210	4.8	1309	0.75	1728.0	5.56	0.34	5.89	5.87	0.50	1066.1	81.4	5.93	0.66
Im7_10_4	210	4.9	1325	0.75	1727.0	5.61	0.34	5.95	5.92	0.45	1075.3	81.2	5.99	0.71
Im7_10_5	210	4.8	1311	0.75	1728.0	5.57	0.34	5.91	5.86	0.73	1068.4	81.5	5.96	0.96
Im7_10_6	210	4.8	1315	0.75	1727.0	5.57	0.34	5.91	5.90	0.19	1068.5	81.3	5.97	0.96
Im7_10_7	210	4.8	1317	0.75	1726.9	5.58	0.34	5.91	5.89	0.35	1068.9	81.2	5.96	0.81
Im7_10_8	210	4.8	1304	0.74	1727.9	5.52	0.34	5.86	5.83	0.52	1059.8	81.3	5.90	0.63
Im7_10_9	210	4.8	1305	0.74	1728.0	5.52	0.34	5.86	5.84	0.38	1060.1	81.2	5.91	0.78
								5.91		**0.53**		**81.3**		**0.68**

The data printed in the previous 2 tables show two different sets of results. The two columns to the right print the results using the stator resistance estimator, while the former columns show the calculated torques resulting from correct stator resistance inputs.

Figure 5.3 shows the resulting errors as a function of torque for both cases, estimated and inputted stator resistance.

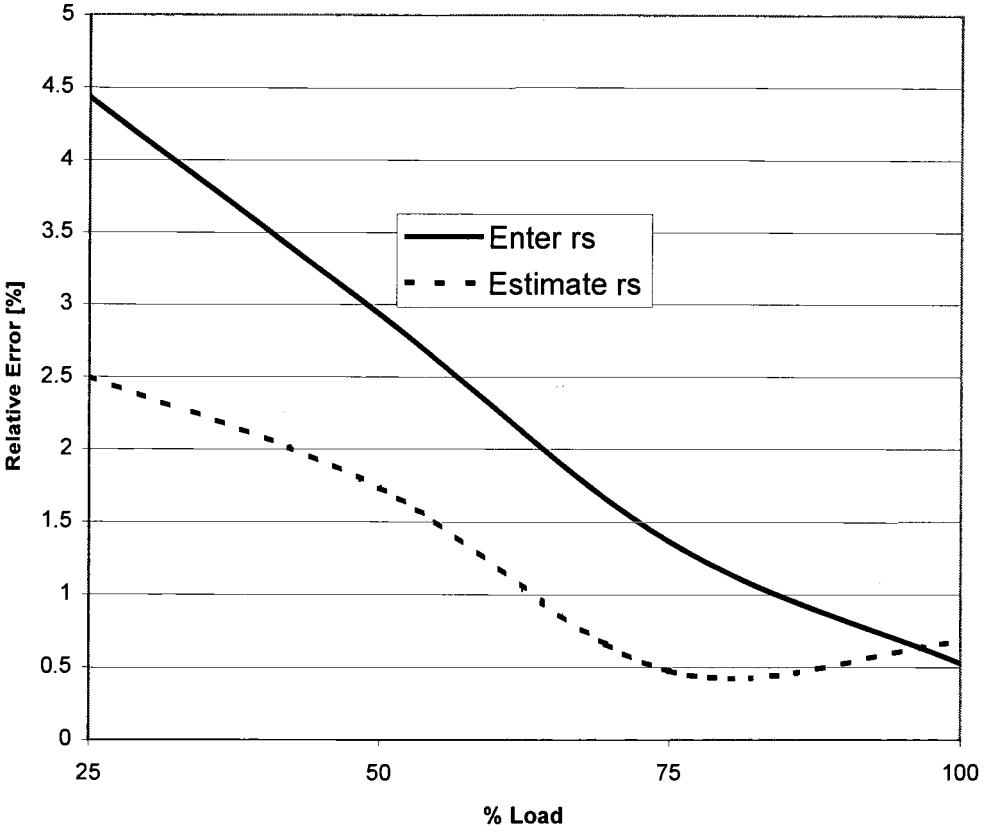


Figure 5.3: Errors committed by the Efficiency Estimator for Motor 7.

The results of Figure 5.3 are counter intuitive. Estimating rs resulted in smaller errors committed in the efficiency estimation than if the correct value for rs was entered. The explanation for this occurrence is that the errors resulting from stator resistance estimation had opposite sign than the other errors, hence they subtracted from each other.

It is also important to point out that common measurement sensors show a percentage of error with regard to top of scale, other than the relative errors shown here. Relative errors of roughly 4% at 25% of operating load would translate into 1% for top of the scale of the motor under test. Nevertheless, it is customary for evaluation of accuracy of efficiency measurements, to express measurement errors relative to the operating load, and not to rating of the machine under test.

6. Conclusions and Recommendations for Future Work

This work has presented a method to estimate the operating efficiency of IMs. The procedure for estimating the operating speed of the IM under test was based on results obtained from a detailed two axis model of an IM with dynamic eccentricity. The method of estimating the operating torque is rooted in the well known two-axis analysis of electrical machinery. The combination of predicted operating torque and speed permits calculating the mechanical output power, which, divided by the measured electrical input power, leads to efficiency prediction.

A proof of concept implementation of the suggested technique was developed and implemented. This involved hardware and software design, including the development of adequate digital signal processing techniques. The feasibility of the approach was proven on extensive testing performed on seven IMs, which were loaded to different operating points on a dynamometer. Their operating efficiency was computed by the measured torque and speed obtained from a torque and speed transducer. The results of comparing the estimated and measured operating efficiencies are very encouraging.

Future work will involve the development of a more accurate, robust, inexpensive and easy to manufacture prototype. The measured currents and voltages can be used to calculate further electrical quantities of interest in industrial settings. Typically plant maintenance personnel are interested in detailed operating voltage and current condition monitoring such as total harmonic distortion, harmonic bar chart representation, crest factor, voltage unbalance and symmetrical component information. By calculating the symmetrical components of the operating condition, it is possible to segregate the operating power factor into its displacement and its harmonic components.

Currently it is intended to include vibration sensing as a further tool allowing condition monitoring. Degrading rotor conditions will be identified using the conjunction of vibration signals, current spectral information and a history based data file.

References:

[1] NEMA, *'Motors and Generators'*, NEMA Standards Publication MG-1, Part 12, Page 23, Table 12-8.

[2] J. S. Hsu, J. D. Kueck, M. Olszewski, D. A. Casada, P. J. Otaduy, L. M. Tolbert, *'Comparison of Induction Motor Field Efficiency Evaluation Methods'*, IEEE IAS Proceedings, 1996, pp. 703.

[3] Wilfried Benning, *'Method of determining the efficiency of asynchronous motors and apparatus for carrying out the method'*, United States Patent #5,659,232.

[4] A. Ferrah, P. J. Hogben-Laing, K. J. Bradley, G. M. Asher, M. S. Woolfson, *'The Effect of Rotor Design on Sensorless Speed Estimation Using Rotor Slot Harmonics Identified by Adaptive Digital Filtering Using the maximum Likelihood Approach'*, IEEE IAS Proceedings, 1997, #3-5.

[5] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, *'Analysis of Electric Machinery'*, IEEE Press New York, ISBN 0-7803-1101-9.

[6] H. D. Lüke, *'Signalübertragung, Grundlagen der digitalen und analogen Nachrichtenübertragungssysteme'*, Springer Verlag Berlin, ISBN 3-540-19435-5.

[7] C. S. Burrus, *'Introduction to Wavelets and Wavelet Transforms'*, Prentice-Hall, ISBN 0-13-489600-9.

[8] G. B. Kliman, R. A. Koegl, W. J. Premerlani, *'A Demonstration of a Sensorless Torque Meter for AC Motors'*, IEEE IAS Proceedings, 1996, pp. 633.

[9] K. D. Hurst, T. G. Habetler, G. Griva, F. Profumo, *'Speed sensorless field-oriented control of induction machines using current harmonic spectral estimation'*, IEEE IAS proceedings 1994 vol. I, pp. 601.

[10] K. D. Hurst, T. G. Habetler, *'A comparison of spectrum estimation techniques for sensorless speed detection in induction machines'*, IEEE IAS proceedings 1995 vol. I, pp. 553.

[11] J. F. Brundy, D. Roger, *'Induction machine speed seonsor based on stator current measurement'*, Power electronics and variable speed drives, Conference publication IEE 1996.

[12] A. Babour, W. T. Thomson, *'Finite element study of rotor slot designs with respect to current monitoring for detecting static airgap eccentricity in squirrel cage induction motors'*, IEEE IAS proceedings 1997, session # 3-3.

[13] M. Brandford, *'Unbalanced magnetic pull in a 6-pole induction motor'*, IEE Proceedings 1968, Vol. 115 No. 11, pp. 1619.

[14] D. G. Dorrell, *'Calculation of unbalanced magnetic pull in cage induction machines'*, Ph.D. thesis 1993, University of Camebridge.

[15] D. G. Dorrell, W. T. Thomson, S. Roach, *'Combined effects of static and dynamic eccentricity on airgap flux waves and the application of current monitoring to detect dynamic eccentricity in 3-phase induction motors'*, Electrical machines and drives, IEE 1995.

[16] D. G. Dorrell, W. T. Thomson, S. Roach, *'Analysis of airgap flux, current and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors'*, IEEE IAS proceedings 1995, vol. I., pp. 563.

[17] P. Pillay, Z. Xu, *'Motor current signature analysis'*, IEEE IAS proceedings 1996, vol. I., pp. 587.

[18] A. Wallace, A. von Jouanne, E. Wiedenbrüg, J. Douglass, C. Wohlgemuth, G. Wainwright, *'A Laboratory Assessment of In-Service Motor Efficiency Testing Methods'*, IEEE IEMDC Proceedings 1997, WC 1.7-1.

[19] John G. Douglass, *'Efficacy of Methods for Estimating In-Service Motor Efficiency'*, Washington State University, Cooperative Extension Energy Program, June 1997.

[20] EASA Technical Manual, © 1996, pp. 8-15, 8-16.

[21] NEMA *'Motors and Generators'*, MG 1, Part 14, page 8.

[22] IEEE, *'Standard Test Procedure for Polyphase Induction Motors and Generators'*, IEEE Std. 112.

[23] C. Shauder, *'Adaptive speed identification for vector control of induction motors without rotational transducers'*, IEEE IAS Proceedigns 1989, pp. 493.

[24] H. Tajima, Y. Hori, *'Speed sensorless field-orientation control of the induction moachine'*, IEEE IAS Transactions vol. 29, pp. 175.

[25] P. Jansen, R. D. Lorenz, *'A physically insightful approach to the design and accuracy assessment of flux observers for field oriented machine drivers'*, IEEE IAS Transactions, vol. 30, pp. 312.

[26] D. S. Zinger, T. A. Lipo, D. W. Novotny, *'Using induction motor stator windings to extract speed information'*, IEEE IAS Proceedings 1989, pp. 213.

[27] R. M. Cuzner, R. D. Lorenz, D. W. Novotny, *'Application of non linear observers for rotor position detection on an induction motor using machine voltages and currents'*, IEEE IAS Proceedings, 1990, pp. 416.

[28] A. Ferrah, K. G. Bradley, G. M. Asher, *'Sensorless speed detection of inverter fed induction motors using rotor slot harmonics and fast fourier transform'*, IEEE PESC Proceedings 1992, pp. 280.

[29] K. J. Binns, D. W. Shimmin, K. M. Al-Aubidy, *'Implicit rotor-position sensing using motor windings for a self-commutating permanent-magnet drive system'*, IEE Proceedings, vol. 138, pt. B, no. 1, pp. 28.

[30] M. Schroedl, *'Operation of the mermanent magnet synchronous machine without a mechanical sensor'*, IEE Proceedings ICPEVSD 1990, pp. 51.

[31] M. Schroedl, *'Sensorless contorl of induction motors at low speed and standstill'*, IEEE ICEM Proceedings 1990, pp. 863.

[32] M. Shroedl, R. S. Wieser, *'Induction motor drive for in locomotives'*, IEE 5th EPE Conference 1995, vol. 3, pp. 62.

[33] M. Shroedl, *'Sensorless control of AC machines at low speed and standstill based on the INFORM method'*, IEEE IAS Proceedings, 1996, pp. 270.

[34] P. L. Jansen, R. D. Lorenz, *'Transducerless position and velocity estimation in induction and salient AC machines'*, IEEE IAS Transactions, 1995, vol. 31, pp. 240.

[35] P. L. Jansen, M. J. Corley, R. D. Lorenz, *'Flux, position and velocity estimation in AC machines at zero speed via tracking of hich frequency saliencies'*, IEE 5th EPE Conference 1995, vol. 3, pp. 154.

[36] P. L. Jansen, R. D. Lorenz, *'Transducerless field orientation concepts employing saturation-induced saliencies in induction machines'*, IEEE IAS Transactions, vol. 32, pp. 1380.

[37] J. Cilia, G. M. Asher, K. Bradley, *'Sensorless position detection for vector controlled induction motor drives using an asymmetric outer-section cage'*, IEEE IAS Proceedings 1996, pp. 286.

[38] M. W. Degner, R. D. Lorenz, *'Position estimation in induction machines utilizing rotor bar slot harmonics and carrier frequency signal injection'*, Proceedings, Power Conversion Conference, Nagaoka, Japan, 1997, pp. 69.

[39] M. W. Degner, R. D. Lorenz, *'Using Multiple Saliencies for the Estimation of Flux, Position, and Velocity in AC Machines'*, IEEE IAS Transactions 1998, vol. 34, no. 5 pp. 1097.

[40] A. B. Plunkett, *'Torque regulating induction motor system'*, United States Patent 4,023,083, May 10th 1977.

[41] M. Matsumoto, K. Kikubunji, *'Torque control systems of induction Motors'*, United States Patent, 4,384,244, May 17th 1983.

[42] S. Smith, K. N. Castleberry, *'Method and Apparatus for monitoring machine performance'*, United States Patent, 5,523, 701, June 4th 1996.

[43] M. D. Hicho, *'Method and Apparatus for Analyzing Rotating Machines'*, United States Patent, 5,109,700, May 5th 1992.

[44] W. Benning, *'Method of determining the efficiency of asynchronous motors and apparatus for carrying out the method'*, United States Patent, 5,659,232, August 19th 1997.

[45] G. Bolegoh, *Method and Apparatus for measuring electric motor efficiency and Loading'*, United States Patent, 5,262,717, November 16th, 1993.

[46] G. F. Lang, D. H. Heagerty, G. R. Kahley, *'Method and Apparatus for determining mechanical performance of polyphase electrical motor systems'*, United States Patent, 5,521,482, May 28th, 1996.

[47] J. D. Kueck, P. J. Otaduy, *'Method for assessing in-service motor efficiency and in-service motor/load efficiency'*, United States Patent, 5,661,386, August 26th 1997.

[48] B. A. Elfner, A. P. P. Comstedt, *'Methods and Apparatuses for detecting the speed of an asynchronous motor'*, United States Patent, 4,358,734, November 9th, 1982.

[49] G. B. Kliman, *'Spectral Analysis of induction motor current to detect rotor faults with reduced false alarms'*, United States Patent, 5,049,815, September 17th 1991.

[50] G. B. Kliman, R. A. A. Koegl, M. W. Schulz Jr., S. E. Grabkowski, *'Rotor Fault Detector for Induction Motors'*, United States Patent 4,761,703, August 2nd, 1988.

Appendices

Appendix A:

The Matlab scripts below are the program used to simulate the machine model that was presented in chapter 2. The function im2.m is called from the main program, Im2ph.m, by the ODE solving suite.

B.1 Function im2.m

```
function xdot = im2(t,x, flag, Ks, Ks1, P, A, phi, omspeed, omelect,
airgapA0, airgapB0, airgapC0, k, Lm, rs, Ls, Lr, rr, Rld, twopi);

rr = rr+Rld;
%Creating the voltages vq and vd and the v vector
vabc = [265*sqrt(2)*cos(omelect*t)
   265*sqrt(2)*cos(omelect*t-2*pi/3)
   265*sqrt(2)*cos(omelect*t+2*pi/3)];
vqd = Ks*vabc;

v(1,1)=vqd(1,1);
v(2,1)=vqd(2,1);
v(3,1)=0;
v(4,1)=0;

%Calculating the Lma, Lmb, Lmc, pLma, pLmb, pLmc as a function of time,
A, phi, omspeed, p
Lma = 0;
Lmb = 0;
Lmc = 0;
Lsa = 0;
Lsb = 0;
Lsc = 0;
Lra = 0;
Lrb = 0;
Lrc = 0;
pLma = 0;
pLmb = 0;
pLmc = 0;
pLsa = 0;
pLsb = 0;
pLsc = 0;
pLra = 0;
```

```
pLrb = 0;
pLrc = 0;
tmpa = 0;
tmpb = 0;
tmpc = 0;
ptmpa = 0;
ptmpb = 0;
ptmpc = 0;
for u = 1 : P                %check whether this should be 1:P instead.
   for i = 1 : max(size(A))
      tmpa = tmpa + A(i)*cos(omspeed*t*i + phi(i)*i/P + u*i*2*pi/P);
      tmpb = tmpb + A(i)*cos(omspeed*t*i + phi(i)*i/P +2*pi*i*(u/P-
1/3));
      tmpc = tmpc + A(i)*cos(omspeed*t*i + phi(i)*i/P
+2*pi*i*(u/P+1/3));
      ptmpa = ptmpa + A(i)*omspeed*i * sin(omspeed*t*i + phi(i)*i +
u*i*2*pi/P);
      ptmpb = ptmpb + A(i)*omspeed*i * sin(omspeed*t*i + phi(i)*i +
2*pi*i*(u/P-1/3));
      ptmpc = ptmpc + A(i)*omspeed*i * sin(omspeed*t*i + phi(i)*i +
+2*pi*i*(u/P+1/3));
   end
   Lma = Lma + airgapA0+tmpa;
   Lmb = Lmb + airgapB0+tmpb;
   Lmc = Lmc + airgapC0+tmpc;
   Lsa = Lsa + 1/(airgapA0+tmpa);
   Lsb = Lsb + 1/(airgapB0+tmpb);
   Lsc = Lsc + 1/(airgapC0+tmpc);
   pLma = pLma + ptmpa;
   pLmb = pLmb + ptmpb;
   pLmc = pLmc + ptmpc;
   pLsa = pLsa + ptmpa/((airgapA0+tmpa)*(airgapA0+tmpa));
   pLsb = pLsb + ptmpb/((airgapB0+tmpb)*(airgapB0+tmpb));
   pLsc = pLsc + ptmpc/((airgapC0+tmpc)*(airgapC0+tmpc));
end
Lsa = Ls * Lsa/P;
Lsb = Ls * Lsb/P;
Lsc = Ls * Lsc/P;
Lra = Lr * Lsa/P;
Lrb = Lr * Lsb/P;
Lrc = Lr * Lsc/P;
Lma = Lm * Lma/P;
Lmb = Lm * Lmb/P;
Lmc = Lm * Lmc/P;
pLma = pLma * Lm/P;
pLmb = pLmb * Lm/P;
pLmc = pLmc * Lm/P;
pLsa = pLsa * Ls/P;
pLsb = pLsb * Ls/P;
pLsc = pLsc * Ls/P;
pLra = pLsa * Lr/P;
pLrb = pLsb * Lr/P;
pLrc = pLsc * Lr/P;

%Defining Lmabc matrix, which includes the phase-phase stator mutuals.
Lmabc(1,:) = [Lma -k*sqrt(Lma*Lmb) -k*sqrt(Lma*Lmc)];
Lmabc(2,:) = [Lmabc(1,2) Lmb -k*sqrt(Lmb*Lmc)];
```

```
Lmabc(3,:) = [Lmabc(1,3) Lmabc(2,3) Lmc];

%Defining the pLmabc matrix, including the stator phase-phase mutuals
pLmabc(1,:) = [pLma   -k*(pLma*Lmb+pLmb*Lma)/(2*sqrt(Lma*Lmb))   -
k*(pLma*Lmc+pLmc*Lma)/(2*sqrt(Lma*Lmc))];
pLmabc(2,:) = [pLmabc(1,2)   pLmb   -
k*(pLmb*Lmc+pLmc*Lmb)/(2*sqrt(Lmb*Lmc))];
pLmabc(3,:) = [pLmabc(1,3)   pLmabc(2,3)   pLmc];

%Defining the Lsabc matrix
Lsabc(1,:) = [Lsa 0 0];
Lsabc(2,:) = [0 Lsb 0];
Lsabc(3,:) = [0 0 Lsc];

%Defining the Lrabc matrix
Lrabc(1,:) = [Lra 0 0];
Lrabc(2,:) = [0 Lrb 0];
Lrabc(3,:) = [0 0 Lrc];

%Defining the pLsabc matrix
pLsabc(1,:) = [pLsa 0 0];
pLsabc(2,:) = [0 pLsb 0];
pLsabc(3,:) = [0 0 pLsc];

%Defining the pLrabc matrix
pLrabc(1,:) = [pLra 0 0];
pLrabc(2,:) = [0 pLrb 0];
pLrabc(3,:) = [0 0 pLrc];

%Translating the Lmabc and pLmabc into dq0, making L and pL
L  = Ks1 * Lmabc  * Ks;
pL = Ks1 * pLmabc * Ks;

%Translating the Lsabc, Lrabc, pLsabc and pLrabc matrices into dq0
making Lsdq Lrdq pLsdq and pLrdq
Lsdq = Ks1 * Lsabc * Ks;
Lrdq = Ks1 * Lrabc * Ks;
pLsdq = Ks1 * pLsabc * Ks;
pLrdq = Ks1 * pLrabc * Ks;

%Elements to build matrices matA and matB
Lq = L(1,1);
Ld = L(2,2);
M1 = L(1,2);
M2 = L(2,1);
pLd = pL(1,1);
pLq = pL(2,2);
pM1 = pL(1,2);
pM2 = pL(2,1);
Ls11 = Lsdq(1,1);
Ls12 = Lsdq(1,2);
Ls21 = Lsdq(2,1);
Ls22 = Lsdq(2,2);
Lr11 = Lrdq(1,1);
Lr12 = Lrdq(1,2);
Lr21 = Lrdq(2,1);
Lr22 = Lrdq(2,2);
```

```
pLs11 = pLsdq(1,1);
pLs12 = pLsdq(1,2);
pLs21 = pLsdq(2,1);
pLs22 = pLsdq(2,2);
pLr11 = pLrdq(1,1);
pLr12 = pLrdq(1,2);
pLr21 = pLrdq(2,1);
pLr22 = pLrdq(2,2);

%Building matA
matA(1,:)=[Lq+Ls11    M1+Ls12    Lq    M1];
matA(2,:)=[M2+Ls21    Ld+Ls22    M2    Ld];
matA(3,:)=[Lq    M1    Lq+Lr11    M1+Lr12];
matA(4,:)=[M2    Ld    M2+Lr21    Ld+Lr22];

%Building matB
matB(1,:)=[rs+pLq+pLs11     pM1+pLs12     pLq     pM1];
matB(2,:)=[pM2+pLs21     rs+pLd+pLs22     pM2     pLd];
matB(3,:)=[pLq     pM1-omspeed*Lq     rr+pLq+pLr11     pM1-
omspeed*(Ld+Lr)+pLr12];
matB(4,:)=[pM2+omspeed*Lq     pLd     pM2+omspeed*(Lq+Lr)+pLr21
rr+pLd+pLr22];

% and now solving for xdot:
xdot=inv(matA)*(-matB*x+v);
```

A.2 Matlab Function Im3ph.m

```
tstep = .0005;
tend = 15;
TSPAN = [0:tstep:tend];
f=[0:1/tend:1/tstep];
twopi = 2*pi;

%Declaring the matrices Ks and Ks1 (to transform to and from the dq
domain).
Ks(1,:)=[1 cos(-2*pi/3) cos(2*pi/3)];
Ks(2,:)=[0 sin(-2*pi/3) sin(2*pi/3)];
Ks(3,:)=[.5 .5 .5];
Ks = 2*Ks/3;

Ks1(1,:)=[1 0 1];
Ks1(2,:)=[cos(-2*pi/3) sin(-2*pi/3) 1];
Ks1(3,:)=[cos(2*pi/3) sin(2*pi/3) 1];

%Declaring the constants (eq. circuit parameters)
P   = 3;                    %polepairs
Rs  = 0.87;                 %stator resistance (ohms)
Rr  = 0.15;                 %rotor resistance  (ohms)
Rld = 5;                    %load resistance   (ohms)
Ls  = 0.802e-3;             %stator leakage inductance (H)
Lr  = .4*0.802e-3;          %rotor leakage inductance  (H)
```

```
Lm   = 34.7e-3;                %medium mutual inductance   (H)

%Calculating speed
omelect  = 2*pi*60;
speed    = 3600/P*(Rld/(Rld+Rr));  % (rpm)
speedHz = speed/60                 % speed in Herz
slipHz   = 60/P-speed/60;          % slip in Herz
omspeed = 2*pi*speed/60;           % (rad/s)

%Defining the vector that describes the airgap irregularities.
airgapA0 = 1;       %these 3 factors should be
airgapB0 = 1;       %1*relative deviation of
airgapC0 = 1;       %mean arigap (accounting for static eccentricity)
A    = [.03];        %absolute values of nth fourier component
phi = [0];          %angle displacement of nth component
pitch = 60;         %winding pitch in degrees.  Standard = 120.
if pitch>60/P
   k = (pitch-(60/P))/pitch;  %k = coupling factor for the phase-phase
mutual inductances
else
   k = 0;
end
options = odeset('AbsTol', 5e-5, 'RelTol', 3e-3);

[t,x] = ode23('im2', TSPAN, [0 0 0 0]', options, Ks, Ks1, P, A, phi,
omspeed, omelect, airgapA0, airgapB0, airgapC0, k, Lm, Rs, Ls, Lr, Rr,
Rld, twopi);

vq = 265*sqrt(2)*cos(omelect*t);
vd = 265*sqrt(2)*sin(omelect*t);
I = sqrt(x(:,1).*x(:,1)+x(:,2).*x(:,2));

hold off
figure(1)
fftim = abs(fft(x(:,1)))';
semilogy(f,fftim,'k');
axis([0 125 min(fftim) max(fftim)]);
title('Iqs');

figure(5)
semilogy(f,fftim,'k');
axis([28 32 min(fftim) max(fftim)]);
title('Iqs');

figure(9)
tmp9 = abs(fft(x(:,2)))';
semilogy(f,tmp9,'k');
axis([0 125 min(tmp9) max(tmp9)]);
title('Ids');

figure(10)
semilogy(f,tmp9,'k');
axis([0 125 min(tmp9) max(tmp9)]);
title('Ids');

figure(2)
fftI = abs(fft(I));
```

```
semilogy(f,fftI,'k');
axis([0 125 min(fftI) max(fftI)]);
title('I vector');

figure(6)
semilogy(f,fftI,'k');
axis([28 32 min(fftI) max(fftI)]);
title('I vector');

figure(3)
pd = vd .* x(:,2);
tmp2 = abs(fft(pd));
semilogy(f,tmp2,'k');
title('pd');
axis([0 125 min(tmp2) max(tmp2)]);

figure(4)
semilogy(f,tmp2,'k');
axis([28 32 min(tmp2) max(tmp2)]);
title('pd');
axis([28 32 min(tmp2) max(tmp2)]);

figure(7)
qd = vq .* x(:,2);
tmp3 = abs(fft(qd));
semilogy(f,tmp3,'k');
title('qd');
axis([0 125 min(tmp3) max(tmp3)]);

figure(8)
semilogy(f,tmp3,'k');
title('qd');
axis([28 32 min(tmp3) max(tmp3)]);
```

Appendix B:

Appendix C shows the most important panels that have been developed and used for speed, torque, and efficiency prediction. The printed panels are not complete, since related families of panels exist for the different channels.

B.1 Main Panel:

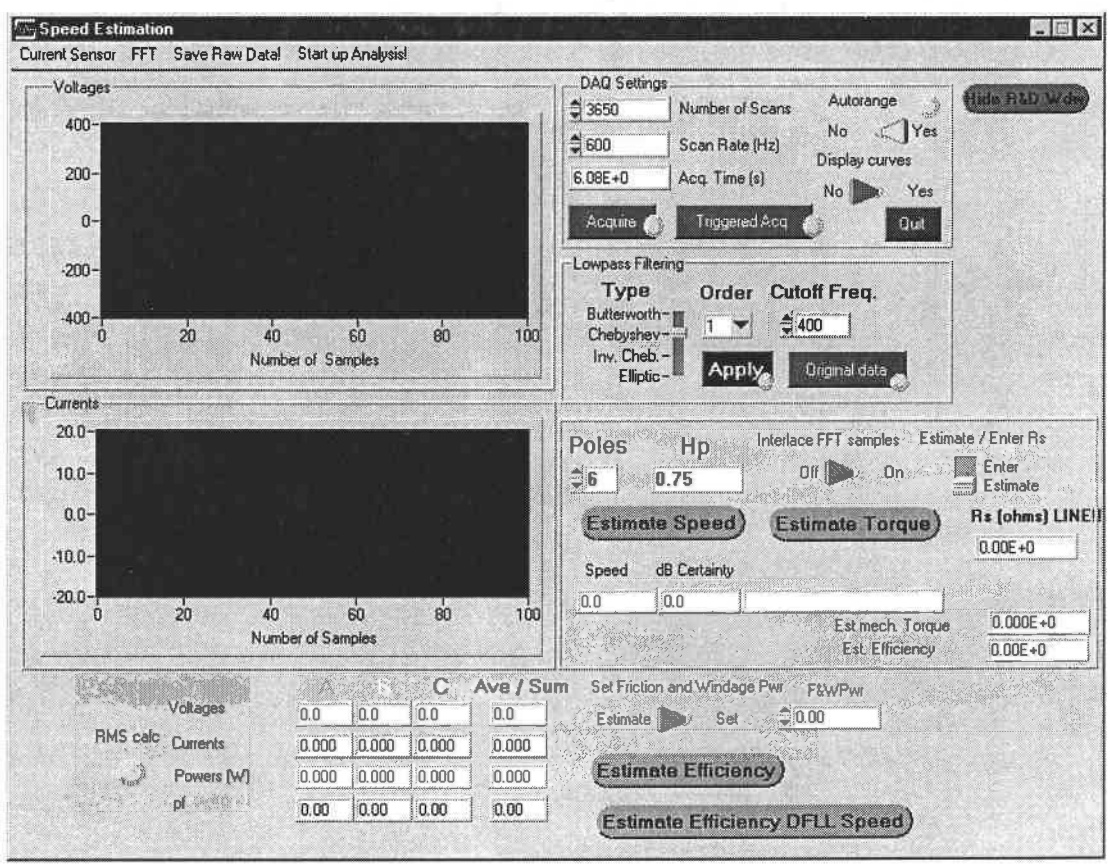


Figure B.1: Main Panel

B.2 Current FFT Panel:

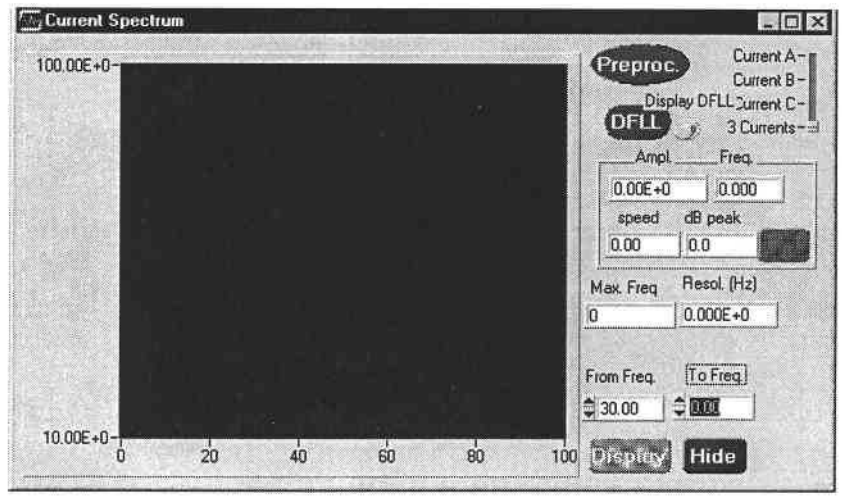


Figure B.2: Current FFT Panel

B.3 Current DFLL Panel:

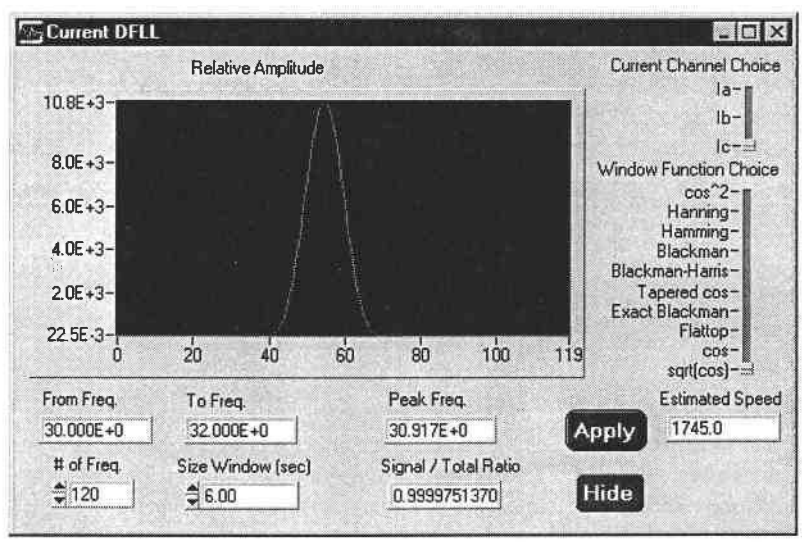


Figure B.3: Current DFLL Panel

B.4 Torque, Id and Iq Panel:

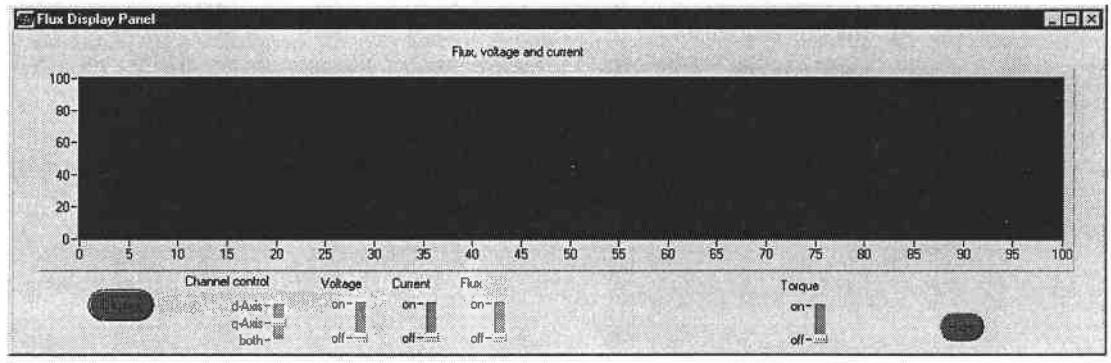


Figure B.4: Torque, Id and Iq Panel.

B.5 Impedance FFT window:

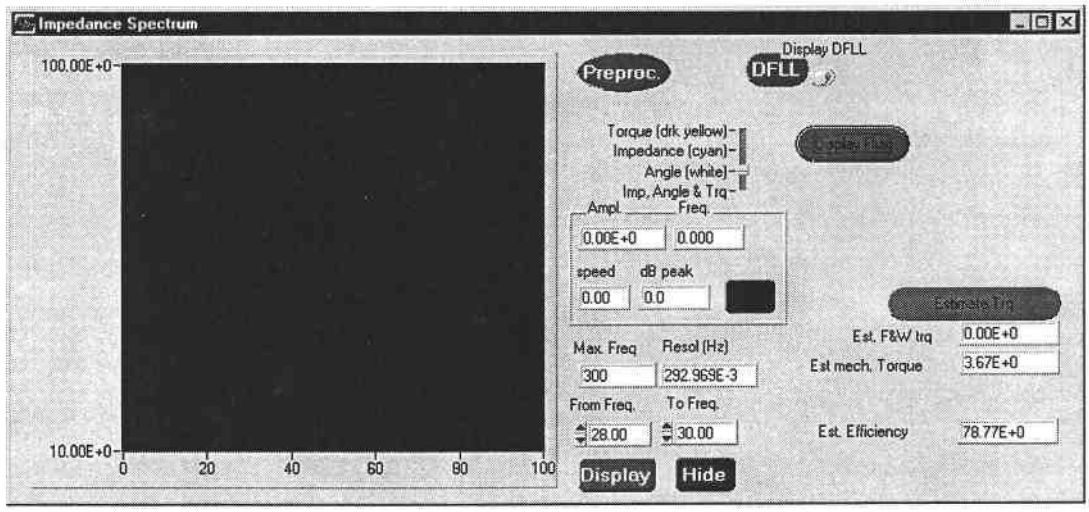


Figure B.5: Impedance FFT window.

Appendix C

C.1 CVI program code of efficiency estimation program:

```
static double       ratedpwr;
static int          fluxdisplayFluxOnOff;
static int          fluxdisplayCurrentOnOff;
static int          fluxdisplayVoltageOnOff;
static int          fluxdisplay;
static double       dfllreqAccuracy;
static int          dfllcurrchan;
static int          dfllvoltchan;
static int          dfllwdwpct;
static int          dfllnumfreq = 50;
static int          dfllNumSamp;
static double       dfllmaxfreq;
static double       dfllminfreq;
static double       dfllmaxamp;
static int          dfllmaxSamp;
static double       dfllminamp;
static double       dfllpeakfreq;
static double       dflldeltafreq;
static double       dfllsmallfreq;
static int          dfllaposterioriCnt = 0;
static double       dfllamp[200];
static double       dfllang[200];
static int          dfllNumPeriods;
#include <analysis.h>
#include <ansi_c.h>
#include <userint.h>
#include <utility.h>
#include <math.h>
#include "speed.h"
#include "easyio.h"
static int      panelHandle;
static int          currentturnHandle;
static int          currentFFTHandle;
static int          voltageFFTHandle;
static int          powerFFTHandle;
static int          impedanceFFTHandle;
static int          FFTWarningHandle;
static int          saveDataHandle;
static int          currentDFLLHandle;
static int          impedDFLLHandle;
static int          voltageDFLLHandle;
static int          fluxdisplayHandle;
static int          dftspeedestimateHandle;
static double       currentFFTstartfreq, voltageFFTstartfreq, powerFFTstartfreq, impedanceFFTstartfreq;
static double       currentFFTendfreq, voltageFFTendfreq, powerFFTendfreq, impedanceFFTendfreq;
static int          currentFFTslide, voltageFFTslide, powerFFTslide, impedanceFFTslide;
static int          currentstartFFTsample, voltagestartFFTsample, powerstartFFTsample, impedancestartFFTsample;
static int          currentendFFTsample, voltageendFFTsample, powerendFFTsample, impedanceendFFTsample;
static int          numFFT;
static int          flagFreeCurrent = 0, flagFreeVoltage =0, flagFreePower = 0, flagFreeImpedance = 0;
static int          dfllcurrentflag = 0;
static int          dfllvoltageflag = 0;
static int          dfllimpedflag   = 0;
static int          i, j;
static int          spectrumlength;
static int          currentturns = 20;
static short        device;
static unsigned long            numChannels;
static unsigned long            numScans;
```

```
static char          channelString[100];
static double        tmp1, tmp2, tmp3;
static double        tmp11[15];
static int           tmp;
static double        speedEstimate = 0.0;
static int           maxcnt;
static int           cnt[15];
static double        wdw = 2.5;      /*speed estimates of within +-wdw will be considered. (rpm)*/
static double      tmp4[200];
static double      actualRate;
static double      upper;
static double      upper2 = 5.0;
static double      lower;
static double      lower2 = -5.0;
static double      rate;
static double      * waveforms = 0;
static double      * waveformsOrig =0;
static double      * currentA = 0, * currentB = 0, * currentC = 0;
static double      * currentAf = 0, * currentBf = 0, * currentCf = 0;
static double      * voltageA = 0, * voltageB = 0, * voltageC = 0;
static double      * voltageAf = 0, * voltageBf = 0, * voltageCf = 0;
static double      * powerA = 0,   * powerB = 0,   * powerC = 0,   *powersum = 0;
static double      * powerAf = 0,   * powerBf = 0,   * powerCf = 0,   *powersumf = 0;
static double      * impedance = 0, * impedance2 = 0, * angle = 0, * angle2 = 0, * vd = 0, * vq = 0, * id = 0, * iq = 0, *
                                lambdad = 0, * lambdaq = 0;
static double        * impedancef = 0, * anglef = 0, * trqmech = 0;
static double        rs = 0;
static double        forgetfactor = .9995;
static double        fricpwr;
static double        nest = 1780.0;
static double        trqFW;
static int           Pp = 2;
static double        trqmechave;
static double        trqest;
static double        * powers = 0;
static double        * powersOrig = 0;
static double        pi = 3.1415926;
static double        twopi = 6.2831852;
static double        acqtime = 0.2;
static int           filterorder;
static double        filtercutoff;
static int           filtertype;
static double      Varms, Vbrms, Vcrms, Vaverms;
static double        Iarms, Ibrms, Icrms, Iaverms;
static double        Parms, Pbrms, Pcrms, Ptotrms;
static double        Vagain = 502.5;
static double        Vbgain = 497.5;
static double        Vcgain = 499.9;
static double        Iagain = 8*3.0644;
static double        Ibgain = 8*3.0718;
static double        Icgain = 8*3.072;
static char          SaveFilename[40];
static char          SaveSerialNumber[40];
static char          SaveMakeHp[40];
static int           SavePoles;
static double        SaveOperatingSpeed;
static double        SaveMechPower;
FILE                 *outFilePointer;
static int ColorArray[8] = {VAL_GREEN, VAL_YELLOW, VAL_BLUE, VAL_GREEN, VAL_BLUE, VAL_MAGENTA,
VAL_GRAY, VAL_RED};



void main ()
{
   panelHandle = LoadPanel (0, "speed.uir", PANEL);
     currentturnHandle  = LoadPanel (0, "speed.uir", CURRENTTUR);
     currentFFTHandle   = LoadPanel (0, "speed.uir", CURRENTFFT);
     FFTWarningHandle   = LoadPanel (0, "speed.uir", FFTWARNING);
```

```
    voltageFFTHandle   = LoadPanel (0, "speed.uir", VOLTAGEFFT);
    powerFFTHandle     = LoadPanel (0, "speed.uir", POWERFFT) ;
    impedanceFFTHandle = LoadPanel (0, "speed.uir", IMPEDFFT);
    saveDataHandle     = LoadPanel (0, "speed.uir", SAVEDATA);
    currentDFLLHandle  = LoadPanel (0, "speed.uir", CURRDFLL);
    impedDFLLHandle    = LoadPanel (0, "speed.uir", IMPEDDFLL);
    voltageDFLLHandle  = LoadPanel (0, "speed.uir", VOLTDFLL);
    fluxdisplayHandle = LoadPanel (0, "speed.uir", FLUXDISPLA);
    dftspeedestimateHandle = LoadPanel (0, "speed.uir", DFTESTIMAT);
    LoadMenuBar (panelHandle, "speed.uir", MenuBar1);
    RateCallback (panelHandle, PANEL_RATE, EVENT_COMMIT, 0, 0, 0);
  SetBreakOnLibraryErrors (0);
  NumScansCallback (panelHandle, PANEL_NUMSCANS, EVENT_COMMIT, 0, 0, 0);
  DeviceCallback (panelHandle, PANEL_DEVICE, EVENT_COMMIT, 0, 0, 0);
  ChannelCallback (panelHandle, PANEL_CHANNEL, EVENT_COMMIT, 0, 0, 0);
  SetBreakOnLibraryErrors (1);
  DisplayPanel (panelHandle);
  RunUserInterface ();
}

int CVICALLBACK DeviceCallback (int panel, int control, int event,
      void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      GetCtrlVal (panelHandle, PANEL_DEVICE, &device);
      ChannelCallback (panelHandle, PANEL_CHANNEL, EVENT_COMMIT, 0, 0, 0);
    break;
  }
  return 0;
}

int CVICALLBACK ChannelCallback (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_RIGHT_CLICK:
    break;
    case EVENT_COMMIT:
      GetCtrlVal (panelHandle, PANEL_CHANNEL, channelString);
      GetNumChannels (device, channelString, ANALOG_INPUT, &numChannels);
      free (waveforms);
      free (waveformsOrig);
      free (powers);
      free (powersOrig);
      waveforms = (double *) malloc (numScans * numChannels * sizeof (double));
      waveformsOrig = (double *) malloc (numScans * numChannels * sizeof (double));
      powers = (double *) malloc (numScans * 6 * sizeof (double));
      powersOrig = (double *) malloc (numScans * 6 * sizeof (double));
    break;
    }
  return 0;
}

int CVICALLBACK ChannelRingCallback (int panel, int control, int event,
                        void *callbackData, int eventData1, int eventData2)
{
  char  * str;
  switch (event) {
  static int index;
  static int length;
  case EVENT_COMMIT:
  GetCtrlIndex (panelHandle, PANEL_CHANNELRING, &index);
  GetValueLengthFromIndex (panelHandle, PANEL_CHANNELRING, index, &length);
  str = (char *) malloc (length+1);
  GetCtrlVal (panelHandle, PANEL_CHANNELRING, str);
  SetCtrlVal (panelHandle, PANEL_CHANNEL, str);
  ChannelCallback (panelHandle, PANEL_CHANNEL, EVENT_COMMIT, 0, 0, 0);
  free (str);
```

```
   break;
   }
   return 0;
}

int CVICALLBACK RateCallback (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
   switch (event)
   {
     case EVENT_COMMIT:
       GetCtrlVal (panelHandle, PANEL_RATE, &rate);
     GetCtrlVal (panelHandle, PANEL_NUMSCANS, &numScans);
     acqtime = numScans / rate;
     SetCtrlVal (panelHandle, PANEL_ACQTIME, acqtime);
     SetAxisRange (panelHandle, PANEL_GRAPH, VAL_AUTOSCALE, 0.0,acqtime*1000, VAL_NO_CHANGE, 0.0, 1.0);
     SetAxisRange (panelHandle, PANEL_GRAPH_2,VAL_AUTOSCALE,0.0,acqtime*1000, VAL_AUTOSCALE, 0.0, 1.0);
  break;
  }
  return 0;
}

int CVICALLBACK NumScansCallback (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
  switch (event) {
  case EVENT_COMMIT:
     GetCtrlVal (panelHandle, PANEL_NUMSCANS, &numScans);
     GetCtrlVal (panelHandle, PANEL_RATE, &rate);
     acqtime = numScans / rate;
     SetCtrlVal (panelHandle, PANEL_ACQTIME, acqtime);
     free (waveforms);
     free (waveformsOrig);
     free (powers);
     free (powersOrig);
     waveforms = (double *) malloc (numScans * numChannels * sizeof (double));
     waveformsOrig = (double *) malloc (numScans * numChannels * sizeof (double));
     powers = (double *) malloc (numScans * 6 * sizeof (double));
     powersOrig = (double *) malloc (numScans * 6 * sizeof (double));
     SetAxisRange (panelHandle, PANEL_GRAPH, VAL_MANUAL, 0.0,acqtime*1000, VAL_AUTOSCALE, 0.0, 1.0);
     SetAxisRange (panelHandle, PANEL_GRAPH_2, VAL_MANUAL, 0.0,acqtime*1000, VAL_AUTOSCALE, 0.0, 1.0);
  break;
  }
  return 0;
}

/*****************************************************************************/
/*                                                                           */
/*  This callback function acquires and plots the data.                      */
/*                                                                           */
/*****************************************************************************/

int CVICALLBACK AcquireCallback (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
     case EVENT_COMMIT:
       SetInputMode (panelHandle, PANEL_LED, 1);
       SetCtrlVal (panelHandle, PANEL_LED, 1);
       AIAcquireWaveforms (device,channelString,numScans,rate,upper,lower,&actualRate,GROUP_BY_CHANNEL,
                           waveforms);
    DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, VAL_DELAYED_DRAW);
    DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, VAL_DELAYED_DRAW);

/*****************************************************************************
The following block calculates the rms value of the 6 channels, then
multiplies the different gains into the waveforms string, then
displays the rms of each of the 6 channels on the acquisition panel.
*****************************************************************************/
      for (i=0; i<numScans; i++)
      {
        waveforms[numScans*0+i] = lagain / currentturns * waveforms[numScans*0+i];
```

```
            waveformsOrig[numScans*0+i] = waveforms[numScans*0+i];
            waveforms[numScans*1+i] = Ibgain / currentturns * waveforms[numScans*1+i];
            waveformsOrig[numScans*1+i] = waveforms[numScans*1+i];
            waveforms[numScans*2+i] = Icgain / currentturns * waveforms[numScans*2+i];
            waveformsOrig[numScans*2+i] = waveforms[numScans*2+i];
            waveforms[numScans*3+i] = Vagain * waveforms[numScans*3+i];
            waveformsOrig[numScans*3+i] = waveforms[numScans*3+i];
            waveforms[numScans*4+i] = Vbgain * waveforms[numScans*4+i];
            waveformsOrig[numScans*4+i] = waveforms[numScans*4+i];
            waveforms[numScans*5+i] = Vcgain * waveforms[numScans*5+i];
            waveformsOrig[numScans*5+i] = waveforms[numScans*5+i];
            powers[numScans*0+i] = waveforms[numScans*0+i] * waveforms[numScans*3+i];  /*calculates pa(t)*/
            powersOrig[numScans*0+i] = waveforms[numScans*0+i] * waveforms[numScans*3+i];  /*calculates pa(t)*/
            powers[numScans*1+i] = waveforms[numScans*1+i] * waveforms[numScans*4+i];  /*calculates pb(t)*/
            powersOrig[numScans*1+i] = waveforms[numScans*1+i] * waveforms[numScans*4+i];  /*calculates pb(t)*/
            powers[numScans*2+i] = waveforms[numScans*2+i] * waveforms[numScans*5+i];  /*calculates pc(t)*/
            powersOrig[numScans*2+i] = waveforms[numScans*2+i] * waveforms[numScans*5+i];  /*calculates pc(t)*/
          }
          RMS (&waveforms[numScans*0], numScans, &Iarms);
          RMS (&waveforms[numScans*1], numScans, &Ibrms);
          RMS (&waveforms[numScans*2], numScans, &Icrms);
          RMS (&waveforms[numScans*3], numScans, &Varms);
          RMS (&waveforms[numScans*4], numScans, &Vbrms);
          RMS (&waveforms[numScans*5], numScans, &Vcrms);
          Integrate (&powers[numScans*0], numScans, 1/rate, 0.0, 0.0,&powers[numScans*3]);
          Integrate (&powers[numScans*1], numScans, 1/rate, 0.0, 0.0,&powers[numScans*4]);
          Integrate (&powers[numScans*2], numScans, 1/rate, 0.0, 0.0,&powers[numScans*5]);
          SetCtrlVal (panelHandle, PANEL_VARMS, Varms);
          SetCtrlVal (panelHandle, PANEL_VBRMS, Vbrms);
          SetCtrlVal (panelHandle, PANEL_VCRMS, Vcrms);
          Vaverms = (Varms + Vbrms + Vcrms)/3;
          SetCtrlVal (panelHandle, PANEL_VAVERMS, Vaverms);
          SetCtrlVal (panelHandle, PANEL_IARMS, Iarms);
          SetCtrlVal (panelHandle, PANEL_IBRMS, Ibrms);
          SetCtrlVal (panelHandle, PANEL_ICRMS, Icrms);
          Iaverms = (Iarms+Ibrms+Icrms)/3;
          SetCtrlVal (panelHandle, PANEL_IAVERMS, Iaverms);
          Parms = powers[numScans*4-1] / acqtime;
          Pbrms = powers[numScans*5-1] / acqtime;
          Pcrms = powers[numScans*6-1] / acqtime;
          Ptotrms = Parms + Pbrms + Pcrms;
          SetCtrlVal (panelHandle, PANEL_POWERA, Parms/1000);  /* power channel A*/
          SetCtrlVal (panelHandle, PANEL_POWERB, Pbrms/1000);  /* power channel B*/
          SetCtrlVal (panelHandle, PANEL_POWERC, Pcrms/1000);  /* power channel C*/
          SetCtrlVal (panelHandle, PANEL_POWERSUM, Ptotrms/1000);
          SetCtrlVal (panelHandle, PANEL_POWERFACTORA,
          powers[numScans*4-1]/(acqtime*Iarms*Varms));     /* display of pf phase A */
          SetCtrlVal (panelHandle, PANEL_POWERFACTORB,
          powers[numScans*5-1]/(acqtime*Ibrms*Vbrms));     /* display of pf phase B */
          SetCtrlVal (panelHandle, PANEL_POWERFACTORC,
          powers[numScans*6-1]/(acqtime*Icrms*Vcrms));     /* display of pf phase C */
          tmp1 = powers[numScans*4-1] + powers[numScans*5-1] + powers[numScans*6-1];     /* total power */
          tmp2 = Varms * Iarms + Vbrms * Ibrms + Vcrms * Icrms;                          /* total complex power */
          SetCtrlVal (panelHandle, PANEL_POWERFACTORAVE,                    /* display of total pf */
                  tmp1/(acqtime*tmp2));
          SetCtrlVal (panelHandle, PANEL_LED, 0);

/*****************************************************************************
The following block plots the six channels into the two windows from
the data acquisition window, if binary switch is set to on
*****************************************************************************/
          GetCtrlVal (panelHandle, PANEL_DISPLAYCURVES, &tmp);
          if (tmp == 1)
          {
            for (i=0; i<numChannels; i++)
            {
                  /*  The first 8 colors are defined by ColorArray[]  */
              if (i>2)
              {
                PlotY (panelHandle, PANEL_GRAPH, &waveforms[numScans*i], numScans, VAL_DOUBLE,
```

```
                    VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i-3]);
            }
            else
            {
              PlotY (panelHandle, PANEL_GRAPH_2, &waveforms[numScans*i], numScans, VAL_DOUBLE,
                    VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i]);
            }
          }
        }
        SetInputMode (currentFFTHandle, CURRENTFFT_PEAKFREQ, 0);
        SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
        SetInputMode (voltageFFTHandle, VOLTAGEFFT_PEAKFREQ, 0);
        SetInputMode (voltageFFTHandle, VOLTAGEFFT_COMMANDBUTTON, 0);
        SetInputMode (powerFFTHandle, POWERFFT_PEAKFREQ, 0);
        SetInputMode (powerFFTHandle, POWERFFT_COMMANDBUTTON, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_PEAKFREQ, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_COMMANDBUTTON, 0);
      break;
    }
    return 0;
}

int CVICALLBACK QuitCallback (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        free (waveforms);
        free (waveformsOrig);
        free (powers);
        free (powersOrig);
        if (flagFreeCurrent == 1)
        {
          free (currentA);
          free (currentB);
          free (currentC);
          flagFreeCurrent =0;
        }
        if (flagFreeVoltage ==1)
        {
          free (voltageA);
          free (voltageB);
          free (voltageC);
          flagFreeVoltage =0;
        }
        if (flagFreePower ==1)
        {
          free (powerA);
          free (powerB);
          free (powerC);
          free (powersum);
          flagFreePower = 0;
        }
        if (flagFreeImpedance ==1)
        {
          free (impedance);
          free (impedance2);
          free (angle);
          free (angle2);
          free (id);
          free (iq);
          free (vd);
          free (vq);
          free (lambdad);
          free (lambdaq);
          free (trqmech);
          flagFreeImpedance = 0;
        }
      QuitUserInterface (0);
      break;
```

```
  }
  return 0;
}

int CVICALLBACK ApplyFilter (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      GetCtrlVal (panelHandle, PANEL_FILTERTYPE, &filtertype);
      GetCtrlVal (panelHandle, PANEL_FILTER_ORDER, &filterorder);
      GetCtrlVal (panelHandle, PANEL_FILTER_CUTOFF, &filtercutoff);
      if (filtertype == 0)
      {
        for (i=0; i<6; i++)
        {
          Bw_LPF (&waveformsOrig[numScans * i], numScans, rate,  /* Butterworthing */
                  filtercutoff, filterorder, &waveforms[numScans * i]);
        }
      }
      if (filtertype == 1)
      {
       for (i=0; i<6; i++)
       {
          Ch_LPF (&waveformsOrig[numScans * i], numScans, rate, /* Chebysheving */
                  filtercutoff, 0.1, filterorder, &waveforms[numScans * i]);
        }
      }
      if (filtertype == 2)
      {
        for (i=0; i<6; i++)
        {
          InvCh_LPF (&waveformsOrig[numScans * i], numScans, rate, /* Inv. Chebysheving */
                    filtercutoff, 40.0, filterorder, &waveforms[numScans * i]);
        }
      }
      if (filtertype == 3)
      {
        for (i=0; i<6; i++)
        {
          Elp_LPF (&waveformsOrig[numScans * i], numScans, rate,  /* Ellyptic Filtering */
                  filtercutoff, 0.1, 40.0, filterorder, &waveforms[numScans * i]);
        }
      }
/* Deleting and redrawing the graphs after the filtering is done. */
      DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, VAL_DELAYED_DRAW);
      DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, VAL_DELAYED_DRAW);
      GetCtrlVal (panelHandle, PANEL_DISPLAYCURVES, &tmp);
      if (tmp == 1)
      {
        for (i=0; i<numChannels; i++)
        {
/*  The first 8 colors are defined by ColorArray[]  */
          if (i>2)
          {
            PlotY (panelHandle, PANEL_GRAPH, &waveforms[numScans*i],  numScans, VAL_DOUBLE,
                   VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i-3]);
          }
          else
          {
            PlotY (panelHandle, PANEL_GRAPH_2, &waveforms[numScans*i], numScans, VAL_DOUBLE,
                   VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i]);
          }
        }
      }
      SetInputMode (currentFFTHandle, CURRENTFFT_PEAKFREQ, 0);
      SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
      SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
      SetInputMode (voltageFFTHandle, VOLTAGEFFT_PEAKFREQ, 0);
      SetInputMode (voltageFFTHandle, VOLTAGEFFT_COMMANDBUTTON, 0);
```

```
        SetInputMode (powerFFTHandle, POWERFFT_PEAKFREQ, 0);
        SetInputMode (powerFFTHandle, POWERFFT_COMMANDBUTTON, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_PEAKFREQ, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_COMMANDBUTTON, 0);
        break;
      }
    return 0;
}

int CVICALLBACK NoFilter (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
      {
      case EVENT_COMMIT:
        DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, VAL_DELAYED_DRAW);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, VAL_DELAYED_DRAW);
/*****************************************************************************
The next block re-installs waveforms and powers from the original backups
waveformsOrig and powersOrig
*****************************************************************************/
        for (tmp = 0; tmp < numScans; tmp ++)
        {
          waveforms[numScans*0+tmp] = waveformsOrig[numScans*0+tmp];
          waveforms[numScans*1+tmp] = waveformsOrig[numScans*1+tmp];
          waveforms[numScans*2+tmp] = waveformsOrig[numScans*2+tmp];
          waveforms[numScans*3+tmp] = waveformsOrig[numScans*3+tmp];
          waveforms[numScans*4+tmp] = waveformsOrig[numScans*4+tmp];
          waveforms[numScans*5+tmp] = waveformsOrig[numScans*5+tmp];
          powers[numScans*0+tmp] = waveforms[numScans*0+tmp] * waveforms[numScans*3+tmp];
          powers[numScans*1+tmp] = waveforms[numScans*1+tmp] * waveforms[numScans*4+tmp];
          powers[numScans*2+tmp] = waveforms[numScans*2+tmp] * waveforms[numScans*5+tmp];
        }
/*****************************************************************************
The next block displays the old data on the graphs of the main panel,
only if the binary switch is on the ON position
*****************************************************************************/
        GetCtrlVal (panelHandle, PANEL_DISPLAYCURVES, &tmp);
        if (tmp == 1)
        {
          for (i=0; i<numChannels; i++)
          {
            if (i>2)
            {
              PlotY (panelHandle, PANEL_GRAPH, &waveformsOrig[numScans*i], numScans, VAL_DOUBLE,
                     VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i-3]);
            }
            else
            {
              PlotY (panelHandle, PANEL_GRAPH_2, &waveformsOrig[numScans*i], numScans, VAL_DOUBLE,
                     VAL_SCATTER, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[i]);
            }
          }
        }
        SetInputMode (currentFFTHandle, CURRENTFFT_PEAKFREQ, 0);
        SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
        SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
        SetInputMode (voltageFFTHandle, VOLTAGEFFT_PEAKFREQ, 0);
        SetInputMode (voltageFFTHandle, VOLTAGEFFT_COMMANDBUTTON, 0);
        SetInputMode (powerFFTHandle, POWERFFT_PEAKFREQ, 0);
        SetInputMode (powerFFTHandle, POWERFFT_COMMANDBUTTON, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_PEAKFREQ, 0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_COMMANDBUTTON, 0);
      break;
      }
    return 0;
}

void CVICALLBACK CurrentFFT (int menuBar, int menuItem, void *callbackData, int panel)
{
    DisplayPanel (currentFFTHandle);
```

```
  DeleteGraphPlot (currentFFTHandle, CURRENTFFT_FFT, -1, VAL_DELAYED_DRAW);
}

void CVICALLBACK VoltageFFT (int menuBar, int menuItem, void *callbackData,
    int panel)
{
  DisplayPanel (voltageFFTHandle);
  DeleteGraphPlot (voltageFFTHandle, VOLTAGEFFT_FFT, -1, VAL_DELAYED_DRAW);
}

void CVICALLBACK PowerFFT (int menuBar, int menuItem, void *callbackData, int panel)
{
  DisplayPanel (powerFFTHandle);
  DeleteGraphPlot (powerFFTHandle, POWERFFT_FFT, -1, VAL_DELAYED_DRAW);
}

void CVICALLBACK ImpedanceFFT (int menuBar, int menuItem, void *callbackData, int panel)
{
  DisplayPanel (impedanceFFTHandle);
  DeleteGraphPlot (impedanceFFTHandle, IMPEDFFT_FFT, -1, VAL_DELAYED_DRAW);
  SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING, rate/2);
  SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING_2, rate/numFFT);
  SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STARTFREQ, 28.0);
  SetCtrlVal (impedanceFFTHandle, IMPEDFFT_ENDFREQ, 30.0);
}

void CVICALLBACK NumberTurns (int menuBar, int menuItem, void *callbackData, int panel)
{
  DisplayPanel (currentturnHandle);
}

int CVICALLBACK CurrentMultiplier (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      GetCtrlVal (currentturnHandle, CURRENTTUR_NUMERIC, &currentturns);
    break;
    }
  return 0;
}

int CVICALLBACK CurrentturnDone (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      GetCtrlVal (currentturnHandle, CURRENTTUR_NUMERIC, &currentturns);
      HidePanel (currentturnHandle);
    break;
  }
  return 0;
}

int CVICALLBACK HideCurrentFFT (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  if (flagFreeCurrent ==1)
  {
    free (currentA);
    free (currentB);
    free (currentC);
    flagFreeCurrent = 0;
    SetInputMode (currentFFTHandle, CURRENTFFT_PEAKFREQ, 0);
    SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 0);
  }
  HidePanel (currentFFTHandle);
  return 0;
}

int CVICALLBACK DisplayCurrentFFT (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
```

```
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, &currentFFTslide);
    GetCtrlVal (currentFFTHandle, CURRENTFFT_ENDFREQ, &currentFFTendfreq);
    GetCtrlVal (currentFFTHandle, CURRENTFFT_STARTFREQ, &currentFFTstartfreq);
    DeleteGraphPlot (currentFFTHandle, CURRENTFFT_FFT, -1, VAL_DELAYED_DRAW);
    currentstartFFTsample = (int) (floor(numFFT*currentFFTstartfreq/rate));
    currentendFFTsample = (int) (ceil(numFFT*currentFFTendfreq/rate));
    if (currentstartFFTsample > currentendFFTsample)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nFrom Freq. < To Freq");
      currentstartFFTsample = 0;
      currentendFFTsample = numFFT/2;
    }
    if (currentendFFTsample > numFFT/2)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nMax Freq. < To Freq.");
      currentstartFFTsample = 0;
    }
    if (currentFFTslide ==4)
    {
    PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentA[currentstartFFTsample],
            currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
      PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentB[currentstartFFTsample],
             -currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
      PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentC[currentstartFFTsample],
             -currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
    }
    else if (currentFFTslide == 3)
    {
      PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentC[currentstartFFTsample],
             -currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
    }
    else if (currentFFTslide ==2)
    {
      PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentB[currentstartFFTsample],
             -currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
    }
    else
    {
      PlotY (currentFFTHandle, CURRENTFFT_FFT, &currentA[currentstartFFTsample],
             -currentstartFFTsample+currentendFFTsample, VAL_DOUBLE,
             VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
    }
    break;
  }
  return 0;
}

int CVICALLBACK FFTwarningOK (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      HidePanel (FFTWarningHandle);
    break;
  }
  return 0;
}

int CVICALLBACK DisplayVoltageFFT (int panel, int control, int event,
```

```
    void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_SLIDE,  &voltageFFTslide);
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_ENDFREQ, &voltageFFTendfreq);
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_STARTFREQ, &voltageFFTstartfreq);
      DeleteGraphPlot (voltageFFTHandle, VOLTAGEFFT_FFT, -1, VAL_DELAYED_DRAW);
      voltagestartFFTsample = (int) (floor(numFFT*voltageFFTstartfreq/rate));
      voltageendFFTsample = (int) (ceil(numFFT*voltageFFTendfreq/rate));
      if (voltagestartFFTsample > voltageendFFTsample)
      {
        DisplayPanel(FFTWarningHandle);
        SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. < From Freq.");
        voltagestartFFTsample = 0;
        voltageendFFTsample = numFFT/2;
      }
      if (voltageendFFTsample > numFFT/2)
      {
        DisplayPanel(FFTWarningHandle);
        SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. > Max. Freq.");
        voltagestartFFTsample = 0;
      }
      if (voltageFFTslide ==4)
      {
        PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageA[voltagestartFFTsample],
            -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
            VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
      PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageB[voltagestartFFTsample],
            -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
            VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
      PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageC[voltagestartFFTsample],
            -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
            VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
      }
      else if (voltageFFTslide == 3)
      {
        PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageC[voltagestartFFTsample],
            -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
            VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
      }
      else if (voltageFFTslide ==2)
      {
        PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageB[voltagestartFFTsample],
              -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
              VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
      }
      else
      {
        PlotY (voltageFFTHandle, VOLTAGEFFT_FFT, &voltageA[voltagestartFFTsample],
              -voltagestartFFTsample+voltageendFFTsample, VAL_DOUBLE,
              VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
      }
      break;
  }
  return 0;
}

int CVICALLBACK HideVoltageFFT (int panel, int control, int event,void *callbackData, int eventData1, int eventData2)
{
  if (flagFreeVoltage ==1)
  {
    free (voltageA);
    free (voltageB);
    free (voltageC);
    flagFreeVoltage = 0;
  }
  HidePanel (voltageFFTHandle);
  SetInputMode (voltageFFTHandle, VOLTAGEFFT_PEAKFREQ, 0);
```

```
  SetInputMode (voltageFFTHandle, VOLTAGEFFT_COMMANDBUTTON, 0);
  return 0;
}

int CVICALLBACK DisplayPowerFFT (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, &powerFFTslide);
    GetCtrlVal (powerFFTHandle, POWERFFT_ENDFREQ, &powerFFTendfreq);
    GetCtrlVal (powerFFTHandle, POWERFFT_STARTFREQ, &powerFFTstartfreq);
    DeleteGraphPlot (powerFFTHandle, POWERFFT_FFT, -1, VAL_DELAYED_DRAW);
    powerstartFFTsample = (int) (floor(numFFT*powerFFTstartfreq/rate));
    powerendFFTsample = (int) (ceil(numFFT*powerFFTendfreq/rate));
    if (powerstartFFTsample > powerendFFTsample)
    {
      DisplayPanel(FFTWarningHandle);
      SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. < From Freq.");
      powerstartFFTsample = 0;
      powerendFFTsample = numFFT/2;
    }
    if (powerendFFTsample > numFFT/2)
    {
      DisplayPanel(FFTWarningHandle);
      SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. > Max. Freq.");
      powerstartFFTsample = 0;
    }
    if (powerFFTslide ==4)
    {
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerA[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerB[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerC[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
    }
    else if (powerFFTslide == 3)
    {
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerC[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[2]);
    }
    else if (powerFFTslide ==2)
    {
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerB[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[1]);
    }
    else if (powerFFTslide ==1)
    {
      PlotY (powerFFTHandle, POWERFFT_FFT, &powerA[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[0]);
    }
    else
    {
      PlotY (powerFFTHandle, POWERFFT_FFT, &powersum[powerstartFFTsample],
          -powerstartFFTsample+powerendFFTsample, VAL_DOUBLE,
          VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, ColorArray[4]);
    }
    break;
  }
  return 0;
}
```

```
int CVICALLBACK HidePowerFFT (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  if (flagFreePower ==1)
  {
    free (powerA);
    free (powerB);
    free (powerC);
    free (powersum);
    flagFreePower = 0;
  }
  SetInputMode (powerFFTHandle, POWERFFT_PEAKFREQ, 0);
  SetInputMode (powerFFTHandle, POWERFFT_COMMANDBUTTON, 0);
  HidePanel (powerFFTHandle);
  return 0;
}

int CVICALLBACK DisplayImpedanceFFT (int panel, int control, int event, void *callbackData, int eventData1, int
                                     eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, &impedanceFFTslide);
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ENDFREQ, &impedanceFFTendfreq);
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_STARTFREQ, &impedanceFFTstartfreq);
    DeleteGraphPlot (impedanceFFTHandle, IMPEDFFT_FFT, -1, VAL_DELAYED_DRAW);
    impedancestartFFTsample = (int) (floor(numFFT*impedanceFFTstartfreq/rate));
    impedanceendFFTsample = (int) (ceil(numFFT*impedanceFFTendfreq/rate));
    if (impedancestartFFTsample > impedanceendFFTsample)
    {
      DisplayPanel(FFTWarningHandle);
      SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. < From Freq.");
      impedancestartFFTsample = 0;
      impedanceendFFTsample = numFFT/2;
    }
    if (impedanceendFFTsample > numFFT/2)
    {
      DisplayPanel(FFTWarningHandle);
      SetCtrlVal(FFTWarningHandle, FFTWARNING_TEXTBOX, "\nTo Freq. > Max. Freq.");
      impedancestartFFTsample = 0;
    }
    if (impedanceFFTslide ==3)
    {
      PlotY (impedanceFFTHandle, IMPEDFFT_FFT, &impedance[impedancestartFFTsample],
            -impedancestartFFTsample+impedanceendFFTsample,
            VAL_DOUBLE, VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, VAL_CYAN);
      PlotY (impedanceFFTHandle, IMPEDFFT_FFT, &angle[impedancestartFFTsample],
            -impedancestartFFTsample+impedanceendFFTsample,
            VAL_DOUBLE, VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, VAL_WHITE);
    }
    else if (impedanceFFTslide == 2)
    {
      PlotY (impedanceFFTHandle, IMPEDFFT_FFT, &angle[impedancestartFFTsample],
            -impedancestartFFTsample+impedanceendFFTsample,
            VAL_DOUBLE, VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, VAL_WHITE);
    }
    else
    {
      PlotY (impedanceFFTHandle, IMPEDFFT_FFT, &impedance[impedancestartFFTsample],
            -impedancestartFFTsample+impedanceendFFTsample,
            VAL_DOUBLE, VAL_CONNECTED_POINTS, VAL_SIMPLE_DOT, VAL_SOLID, 1, VAL_CYAN);
    }
    break;
  }
  return 0;
}

int CVICALLBACK HideImpedanceFFT (int panel, int control, int event, void *callbackData, int eventData1, int
                                  eventData2)
{
```

```
  if (flagFreeImpedance ==1)
  {
    free (impedance);
    free (impedance2);
    free (angle);
    free (angle2);
    free (id);
    free (iq);
    free (vd);
    free (vq);
    free (lambdad);
    free (lambdaq);
    free (trqmech);
    flagFreeImpedance = 0;
  }
  HidePanel (impedanceFFTHandle);
  SetInputMode (impedanceFFTHandle, IMPEDFFT_PEAKFREQ, 0);
  SetInputMode (impedanceFFTHandle, IMPEDFFT_COMMANDBUTTON, 0);
  return 0;
}

int CVICALLBACK PeakFreqCurrent (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, &currentFFTslide);
    GetCtrlVal (currentFFTHandle, CURRENTFFT_ENDFREQ, &currentFFTendfreq);
    GetCtrlVal (currentFFTHandle, CURRENTFFT_STARTFREQ, &currentFFTstartfreq);
    currentstartFFTsample = (int) (floor(numFFT*currentFFTstartfreq/rate));
    currentendFFTsample = (int) (ceil(numFFT*currentFFTendfreq/rate));
    if (currentstartFFTsample > currentendFFTsample)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nFrom Freq. < To Freq");
      currentstartFFTsample = 0;
      currentendFFTsample = numFFT/2;
    }
    if (currentendFFTsample > numFFT/2)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nMax Freq. < To Freq.");
      currentstartFFTsample = 0;
    }
    tmp1 = 0;
    tmp2 = 0;
    for (tmp = 0; tmp < currentendFFTsample-currentstartFFTsample; tmp++)
      {
        if (currentFFTslide == 3)
        {
          if(tmp2 < currentC[currentstartFFTsample+tmp])
          {
            tmp2 = currentC[currentstartFFTsample+tmp];
            tmp1 = currentstartFFTsample+tmp;
          }
        }
        else if (currentFFTslide ==2)
        {
          if(tmp2 < currentB[currentstartFFTsample+tmp])
          {
            tmp2 = currentB[currentstartFFTsample+tmp];
            tmp1 = currentstartFFTsample+tmp;
          }
        }
        else
        {
          if(tmp2 < currentA[currentstartFFTsample+tmp])
          {
            tmp2 = currentA[currentstartFFTsample+tmp];
            tmp1 = currentstartFFTsample+tmp;
```

```
            }
          }
        }
        SetCtrlVal (currentFFTHandle, CURRENTFFT_PEAKFREQNUM, tmp1*rate/numFFT);
        SetCtrlVal (currentFFTHandle, CURRENTFFT_ESTSPEED, 1800-60*((tmp1*rate/numFFT)-30));
        SetCtrlVal (currentFFTHandle, CURRENTFFT_PEAKFREQAMPL, tmp2);
      break;
      }
    return 0;
}

int CVICALLBACK PeakFreqVoltage (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_SLIDE, &voltageFFTslide);
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_ENDFREQ, &voltageFFTendfreq);
      GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_STARTFREQ, &voltageFFTstartfreq);
      voltagestartFFTsample = (int) (floor(numFFT*voltageFFTstartfreq/rate));
      voltageendFFTsample = (int) (ceil(numFFT*voltageFFTendfreq/rate));
      if (voltagestartFFTsample > voltageendFFTsample)
      {
        DisplayPanel (FFTWarningHandle);
        SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nFrom Freq. < To Freq");
        voltagestartFFTsample = 0;
        voltageendFFTsample = numFFT/2;
      }
      if (voltageendFFTsample > numFFT/2)
      {
        DisplayPanel (FFTWarningHandle);
        SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nMax Freq. < To Freq.");
        voltagestartFFTsample = 0;
      }
      tmp1 = 0;
      tmp2 = 0;
      for (tmp = 0; tmp < voltageendFFTsample-voltagestartFFTsample; tmp++)
        {
          if (voltageFFTslide == 3)
          {
            if(tmp2 < voltageC[voltagestartFFTsample+tmp])
            {
              tmp2 = voltageC[voltagestartFFTsample+tmp];
              tmp1 = voltagestartFFTsample+tmp;
            }
          }
          else if (voltageFFTslide ==2)
          {
            if(tmp2 < voltageB[voltagestartFFTsample+tmp])
            {
              tmp2 = voltageB[voltagestartFFTsample+tmp];
              tmp1 = voltagestartFFTsample+tmp;
            }
          }
          else
          {
            if(tmp2 < voltageA[voltagestartFFTsample+tmp])
            {
              tmp2 = voltageA[voltagestartFFTsample+tmp];
              tmp1 = voltagestartFFTsample+tmp;
            }
          }
        }
        SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_PEAKFREQNUM, tmp1*rate/numFFT);
        SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_PEAKFREQAMPL, tmp2);
      break;
    }
    return 0;
}
```

```
int CVICALLBACK PeakFreqImpedance (int panel, int control, int event, void *callbackData, int eventData1,
                                   int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, &impedanceFFTslide);
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ENDFREQ, &impedanceFFTendfreq);
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_STARTFREQ, &impedanceFFTstartfreq);
    impedancestartFFTsample = (int) (floor(numFFT*impedanceFFTstartfreq/rate));
    impedanceendFFTsample = (int) (ceil(numFFT*impedanceFFTendfreq/rate));
    if (impedancestartFFTsample > impedanceendFFTsample)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nFrom Freq. < To Freq");
      impedancestartFFTsample = 0;
      impedanceendFFTsample = numFFT/2;
    }
    if (impedanceendFFTsample > numFFT/2)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nMax Freq. < To Freq.");
      impedancestartFFTsample = 0;
    }
    tmp1 = 0;
    tmp2 = 0;
    for (tmp = 0; tmp < impedanceendFFTsample-impedancestartFFTsample; tmp++)
      {
        if (impedanceFFTslide == 1)
        {
          if(tmp2 < impedance[impedancestartFFTsample+tmp])
          {
            tmp2 = impedance[impedancestartFFTsample+tmp];
            tmp1 = impedancestartFFTsample+tmp;
          }
        }
        else
        {
          if(tmp2 < angle[impedancestartFFTsample+tmp])
          {
            tmp2 = angle[impedancestartFFTsample+tmp];
            tmp1 = impedancestartFFTsample+tmp;
          }
        }
      }
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_PEAKFREQNUM, tmp1*rate/numFFT);
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_ESTSPEED, 1800+60*((tmp1*rate/numFFT)-30));
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_PEAKFREQAMPL, tmp2);
    break;
  }
  return 0;
}

int CVICALLBACK PeakFreqPower (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, &powerFFTslide);
    GetCtrlVal (powerFFTHandle, POWERFFT_ENDFREQ, &powerFFTendfreq);
    GetCtrlVal (powerFFTHandle, POWERFFT_STARTFREQ, &powerFFTstartfreq);
    powerstartFFTsample = (int) (floor(numFFT*powerFFTstartfreq/rate));
    powerendFFTsample = (int) (ceil(numFFT*powerFFTendfreq/rate));
    if (powerstartFFTsample > powerendFFTsample)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nFrom Freq. < To Freq");
      powerstartFFTsample = 0;
      powerendFFTsample = numFFT/2;
    }
```

```
    if (powerendFFTsample > numFFT/2)
    {
      DisplayPanel (FFTWarningHandle);
      SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\nMax Freq. < To Freq.");
      powerstartFFTsample = 0;
    }
    tmp1 = 0;
    tmp2 = 0;
    for (tmp = 0; tmp < powerendFFTsample-powerstartFFTsample; tmp++)
    {
      if (powerFFTslide == 3)
      {
        if(tmp2 < powerC[powerstartFFTsample+tmp])
        {
          tmp2 = powerC[powerstartFFTsample+tmp];
          tmp1 = powerstartFFTsample+tmp;
        }
      }
      else if (powerFFTslide ==2)
      {
        if(tmp2 < powerB[powerstartFFTsample+tmp])
        {
          tmp2 = powerB[powerstartFFTsample+tmp];
          tmp1 = powerstartFFTsample+tmp;
        }
      }
      else
      {
        if(tmp2 < powerA[powerstartFFTsample+tmp])
        {
          tmp2 = powerA[powerstartFFTsample+tmp];
          tmp1 = powerstartFFTsample+tmp;
        }
      }
    }
    SetCtrlVal (powerFFTHandle, POWERFFT_PEAKFREQNUM, tmp1*rate/numFFT);
    SetCtrlVal (powerFFTHandle, POWERFFT_ESTSPEED, 1800+60*((tmp1*rate/numFFT)-30));
    SetCtrlVal (powerFFTHandle, POWERFFT_PEAKFREQAMPL, tmp2);
  break;
  }
  return 0;
}

int CVICALLBACK PreprocessCurrent (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    if (flagFreeCurrent == 1)
    {
      free (currentA);                                    /* preparing memory for current strings */
      free (currentB);
      free (currentC);
    }
    currentA = (double *) malloc (numScans * sizeof (double));
    currentB = (double *) malloc (numScans * sizeof (double));
    currentC = (double *) malloc (numScans * sizeof (double));
    flagFreeCurrent = 1;
    for (tmp=0; tmp<numScans; tmp ++)
    {
      currentA[tmp] = waveforms[numScans*0+tmp];
      currentB[tmp] = waveforms[numScans*1+tmp];
      currentC[tmp] = waveforms[numScans*2+tmp];
    }
    numFFT = 1;
    while (numFFT <= numScans)          /* figuring out the # of bits in numScans  */
    {
      numFFT = 2 * numFFT;
    }
    numFFT = numFFT / 2;
```

```
    Spectrum (currentA, numFFT);            /*  performing the spectral functions  */
    Spectrum (currentB, numFFT);
    Spectrum (currentC, numFFT);
    SetCtrlVal (currentFFTHandle, CURRENTFFT_STRING, rate/2);   /* setting display default str. in current fft wdw */
    SetCtrlVal (currentFFTHandle, CURRENTFFT_STRING_2, rate/numFFT);
    SetCtrlVal (currentFFTHandle, CURRENTFFT_STARTFREQ, 30.0);
    SetCtrlVal (currentFFTHandle, CURRENTFFT_ENDFREQ, 32.0);
    SetInputMode (currentFFTHandle, CURRENTFFT_PEAKFREQ, 1);
    SetInputMode (currentFFTHandle, CURRENTFFT_COMMANDBUTTON, 1);
  break;
  }
  return 0;
}

int CVICALLBACK PreprocessVoltage (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      if(flagFreeVoltage ==1)
      {
        free (voltageA);                                    /* preparing memory for voltage strings */
        free (voltageB);
        free (voltageC);
      }
      voltageA = (double *) malloc (numScans * sizeof (double));
      voltageB = (double *) malloc (numScans * sizeof (double));
      voltageC = (double *) malloc (numScans * sizeof (double));
      flagFreeVoltage = 1;
      for (tmp=0; tmp<numScans; tmp ++)
      {
        voltageA[tmp] = waveforms[numScans*3+tmp];
        voltageB[tmp] = waveforms[numScans*4+tmp];
        voltageC[tmp] = waveforms[numScans*5+tmp];
      }
      numFFT = 1;
      while (numFFT <= numScans)            /* figuring out the # of bits in numScans  */
      {
        numFFT = 2 * numFFT;
      }
      numFFT = numFFT / 2;
      Spectrum (voltageA, numFFT);          /*  performing the spectral functions  */
      Spectrum (voltageB, numFFT);
      Spectrum (voltageC, numFFT);
      SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_STRING, rate/2);       /* settg displ & default strs.in vltge fft wdw */
      SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_STRING_2, rate/numFFT);
      SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_STARTFREQ, 28.0);
      SetCtrlVal (voltageFFTHandle, VOLTAGEFFT_ENDFREQ, 32.0);
      SetInputMode (voltageFFTHandle, VOLTAGEFFT_PEAKFREQ, 1);
      SetInputMode (voltageFFTHandle, VOLTAGEFFT_COMMANDBUTTON, 1);
    break;
  }
  return 0;
}

int CVICALLBACK PreprocessPower (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      if(flagFreePower ==1)
      {
        free (powerA);                                      /* preparing memory for voltage strings */
        free (powerB);
        free (powerC);
        free (powersum);
      }
      powerA = (double *) malloc (numScans * sizeof (double));
      powerB = (double *) malloc (numScans * sizeof (double));
      powerC = (double *) malloc (numScans * sizeof (double));
```

```
        powersum = (double *) malloc (numScans * sizeof (double));
        flagFreePower = 1;
        for (tmp=0; tmp<numScans; tmp ++)
        {
          powerA[tmp] = waveforms[numScans*3+tmp] * waveforms[numScans*0+tmp];
          powerB[tmp] = waveforms[numScans*4+tmp] * waveforms[numScans*1+tmp];
          powerC[tmp] = waveforms[numScans*5+tmp] * waveforms[numScans*2+tmp];
          powersum[tmp] = powerA[tmp]+powerB[tmp]+powerC[tmp];
        }
        numFFT = 1;
        while (numFFT <= numScans)              /* figuring out the # of bits in numScans  */
        {
          numFFT = 2 * numFFT;
        }
        numFFT = numFFT / 2;
        Spectrum (powerA, numFFT);            /*  performing the spectral functions  */
        Spectrum (powerB, numFFT);
        Spectrum (powerC, numFFT);
        Spectrum (powersum, numFFT);
        SetCtrlVal (powerFFTHandle, POWERFFT_STRING, rate/2);          /* setting display & defit strgs vltg fft window */
        SetCtrlVal (powerFFTHandle, POWERFFT_STRING_2, rate/numFFT);
        SetCtrlVal (powerFFTHandle, POWERFFT_STARTFREQ, 28.0);
        SetCtrlVal (powerFFTHandle, POWERFFT_ENDFREQ, 30.0);
        SetInputMode (powerFFTHandle, POWERFFT_PEAKFREQ, 1);
        SetInputMode (powerFFTHandle, POWERFFT_COMMANDBUTTON, 1);
      break;
    }
    return 0;
}

int CVICALLBACK PreprocessImpedance (int panel, int control, int event, void *callbackData, int eventData1,
                                                    int eventData2)
{/**************************************************************************
   allocates memory for impedance, angle, vd, vq, lambdad, lamdaq,
   id, iq.  calculates vd, vq, id, iq according to 2 axis theory.
   integrates stator voltage minus stator current times stator resistance
   to get the fluxes of d and q axis. calculates the instantaneous angle
   between the current and the voltage vector.  calculates the
   instantaneous impedance.
   makes the estimate torque button active.
   sets the values into the displays of the fft impedance window.
**************************************************************************/
   switch (event)
   {
     case EVENT_COMMIT:
       if(flagFreeImpedance == 1)
       {
         free (impedance);
         free (impedance2);
         free (angle);
         free (angle2);
         free (vd);
         free (vq);
         free (lambdad);
         free (lambdaq);
         free (id);
         free (iq);
         free (trqmech);
       }
       impedance = (double *) malloc (numScans * sizeof (double));
       impedance2 = (double *) malloc (numScans * sizeof (double));
       angle = (double *) malloc (numScans * sizeof (double));
       angle2 = (double *) malloc (numScans * sizeof (double));
       vd =  (double *) malloc (numScans * sizeof (double));
       vq = (double *) malloc (numScans * sizeof (double));
       lambdad =  (double *) malloc (numScans * sizeof (double));
       lambdaq =  (double *) malloc (numScans * sizeof (double));
       id = (double *) malloc (numScans * sizeof (double));
       iq = (double *) malloc (numScans * sizeof (double));
       trqmech = (double *) malloc (numScans * sizeof (double));
```

```
        flagFreeImpedance =1;
        for (tmp=1; (tmp< numScans -1) ; tmp ++)
        {
          vq[tmp] = .666667 * waveforms[numScans*3+tmp] - .3333333 * waveforms[numScans*4+tmp] - .333333 *
                    waveforms[numScans*5+tmp];
          vd[tmp] = -.5773503 * waveforms[numScans*4+tmp] + .5773503 * waveforms[numScans*5+tmp];
          iq[tmp] = .666667 * waveforms[numScans*0+tmp] - .3333333 * waveforms[numScans*1+tmp] - .333333 *
                    waveforms[numScans*2+tmp];
          id[tmp] = -.5773503 * waveforms[numScans*1+tmp] + .5773503 * waveforms[numScans*2+tmp];
          impedance[tmp] = sqrt((vd[tmp]*vd[tmp]+vq[tmp]*vq[tmp])/(id[tmp]*id[tmp]+iq[tmp]*iq[tmp]));
          /*  next block has the zero phase shift integrator for the flux.
              zero phase means here that next sample equals the previous plus the average of the
              present and the future sample.  Equal approach to Kliman's, paper number one. */
          tmp1 = 180.0/pi*atan2(vq[tmp],vd[tmp]);    /*phi v*/
          tmp2 = 180.0/pi*atan2(iq[tmp],id[tmp]);   /*phi i*/
          if ((tmp1 - tmp2) < 0)                    /*angle (v, i)*/
            angle[tmp] = 360 + tmp1 - tmp2;
          else
            angle[tmp] = tmp1 - tmp2;
          impedance2[tmp] = impedance[tmp];
          angle2[tmp] = angle[tmp];
        }
        lambdaq[1] = 0;
        lambdad[1] = 0;
        for (tmp = 1; (tmp< numScans -1); tmp++)
        {
          lambdaq[tmp+1] = forgetfactor * lambdaq[tmp] + 0.5 * (vq[tmp] + vq[tmp+1] - rs * (iq[tmp] + iq[tmp+1]))/rate;
          lambdad[tmp+1] = forgetfactor * lambdad[tmp] + 0.5 * (vd[tmp] + vd[tmp+1] - rs * (id[tmp] + id[tmp+1]))/rate;
        }
        numFFT=1;
        while (numFFT<= numScans)
        {
          numFFT = 2 * numFFT;
        }
        numFFT = numFFT /2;
        Spectrum (impedance, numFFT);
        Spectrum (angle, numFFT);
        SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING, rate/2);
        SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING_2, rate/numFFT);
        SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STARTFREQ, 28.0);
        SetCtrlVal (impedanceFFTHandle, IMPEDFFT_ENDFREQ, 30.0);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_PEAKFREQ, 1);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_COMMANDBUTTON, 1);
        SetInputMode (impedanceFFTHandle, IMPEDFFT_ESTIMATERS, 1);
      break;
    }
    return 0;
}

void CVICALLBACK DisplaySaveDataWindow (int menuBar, int menuItem, void *callbackData, int panel)
{
    DisplayPanel (saveDataHandle);
}

int CVICALLBACK HideSaveWindow (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        HidePanel (saveDataHandle);
      break;
    }
    return 0;
}

int CVICALLBACK SaveData (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
```

```
        GetCtrlVal (saveDataHandle, SAVEDATA_MAKEhp, SaveMakeHp);
        GetCtrlVal (saveDataHandle, SAVEDATA_SN, SaveSerialNumber);
        GetCtrlVal (saveDataHandle, SAVEDATA_POLES, &SavePoles);
        GetCtrlVal (saveDataHandle, SAVEDATA_OPERATINGSPEED, &SaveOperatingSpeed);
        GetCtrlVal (saveDataHandle, SAVEDATA_OPERATINGMECHPOWER, &SaveMechPower);
        outFilePointer = fopen("RawData.txt", "w");    /* opens the file RawData.txt */
        fprintf(outFilePointer, "Make, hp %s\nSerial #%s\n# of Poles  %i\n", SaveMakeHp, SaveSerialNumber, SavePoles);
        fprintf(outFilePointer, "Operating Speed  %lf\n", SaveOperatingSpeed);
        fprintf(outFilePointer, "Mechanical Power %lf\n", SaveMechPower);
        fprintf(outFilePointer, "\n\nNumber of Scans  %d\n", numScans);
        fprintf(outFilePointer, "Sampling rate    %lf\n\n\n", rate);
        for (tmp = 0; tmp < 6 * numScans; tmp++)
        {
          fprintf(outFilePointer, "%lf\n", waveformsOrig[tmp]);
        }
        HidePanel(saveDataHandle);
        fclose(outFilePointer);
      break;
    }
    return 0;
}

int CVICALLBACK CurrentDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        DisplayPanel (currentDFLLHandle);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_WDWPCT, 800/dfllnumfreq);
      break;
    }
    return 0;
}

int CVICALLBACK HideCurrentDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        HidePanel (currentDFLLHandle);
        SetInputMode (currentDFLLHandle, CURRDFLL_SUBTRACT, 0);
        DeleteGraphPlot (currentDFLLHandle, CURRDFLL_GRAPH, -1, VAL_IMMEDIATE_DRAW);
        dfllcurrentflag = 0;
      break;
    }
    return 0;
}

int CVICALLBACK ApplyCurrentDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        dfllaposterioriCnt = 0;
        /*************************************************************************
          Getting the required variables for performing one iteration of the
          Digital Frequency Locked Loop
        *************************************************************************/
        if (dfllcurrentflag == 0)
        {
          GetCtrlVal (currentDFLLHandle, CURRDFLL_MINFREQ, &dfllminfreq);
          GetCtrlVal (currentDFLLHandle, CURRDFLL_MAXFREQ, &dfllmaxfreq);
        }
        GetCtrlVal (currentDFLLHandle, CURRDFLL_ACCURACY, &dfllreqAccuracy);
        do
        {
          dflldeltafreq = dfllmaxfreq - dfllminfreq;
          GetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, &dfllcurrchan);
          GetCtrlVal (currentDFLLHandle, CURRDFLL_NUMFREQ, &dfllnumfreq);
          GetCtrlVal (currentDFLLHandle, CURRDFLL_WDWPCT, &dfllwdwpct);
```

```
if (dfllcurrchan == 4)
  dfllcurrchan = 1;
/*************************************************************************
  Finding the maximum time length and the maximum number of samples
  to be used in order to avoid loosing ghost frequency information.
  The min requirement will be divided by three, so that there is a
  smaller distortion between the amplitude of a frequency close to a ghost
  frequency and an other frequency, which has the gain of one because
  it lays directly on the gatterpoint.
*************************************************************************/
if (acqtime > 0.3 * dfllnumfreq/dflldeltafreq)
{
  dfllNumSamp = floor(.5 * dfllnumfreq * rate / dflldeltafreq);
  if (dfllNumSamp>numScans)
    dfllNumSamp = numScans;
}
else
{
  dfllNumSamp = numScans;
}
/*************************************************************************
    Accelerating the procedure, by using only a part of the samples, if the
    last iteration was 'very stable' ((dfllminamp/dfllmaxamp)^2 << 1)
*************************************************************************/
if (dfllcurrentflag ==1)
{
  dfllNumSamp = ceil((.2 + .8 * (dfllminamp*dfllminamp) / (dfllmaxamp*dfllmaxamp)) * dfllNumSamp);
}
dfllmaxamp = 0;
dfllminamp = 500000;
dfllpeakfreq = 0;
dfllsmallfreq = 0;
/*************************************************************************
  Starting the outer loop.  Outer loop = frequency sweep variable is tmp
*************************************************************************/
for (tmp = 0; tmp < dfllnumfreq; tmp++)
{
  tmp3 = twopi/rate*(tmp*dflldeltafreq/dfllnumfreq+dfllminfreq);
  tmp1 = 0;
  tmp2 = 0;
  /*************************************************************************
      making dfllNumSamp such that it portrays an integer number of periods
      of the frequency generated (innter product = 0).
  *************************************************************************/
  dfllNumPeriods = (int) ceil(dfllNumSamp*(dfllminfreq+tmp*dflldeltafreq)/rate);
  if (dfllNumPeriods/(dfllminfreq+tmp*dflldeltafreq) > acqtime)
  {
    dfllNumPeriods = dfllNumPeriods -1;
  }
  dfllNumSamp = (int) floor(dfllNumPeriods*rate/(dfllminfreq+tmp*dflldeltafreq));
  /*************************************************************************
    Starting the inner loop.  Inner loop = time sweep; variable is i
  *************************************************************************/
  for (i = 0; i < dfllNumSamp; i++)
  {
    tmp1 = tmp1 + sin(tmp3*i) * waveforms[numScans*(dfllcurrchan-1)+i];
    tmp2 = tmp2 + cos(tmp3*i) * waveforms[numScans*(dfllcurrchan-1)+i];
    dfllamp[tmp] += sin(tmp3*i) * sin(tmp3*i);
  }
  tmp4[tmp] = sqrt(tmp1*tmp1+tmp2*tmp2)/dfllNumSamp;
  dfllamp[tmp] = tmp4[tmp] / dfllamp[tmp]; /* which makes dfllamp the relative amplitude of that frequency */
  dfllang[tmp] = atan2(tmp2,tmp1);
  if (dfllmaxamp < tmp4[tmp])
  {
    dfllmaxSamp = tmp;
    dfllmaxamp = tmp4[tmp];
    dfllpeakfreq = dflldeltafreq*tmp/dfllnumfreq + dfllminfreq;
  }
  if (dfllminamp > tmp4[tmp])
  {
```

```
            dfllminamp = tmp4[tmp];
          }
        }
        DeleteGraphPlot (currentDFLLHandle, CURRDFLL_GRAPH, -1, VAL_DELAYED_DRAW);
        PlotY (currentDFLLHandle, CURRDFLL_GRAPH, &tmp4, dfllnumfreq, VAL_DOUBLE, VAL_THIN_LINE,
               VAL_SOLID_CIRCLE, VAL_SOLID, 1, VAL_RED);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_DELTAFREQO, (dfllmaxfreq-dfllminfreq)/dfllnumfreq);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_PEAKFREQ, dfllpeakfreq);
        /*****************************************************************************
            A-posteriori check.  If the max value happens to be at one of the borders of the
            frequency window that is looked at, then no zooming will happen, just the window
            will be shifted towards the side with the max value.
        *****************************************************************************/
        if (dfllmaxSamp == 0 )
        {
          dfllmaxfreq = dfllmaxfreq - .5 * dflldeltafreq;
          dfllminfreq = dfllminfreq - .6 * dflldeltafreq;
          dfllaposterioriCnt++;
        }
        else if (dfllmaxSamp == (dfllnumfreq-1))
        {
          dfllmaxfreq = dfllmaxfreq + .6 * dflldeltafreq;
          dfllminfreq = dfllminfreq + .5 * dflldeltafreq;
          dfllaposterioriCnt++;
        }
        /*****************************************************************************
            End of a-posteriori check, and zooming in of frequencies if a-posteriori negative.
        *****************************************************************************/
        else if (dfllmaxSamp < (dfllnumfreq-1) && dfllmaxSamp >0)
        {
          dfllmaxfreq = dfllpeakfreq + dflldeltafreq*dfllwdwpct/200;
          dfllminfreq = dfllpeakfreq - dflldeltafreq*dfllwdwpct/200;
          dfllaposterioriCnt = 0;
        }
        SetCtrlVal (currentDFLLHandle, CURRDFLL_MAXFREQN, dfllmaxfreq);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_MINFREQN, dfllminfreq);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_ESTSPEED, 60*(60-dfllpeakfreq));
        dfllcurrentflag = 1;
        if (dfllaposterioriCnt >= 3)
        {
          DisplayPanel (FFTWarningHandle);
          SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\n3 a-posteriori checks.  Loss of reliable
                zooming.");
        }
      } while ((dfllaposterioriCnt <= 3)&&(60*dflldeltafreq/dfllnumfreq)>dfllreqAccuracy);
      SetInputMode (currentDFLLHandle, CURRDFLL_SUBTRACT, 1);
    break;
  }
  return 0;
}

int CVICALLBACK DfllCurrentPctWdw (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
    {
      case EVENT_COMMIT:
        GetCtrlVal (currentDFLLHandle, CURRDFLL_NUMFREQ, &dfllnumfreq);
        SetCtrlVal (currentDFLLHandle, CURRDFLL_WDWPCT, 800/dfllnumfreq);
      break;
    }
  return 0;
}

int CVICALLBACK DfllCurrentSubtractFreq (int panel, int control, int event, void *callbackData, int eventData1,
                                         int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      SetInputMode (currentDFLLHandle, CURRDFLL_SUBTRACT, 0);
```

```
      for (i = 0; i < numScans; i++)
      {
        waveforms[numScans*(dfllcurrchan-1)+i] += -dfllamp[dfllmaxSamp] *
                                                  sin(twopi*dfllpeakfreq*i/rate+dfllang[dfllmaxSamp]);
      }
    break;
  }
  return 0;
}

int CVICALLBACK ApplyVoltageDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      dfllaposterioriCnt = 0;
      /*********************************************************************
        Getting the required variables for performing one iteration of the
        Digital Frequency Locked Loop
      *********************************************************************/
      if (dfllvoltageflag == 0)
      {
        GetCtrlVal (voltageDFLLHandle, VOLTDFLL_MINFREQ, &dfllminfreq);
        GetCtrlVal (voltageDFLLHandle, VOLTDFLL_MAXFREQ, &dfllmaxfreq);
      }
      GetCtrlVal (voltageDFLLHandle, VOLTDFLL_ACCURACY, &dfllreqAccuracy);
      do
      {
        dflldeltafreq = dfllmaxfreq - dfllminfreq;
        GetCtrlVal (voltageFFTHandle, VOLTAGEFFT_SLIDE, &dfllvoltchan);
        GetCtrlVal (voltageDFLLHandle, VOLTDFLL_NUMFREQ, &dfllnumfreq);
        SetCtrlVal (voltageDFLLHandle, VOLTDFLL_WDWPCT, 800/dfllnumfreq);
        GetCtrlVal (voltageDFLLHandle, VOLTDFLL_WDWPCT, &dfllwdwpct);
        if (dfllvoltchan == 4)
          dfllvoltchan = 3;
        /*************************************************************************
          Finding the maximum time length and the maximum number of samples
          to be used in order to avoid loosing ghost frequency information.
          The min requirement will be divided by three, so that there is a
          smaller distortion between the amplitude of a frequency close to a ghost
          frequency and an other frequency, which has the gain of one because
          it lays directly on the gatterpoint.
        *************************************************************************/
        if (acqtime > 0.5 * dfllnumfreq/dflldeltafreq)
        {
          dfllNumSamp = floor(.5 * dfllnumfreq * rate / dflldeltafreq);
          if (dfllNumSamp>numScans)
            dfllNumSamp = numScans;
        }
        else
        {
          dfllNumSamp = numScans;
        }
        /*************************************************************************
            Accelerating the procedure, by using only a part of the samples, if the
            last iteration was 'very stable' ((dfllminamp/dfllmaxamp)^2 << 1)
        *************************************************************************/
        if (dfllvoltageflag ==1)
        {
          dfllNumSamp = ceil((.2 + .8 * (dfllminamp*dfllminamp) / (dfllmaxamp*dfllmaxamp)) * dfllNumSamp);
        }
        dfllmaxamp = 0;
        dfllminamp = 500000;
        dfllpeakfreq = 0;
        dfllsmallfreq = 0;
        /*************************************************************************
          Starting the outer loop.  Outer loop = frequency sweep variable is tmp
        *************************************************************************/
        for (tmp = 0; tmp < dfllnumfreq; tmp++)
        {
```

```
    tmp3 = twopi/rate*(tmp*dflldeltafreq/dfllnumfreq+dfllminfreq);
    tmp1 = 0;
    tmp2 = 0;
    /*************************************************************************
        making dfllNumSamp such that it portrays an integer number of periods
        of the frequency generated (innter product = 0).
    *************************************************************************/
    dfllNumPeriods = (int) ceil(dfllNumSamp*(dfllminfreq+tmp*dfllde ltafreq)/rate);
    if (dfllNumPeriods/(dfllminfreq+tmp*dfllde ltafreq) > acqtime)
    {
      dfllNumPeriods = dfllNumPeriods -1;
    }
    dfllNumSamp = (int) floor(dfllNumPeriods*rate/(dfllminfreq+tmp*dfllde ltafreq));

    /*************************************************************************
      Starting the inner loop.  Inner loop = time sweep; variable is i
    *************************************************************************/
    for (i = 0; i < dfllNumSamp; i++)
    {
      tmp1 = tmp1 + sin(tmp3*i) * waveforms[numScans*(dfllvoltchan-1+3)+i];
      tmp2 = tmp2 + cos(tmp3*i) * waveforms[numScans*(dfllvoltchan-1+3)+i];
      dfllamp[tmp] += sin(tmp3*i) * sin(tmp3*i);
    }
    tmp4[tmp] = sqrt(tmp1*tmp1+tmp2*tmp2);       /*dfllNumSamp;*/
    dfllamp[tmp] = tmp4[tmp] / dfllamp[tmp]; /* which makes dfllamp the relative amplitude of that frequency */
    dfllang[tmp] = atan2(tmp2,tmp1);
    if (dfllmaxamp < tmp4[tmp])
    {
      dfllmaxSamp = tmp;
      dfllmaxamp = tmp4[tmp];
      dfllpeakfreq = dfllde ltafreq*tmp/dfllnumfreq + dfllminfreq;
    }
    if (dfllminamp > tmp4[tmp])
    {
      dfllminamp = tmp4[tmp];
    }
}
DeleteGraphPlot (voltageDFLLHandle, VOLTDFLL_GRAPH, -1, VAL_DELAYED_DRAW);
PlotY (voltageDFLLHandle, VOLTDFLL_GRAPH, &tmp4, dfllnumfreq,
       VAL_DOUBLE, VAL_THIN_LINE, VAL_SOLID_CIRCLE, VAL_SOLID, 1, VAL_RED);
SetCtrlVal (voltageDFLLHandle, VOLTDFLL_DELTAFREQO, (dfllmaxfreq-dfllminfreq)/dfllnumfreq);
SetCtrlVal (voltageDFLLHandle, VOLTDFLL_PEAKFREQ, dfllpeakfreq);
/*************************************************************************************
    A-posteriori check.  If the max value happens to be at one of the borders of the
    frequency window that is looked at, then no zooming will happen, just the window
    will be shifted towards the side with the max value.
*************************************************************************************/
if (dfllmaxSamp == 0 )
{
  dfllmaxfreq = dfllmaxfreq - .5 * dfllde ltafreq;
  dfllminfreq = dfllminfreq - .6 * dfllde ltafreq;
  dfllaposterioriCnt++;
}
else if (dfllmaxSamp == (dfllnumfreq-1))
{
  dfllmaxfreq = dfllmaxfreq + .6 * dfllde ltafreq;
  dfllminfreq = dfllminfreq + .5 * dfllde ltafreq;
  dfllaposterioriCnt++;
}
/*************************************************************************************
    End of a-posteriori check, and zooming in of frequencies if a-posteriori negative.
*************************************************************************************/
else if (dfllmaxSamp < (dfllnumfreq-1) && dfllmaxSamp >0)
{
  dfllmaxfreq = dfllpeakfreq + dfllde ltafreq*dfllwdwpct/200;
  dfllminfreq = dfllpeakfreq - dfllde ltafreq*dfllwdwpct/200;
  dfllaposterioriCnt = 0;
}
SetCtrlVal (voltageDFLLHandle, VOLTDFLL_MAXFREQN, dfllmaxfreq);
SetCtrlVal (voltageDFLLHandle, VOLTDFLL_MINFREQN, dfllminfreq);
```

```
            SetCtrlVal (voltageDFLLHandle, VOLTDFLL_ESTSPEED, 60*(60-dfllpeakfreq));
            dfllvoltageflag = 1;
            if (dfllaposterioriCnt >= 3)
            {
              DisplayPanel (FFTWarningHandle);
              SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\n3 a-posteriori checks.  Loss of reliable
                          zooming.");
            }
          } while ((dfllaposterioriCnt <= 3)&&(60*dflldeltafreq/dfllnumfreq)>dfllreqAccuracy);
          SetInputMode (voltageDFLLHandle, VOLTDFLL_SUBTRACT, 1);
        break;
      }
      return 0;
}


int CVICALLBACK DfllVoltagePctWdw (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
      switch (event)
        {
          case EVENT_COMMIT:
            GetCtrlVal (voltageDFLLHandle, VOLTDFLL_NUMFREQ, &dfllnumfreq);
            SetCtrlVal (voltageDFLLHandle, VOLTDFLL_WDWPCT, 800/dfllnumfreq);
          break;
        }
      return 0;
}

int CVICALLBACK DfllVoltageSubtractFreq (int panel, int control, int event, void *callbackData, int eventData1,
                                         int eventData2)
{
      switch (event)
      {
        case EVENT_COMMIT:
          SetInputMode (voltageDFLLHandle, VOLTDFLL_SUBTRACT, 0);
          for (i = 0; i < numScans; i++)
          {
            waveforms[numScans*(dfllvoltchan-1+3)+i] +=
                                          -dfllamp[dfllmaxSamp] * sin(twopi*dfllpeakfreq*i/rate+dfllang[dfllmaxSamp]);
          }
        break;
      }
      return 0;
}

int CVICALLBACK VoltageDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
      switch (event)
      {
        case EVENT_COMMIT:
          DisplayPanel (voltageDFLLHandle);
        break;
      }
      return 0;
}

int CVICALLBACK HideVoltageDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
      switch (event)
      {
        case EVENT_COMMIT:
          HidePanel (voltageDFLLHandle);
          SetInputMode (voltageDFLLHandle, VOLTDFLL_SUBTRACT, 0);
          DeleteGraphPlot (voltageDFLLHandle, VOLTDFLL_GRAPH, -1, VAL_IMMEDIATE_DRAW);
          dfllvoltageflag = 0;
        break;
      }
      return 0;
}
```

```
int CVICALLBACK EstimateRs (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      rs = .08 * Vaverms/Iaverms;
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_RS, rs);
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING_3, "\nestimated");
      SetInputMode (impedanceFFTHandle, IMPEDFFT_ESTIMATETRQ, 1);
    break;
  }
  return 0;
}

int CVICALLBACK RsEntered (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_STRING_3, "\nentered");
    break;
  }
  return 0;
}

int CVICALLBACK EstimateTrq (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
    GetCtrlVal (impedanceFFTHandle, IMPEDFFT_RS, &rs);
    /*****************************************************************************
        Lookup table of fricpwr (power lost due to friction and windage)
        as a function of rated power.  The function is implemented to
        have a continuous derivative.
        Calculates the estimated F&W torque losses, using speedEstimate
    *****************************************************************************/
      GetCtrlVal (panelHandle, PANEL_RATEDPWR, &ratedpwr);
      if (ratedpwr < 1)
      {
        fricpwr =  1 * ratedpwr;                    /* because air clutch at Baker, 2* factor */
      }
      else
      {
        if (ratedpwr < 5)
        {
          fricpwr = (1 + 8 * (ratedpwr - 1.0));    /* because air clutch at Baker, 2* factor */
        }
        else
        {
          if (ratedpwr < 30)
          {
            fricpwr = (42 + 6 * (ratedpwr - 5.0));
          }
          else
          {
            if (ratedpwr < 100)
            {
              fricpwr = (202 + 4 * (ratedpwr - 30));
            }
            else
            {
              if (ratedpwr < 300)
              {
                fricpwr = (482 + 3.5 * (ratedpwr - 100));
              }
              else
              {
                fricpwr = (1182 + 3 * (ratedpwr - 300));
              }
```

```
            }
          }
        }
      }
      fricpwr = 40*fricpwr;
      for (tmp=1; tmp<numScans-2; tmp ++)
      {
        lambdaq[tmp+1] = forgetfactor * lambdaq[tmp] + 0.5 * (vq[tmp] + vq[tmp+1] - rs * (iq[tmp] + iq[tmp+1]))/rate;
        lambdad[tmp+1] = forgetfactor * lambdad[tmp] + 0.5 * (vd[tmp] + vd[tmp+1] - rs * (id[tmp] + id[tmp+1]))/rate;
        GetCtrlVal (impedanceFFTHandle, IMPEDFFT_POLEPAIRS, &Pp);
        trqmech[tmp] = (3*Pp/4)*(lambdad[tmp]*iq[tmp]-lambdaq[tmp]*id[tmp]);
      }
      Mean (trqmech, numScans-1, &trqmechave);
      trqFW = fricpwr * 60 / (twopi * speedEstimate);
      trqest = trqmechave - trqFW;
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_FRICTIONTRQ, trqFW);
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_TRQEST, trqest);
      SetCtrlVal (impedanceFFTHandle, IMPEDFFT_EFFEST, twopi*speedEstimate*trqest*100/(Ptotrms*60));
    break;
  }
  return 0;
}

int CVICALLBACK ImpedanceDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      DisplayPanel (impedDFLLHandle);
      SetCtrlVal (impedDFLLHandle, IMPEDDFLL_WDWPCT, 800/dfllnumfreq);
    break;
  }
  return 0;
}

int CVICALLBACK ApplyImpedDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      dfllaposterioriCnt = 0;
      /*************************************************************************
        Getting the required variables for performing one iteration of the
        Digital Frequency Locked Loop
      *************************************************************************/
      if (dfllimpedflag == 0)
      {
        GetCtrlVal (impedDFLLHandle, IMPEDDFLL_MINFREQ, &dfllminfreq);
        GetCtrlVal (impedDFLLHandle, IMPEDDFLL_MAXFREQ, &dfllmaxfreq);
      }
      GetCtrlVal (impedDFLLHandle, IMPEDDFLL_ACCURACY, &dfllreqAccuracy);
      do
      {
        dflldeltafreq = dfllmaxfreq - dfllminfreq;
        GetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, &dfllcurrchan);
        GetCtrlVal (impedDFLLHandle, IMPEDDFLL_NUMFREQ, &dfllnumfreq);
        GetCtrlVal (impedDFLLHandle, IMPEDDFLL_WDWPCT, &dfllwdwpct);
        if (dfllcurrchan == 4)
          dfllcurrchan = 1;
        /*************************************************************************
          Finding the maximum time length and the maximum number of samples
          to be used in order to avoid loosing ghost frequency information.
          The min requirement will be divided by three, so that there is a
          smaller distortion between the amplitude of a frequency close to a ghost
          frequency and an other frequency, which has the gain of one because
          it lays directly on the gatterpoint.
        *************************************************************************/
        if (acqtime > 0.3 * dfllnumfreq/dflldeltafreq)
        {
          dfllNumSamp = floor(.5 * dfllnumfreq * rate / dflldeltafreq);
```

```
  if (dfllNumSamp>numScans)
    dfllNumSamp = numScans;
}
else
{
  dfllNumSamp = numScans;
}
/*************************************************************************
    Accelerating the procedure, by using only a part of the samples, if the
    last iteration was 'very stable' ((dfllminamp/dfllmaxamp)^2 << 1)
*************************************************************************/
if (dfllimpedflag ==1)
{
  dfllNumSamp = ceil((.2 + .8 * (dfllminamp*dfllminamp) / (dfllmaxamp*dfllmaxamp)) * dfllNumSamp);
}
dfllmaxamp = 0;
dfllminamp = 500000;
dfllpeakfreq = 0;
dfllsmallfreq = 0;
/*************************************************************************
  Starting the outer loop.  Outer loop = frequency sweep variable is tmp
*************************************************************************/
for (tmp = 0; tmp < dfllnumfreq; tmp++)
{
  tmp3 = twopi/rate*(tmp*dflldeltafreq/dfllnumfreq+dfllminfreq);
  tmp1 = 0;
  tmp2 = 0;
  /*************************************************************************
      making dfllNumSamp such that it portrays an integer number of periods
      of the frequency generated (innter product = 0).
  *************************************************************************/
  dfllNumPeriods = (int) ceil(dfllNumSamp*(dfllminfreq+tmp*dflldeltafreq)/rate);
  if (dfllNumPeriods/(dfllminfreq+tmp*dflldeltafreq) > acqtime)
  {
    dfllNumPeriods = dfllNumPeriods -1;
  }
  dfllNumSamp = (int) floor(dfllNumPeriods*rate/(dfllminfreq+tmp*dflldeltafreq));
  /*************************************************************************
    Starting the inner loop.  Inner loop = time sweep; variable is i
  *************************************************************************/
  if (dfllcurrchan == 1)
  {
    for (i = 0; i < dfllNumSamp; i++)
    {
      tmp1 = tmp1 + sin(tmp3*i) * impedance2[i];
      tmp2 = tmp2 + cos(tmp3*i) * impedance2[i];
      dfllamp[tmp] += sin(tmp3*i) * sin(tmp3*i);
    }
  }
  else
  {
    for (i = 0; i < dfllNumSamp; i++)
    {
      tmp1 = tmp1 + sin(tmp3*i) * angle2[i];
      tmp2 = tmp2 + cos(tmp3*i) * angle2[i];
      dfllamp[tmp] += sin(tmp3*i) * sin(tmp3*i);
    }
  }
  tmp4[tmp] = sqrt(tmp1*tmp1+tmp2*tmp2)/dfllNumSamp;
  dfllamp[tmp] = tmp4[tmp] / dfllamp[tmp]; /* which makes dfllamp the relative amplitude of that frequency */
  dfllang[tmp] = atan2(tmp2,tmp1);
  if (dfllmaxamp < tmp4[tmp])
  {
    dfllmaxSamp = tmp;
    dfllmaxamp = tmp4[tmp];
    dfllpeakfreq = dflldeltafreq*tmp/dfllnumfreq + dfllminfreq;
  }
  if (dfllminamp > tmp4[tmp])
  {
    dfllminamp = tmp4[tmp];
```

```
            }
          }
          DeleteGraphPlot (impedDFLLHandle, IMPEDDFLL_GRAPH, -1, VAL_DELAYED_DRAW);
          PlotY (impedDFLLHandle, IMPEDDFLL_GRAPH, &tmp4, dfllnumfreq, VAL_DOUBLE, VAL_THIN_LINE,
                 VAL_SOLID_CIRCLE, VAL_SOLID, 1, VAL_RED);
          SetCtrlVal (impedDFLLHandle, IMPEDDFLL_DELTAFREQO, (dfllmaxfreq-dfllminfreq)/dfllnumfreq);
          SetCtrlVal (impedDFLLHandle, IMPEDDFLL_PEAKFREQ, dfllpeakfreq);
          /*****************************************************************************
              A-posteriori check.  If the max value happens to be at one of the borders of the
              frequency window that is looked at, then no zooming will happen, just the window
              will be shifted towards the side with the max value.
          *****************************************************************************/
          if (dfllmaxSamp == 0 )
          {
            dfllmaxfreq = dfllmaxfreq - .5 * dflldeltafreq;
            dfllminfreq = dfllminfreq - .6 * dflldeltafreq;
            dfllaposterioriCnt++;
          }
          else if (dfllmaxSamp == (dfllnumfreq-1))
          {
            dfllmaxfreq = dfllmaxfreq + .6 * dflldeltafreq;
            dfllminfreq = dfllminfreq + .5 * dflldeltafreq;
            dfllaposterioriCnt++;
          }
          /*****************************************************************************
              End of a-posteriori check, and zooming in of frequencies if a-posteriori negative.
          *****************************************************************************/
          else if (dfllmaxSamp < (dfllnumfreq-1) && dfllmaxSamp >0)
          {
            dfllmaxfreq = dfllpeakfreq + dflldeltafreq*dfllwdwpct/200;
            dfllminfreq = dfllpeakfreq - dflldeltafreq*dfllwdwpct/200;
            dfllaposterioriCnt = 0;
          }
          SetCtrlVal (impedDFLLHandle, IMPEDDFLL_MAXFREQN, dfllmaxfreq);
          SetCtrlVal (impedDFLLHandle, IMPEDDFLL_MINFREQN, dfllminfreq);
          SetCtrlVal (impedDFLLHandle, IMPEDDFLL_ESTSPEED, 1800-60*(30-dfllpeakfreq));
          dfllimpedflag = 1;
          if (dfllaposterioriCnt >= 3)
          {
            DisplayPanel (FFTWarningHandle);
            SetCtrlVal (FFTWarningHandle, FFTWARNING_TEXTBOX, "\n3 a-posteriori checks.  Loss of reliable
                        zooming.");
          }
        } while ((dfllaposterioriCnt <= 3)&&(60*dflldeltafreq/dfllnumfreq)>dfllreqAccuracy);
        SetInputMode (impedDFLLHandle, IMPEDDFLL_SUBTRACT, 1);
      break;
    }
    return 0;
}

int CVICALLBACK HideImpedDFLL (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        HidePanel (impedDFLLHandle);
        SetInputMode (impedDFLLHandle, IMPEDDFLL_SUBTRACT, 0);
        DeleteGraphPlot (impedDFLLHandle, IMPEDDFLL_GRAPH, -1, VAL_IMMEDIATE_DRAW);
        dfllimpedflag = 0;
      break;
    }
    return 0;
}

int CVICALLBACK DfllImpedSubtractFreq (int panel, int control, int event, void *callbackData, int eventData1,
                                       int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
```

```
        SetInputMode (impedDFLLHandle, IMPEDDFLL_SUBTRACT, 0);
        for (i = 0; i < numScans; i++)
        {
          waveforms[numScans*(dfllcurrchan-1)+i] +=
                -dfllamp[dfllmaxSamp] * sin(twopi*dfllpeakfreq*i/rate+dfllang[dfllmaxSamp]);
        }

      break;
    }
    return 0;
}

int CVICALLBACK DfllImpedPctWdw (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        GetCtrlVal (impedDFLLHandle, IMPEDDFLL_NUMFREQ, &dfllnumfreq);
        SetCtrlVal (impedDFLLHandle, IMPEDDFLL_WDWPCT, 800/dfllnumfreq);
      break;
    }
    return 0;
}

int CVICALLBACK DisplayFluxGraph (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
      DisplayPanel (fluxdisplayHandle);
      break;
    }
    return 0;
}

int CVICALLBACK DrawFluxGraphs (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        GetCtrlVal (fluxdisplayHandle, FLUXDISPLA_CHANNELCHOICE, &fluxdisplay);
        GetCtrlVal (fluxdisplayHandle, FLUXDISPLA_VOLTAGEON_OFF, &fluxdisplayVoltageOnOff);
        GetCtrlVal (fluxdisplayHandle, FLUXDISPLA_CURRENTON_OFF, &fluxdisplayCurrentOnOff);
        GetCtrlVal (fluxdisplayHandle, FLUXDISPLA_FLUXON_OFF, &fluxdisplayFluxOnOff);
        if (flagFreeImpedance == 1)
        {
          DeleteGraphPlot (fluxdisplayHandle, FLUXDISPLA_GRAPH, -1, VAL_IMMEDIATE_DRAW);
          if (fluxdisplay == 0 | fluxdisplay == 2)
          {
            if (fluxdisplayVoltageOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &vd[0], numScans-1,
                  VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_BLUE);
            }
            if (fluxdisplayFluxOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &lambdad[0],
                    numScans-1, VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_RED);
            }
            if (fluxdisplayCurrentOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &id[0], numScans-1,
                  VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_GREEN);
            }
          }
          if(fluxdisplay == 1 | fluxdisplay == 2)
          {
            if (fluxdisplayVoltageOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &vq[0], numScans-1,
```

```
                    VAL_DOUBLE, VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_CYAN);
            }
            if (fluxdisplayFluxOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &lambdaq[0], numScans-1,VAL_DOUBLE,
                    VAL_THIN_LINE, VAL_EMPTY_SQUARE, VAL_SOLID, 1, VAL_MAGENTA);
            }
            if (fluxdisplayCurrentOnOff == 1)
            {
              PlotY (fluxdisplayHandle, FLUXDISPLA_GRAPH, &iq[0], numScans-1, VAL_DOUBLE, VAL_THIN_LINE,
                    VAL_EMPTY_SQUARE, VAL_SOLID,1, VAL_PANEL_GRAY);
            }
          }
        }
      break;
    }
    return 0;
}

int CVICALLBACK HideFluxDisplayPanel (int panel, int control, int event, void *callbackData, int eventData1, int
eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        HidePanel (fluxdisplayHandle);
      break;
    }
    return 0;
}

int CVICALLBACK EstimateSpeed (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
    switch (event)
    {
      case EVENT_COMMIT:
        SetCtrlVal (panelHandle, PANEL_NUMSCANS, 12000);
        SetCtrlVal (panelHandle, PANEL_RATE, 600.0);
        rate = 600;
        numScans = 12000;
        AcquireCallback (panel, control, event, callbackData, eventData1, eventData2);
        SetCtrlVal (panelHandle, PANEL_ESTIMATEDSPEED, 0.0);
        SetCtrlVal (panelHandle, PANEL_SPEED_MSG, "\nplease wait");
        DisplayPanel (dftspeedestimateHandle);
        PreprocessCurrent (panel, control, event, callbackData, eventData1, eventData2);
        SetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, 1);
        PeakFreqCurrent (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (currentFFTHandle, CURRENTFFT_ESTSPEED, &tmp11[1]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_CURRA, tmp11[1]); /*displ ia speed estimate*/
        SetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, 2);
        PeakFreqCurrent (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (currentFFTHandle, CURRENTFFT_ESTSPEED, &tmp11[2]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_CURRB, tmp11[2]); /*displ ib speed estimate*/
        SetCtrlVal (currentFFTHandle, CURRENTFFT_SLIDE, 3);
        PeakFreqCurrent (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (currentFFTHandle, CURRENTFFT_ESTSPEED, &tmp11[3]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_CURRC, tmp11[3]); /*displ ic speed estimate*/
        PreprocessPower (panel, control, event, callbackData, eventData1, eventData2);
        SetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, 1);
        PeakFreqPower (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (powerFFTHandle, POWERFFT_ESTSPEED, &tmp11[4]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_POWA, tmp11[4]);    /*displ pa speed estimate*/
        SetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, 2);
        PeakFreqPower (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (powerFFTHandle, POWERFFT_ESTSPEED, &tmp11[5]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_POWB, tmp11[5]);    /*displ pb speed estimate*/
        SetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, 3);
        PeakFreqPower (panel, control, event, callbackData, eventData1, eventData2);
        GetCtrlVal (powerFFTHandle, POWERFFT_ESTSPEED, &tmp11[6]);
        SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_POWC, tmp11[6]);    /*displ pc speed estimate*/
```

```
SetCtrlVal (powerFFTHandle, POWERFFT_SLIDE, 5);
PeakFreqPower (panel, control, event, callbackData, eventData1, eventData2);
GetCtrlVal (powerFFTHandle, POWERFFT_ESTSPEED, &tmp11[7]);
SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_POWSUM, tmp11[7]);   /*displ psum spd estimate*/
PreprocessImpedance (panel, control, event, callbackData, eventData1, eventData2);
SetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, 1);
PeakFreqImpedance (panel, control, event, callbackData, eventData1, eventData2);
GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ESTSPEED, &tmp11[8]);
SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_IMPED, tmp11[8]);          /*displ imped spd estimate*/
SetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, 2);
PeakFreqImpedance (panel, control, event, callbackData, eventData1, eventData2);
GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ESTSPEED, &tmp11[9]);
SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_ANGLE, tmp11[9]);         /*displ angl spd estimate*/
SetCtrlVal (panelHandle, PANEL_FILTER_ORDER, 8);
SetCtrlVal (panelHandle, PANEL_FILTERTYPE, 1);
SetCtrlVal (panelHandle, PANEL_FILTER_CUTOFF, 40.0);
ApplyFilter (panel, control, event, callbackData, eventData1, eventData2);
PreprocessImpedance (panel, control, event, callbackData, eventData1, eventData2);
SetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, 1);
PeakFreqImpedance (panel, control, event, callbackData, eventData1, eventData2);
GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ESTSPEED, &tmp11[10]);
SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_IMPED_LP, tmp11[10]);
                                                /*displ imped spd estimate after 40Hz Chev LP*/
SetCtrlVal (impedanceFFTHandle, IMPEDFFT_SLIDE, 2);
PeakFreqImpedance (panel, control, event, callbackData, eventData1, eventData2);
GetCtrlVal (impedanceFFTHandle, IMPEDFFT_ESTSPEED, &tmp11[11]);
SetCtrlVal (dftspeedestimateHandle, DFTESTIMAT_S_EST_ANGLE_LP, tmp11[11]);
                                                /*display angle speed estimate after 40Hz Chev LP*/
/* The next block sets a permissible window around which the speed is
seen as to agree on several different channels. Then the channels are swept
one by one and the # of channels laying within the permissible window of the \
"pivot" channel is counted. The channel that has the largest amount of agreement
with other channels will be selected as the "right pivot". Then an average of all
estimated speeds laying within the permissible window is performed, giving the final
result. */
maxcnt = 0;
cnt[0] = 0;
for (j = 1; j < 12; j++)
{
  cnt[j] = 0;
  for (i = 1; i < 12; i++)
  {
    if ((tmp11[j]+wdw >= tmp11[i]) & (tmp11[j]-wdw <= tmp11[i]))
    {
      cnt[j]++;
      if (cnt[j] > cnt[maxcnt])
        maxcnt = j;
    }
  }
}
tmp1 = 0.0;
for (j = 1; j < 12; j++)
{
  if ((tmp11[maxcnt]+wdw >= tmp11[j]) & (tmp11[maxcnt]-wdw <= tmp11[j]))
    tmp1 += tmp11[j];
}
speedEstimate = tmp1 / cnt[maxcnt];
SetCtrlVal (panelHandle, PANEL_ESTIMATEDSPEED, speedEstimate);
if (cnt[maxcnt] > 7)
  SetCtrlVal (panelHandle, PANEL_SPEED_MSG, "\nvery good certainty");
else
  if (cnt[maxcnt] > 5)
    SetCtrlVal (panelHandle, PANEL_SPEED_MSG, "\ngood certainty");
  else
    if (cnt[maxcnt] > 3)
      SetCtrlVal (panelHandle, PANEL_SPEED_MSG, "\nfair certainty");
    else
      SetCtrlVal (panelHandle, PANEL_SPEED_MSG, "\nunlikely.");
NoFilter (panel, control, event, callbackData, eventData1, eventData2);
HidePanel (dftspeedestimateHandle);
```

```
    break;
  }
  return 0;
}

int CVICALLBACK HideDFTSpeedEst (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      HidePanel (dftspeedestimateHandle);
    break;
  }
  return 0;
}

int CVICALLBACK EstimateEfficiency (int panel, int control, int event, void *callbackData, int eventData1, int eventData2)
{
  switch (event)
  {
    case EVENT_COMMIT:
      EstimateSpeed (panel, control, event, callbackData, eventData1, eventData2 );
      SetCtrlVal (panelHandle, PANEL_RATE, 12000.0);
      rate = 12000;
      RateCallback (panelHandle, PANEL_RATE, EVENT_COMMIT, 0, 0, 0);
      AcquireCallback (panel, control, event, callbackData, eventData1, eventData2);
      PreprocessImpedance (panel, control, event, callbackData, eventData1, eventData2);
      EstimateRs (panel, control, event, callbackData, eventData1, eventData2);
      EstimateTrq (panel, control, event, callbackData, eventData1, eventData2);
    break;
  }
  return 0;
}
```

Appendix D:

D.1: IEEE Paper to be published in PES Electrical Manufacturing Coil Winding Association

Ernesto J. Wiedenbrüg, Alan K. Wallace, *'Induction Machine Speed Extraction by Analysis of Stator Current Signatures'*

D.2 IEEE Paper to be published in the IAS Pulp and Paper Conference

Alan K. Wallace, Ernesto J. Wiedenbrüg, *'Motor Efficiency Determination: From Testing Laboratory to Plant Installation'*

D.3 IEEE Paper published in the IAS Pulp and Paper Conference 1997

Ernesto J. Wiedenbrüg, Alan K. Wallace, Annette von Jouanne, *'Using the New EPRI/BPA Testing Facility to Evaluate the Latest Technologies in Motors and Drives'*

D.4 IEEE Paper published in the IEMDC 1997

Alan K. Wallace, A. von Jouanne, E. Wiedenbrüg, J. Douglass, C. Wohlgemuth, G. Wainwright *'A Laboratory Assessment of In-Service Motor Efficiency Testing Methods'*

D.5 IEE Paper published in the Electrical Machines and Drives Conference, Camebridge, 1997

A K. Wallace, A. R. von Jouanne, E. J. Wiedenbrüg, P. S. Andrews *'The Measured Effects of Under-Voltage, Over-Voltage and Unbalanced Voltage on the Efficiency and Power Factor of Induction Motors over Wide Ranges of Load'*

INDUCTION MACHINE SPEED EXTRACTION BY ANALYSIS OF STATOR CURRENT SIGNATURES

Ernesto J. Wiedenbrüg
Baker Instrument Company

Alan K. Wallace
Oregon State University

Abstract: This paper presents a method of accurately finding the steady state speed of an IM by analyzing its stator current signature. It also links these practical findings to a simplified model, which describes the modulations occurring in the mutual inductance of an IM caused by dynamic eccentricities.

Key words: Induction machine, speed, measurement, current signature, eccentricity.

I. INTRODUCTION

The speed at which an induction machine (IM) operates is frequently used in a plant environment as a rough estimation of its loading. The typical ways to measure this quantity are via pre-installed transducers, tachogenerators, optical sensors, stroboscopes, or magnetic pick-ups. All these methods share the disadvantage of intrusiveness, since the shaft of the machine may be either inaccessible (sunken pumps), or difficult to reach (enclosures and housings). In such settings it is still a simple task to measure the stator currents, which can be done either at the motor terminals or at the motor control cabinets.

Efforts in recent years have been increased to find ways of estimating the operating speed of IMs without the need of physical proximity. Classically, current signature analysis is employed, using frequency domain analysis. Each strategy focuses on a particular mechanism, which creates a side band on a certain location of the current frequency spectrum. The exact location of the peak contains information of the operating speed of the IM.

The most common scheme that is successfully employed focuses on side bands created by rotor slot harmonics [1-5]. An advantage to this particular scheme is that the frequencies which are being sought are of the kHz order, enabling the duration of data acquisition to be kept short. The main disadvantage of this technique, however, is that the rotor bar count has to be known, so that the location of the current side band can be translated into rotating speed. It is rare to find the rotor bar count on the nameplate, and frequently the nameplate itself may not be readily accessible. This makes methods based on rotor bar harmonics unsuitable for most speed estimation processes for line driven machinery.

Another method, based on the presence of dynamic eccentricity, yields to accurate speed estimations without the need of rotor bar count knowledge. [6-9] are previous works investigating this phenomenon. [10] is an approach that shows unorthodox signal processing techniques of dynamic eccentricity based current side band analysis.

This paper presents a simple model by which dynamic airgap irregularities can be simulated, decomposing the non constant airgap into a Fourier series, which rotates at rotor speed with respect to the stator. The rotor-stator mutual magnetic path crosses the airgap, hence is modulated by the time varying airgap size. The rotor-stator mutual inductance, assumed to be inverse proportional to the airgap size, is the only time-varying element of a single phase equivalent model of an IM.

This very simple model is of a comprehensive nature, yet is accurate enough to deliver current signatures for many different types of rotor eccentricities. The results of the model are compared with current signatures obtained from several IMs operated at a variety of load points.

II. ECCENTRICITY MODEL

A. Eccentricity cases:

The operating efficiency of IMs decays rapidly as the size of their air gaps increases. This causes IMs to be designed and built with the airgap bounded by realizable manufacturing tolerances. The resulting air gaps are not of a constant size around the rotor, but have inherent tolerances. The resulting inaccuracies are commonly called eccentricities, and can be divided into static and dynamic forms. Static eccentricities can be caused by a misalignment of the rotor rotational and symmetry axis with respect to the bore axis, or by out-of-round components of the stator bore. Dynamic eccentricities are those components which either describe the misalignment of a round rotor's symmetry axis with respect to its own rotational axis, or the components that describe the out-of-roundness of the rotor. These four cases of eccentricity are represented schematically in

Fig. 1. The out of round eccentricities that are shown in Fig. 1 are simpler than will be encountered in real life, since the circles and ellipses are mathematical shapes, not possible to manufacture with perfect accuracy.

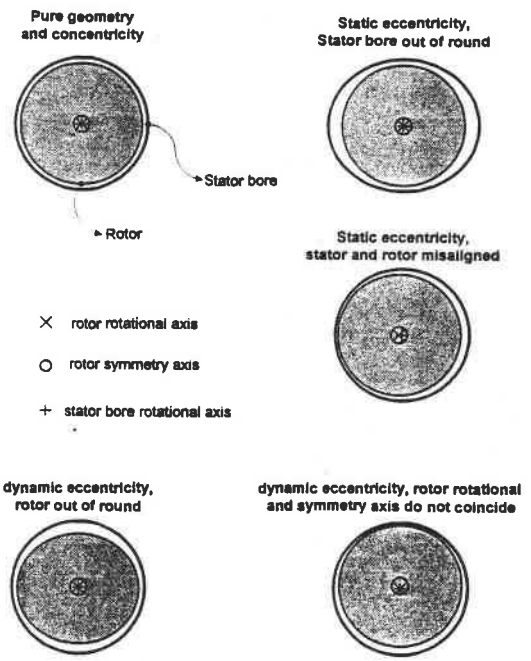


Figure 1: Static and dynamic eccentricity cases.

The airgap in the "out-of-round" cases is the resulting difference between the circular and elliptic shapes. This can also be formulated as two concentric circles, one of which has its radius modulated by a sinusoid of low amplitude and of a rotational frequency equal to twice the angle in polar coordinates.

B. Mutual inductance model:

The presented analysis investigates the effect of dynamic eccentricities on the stator current signatures. The irregular airgap size of the IM can be modeled by an infinite Fourier series. Fourier series decomposition was chosen over other decomposition methods because the rotor irregularities repeat themselves every 2π, as shown in Fig. 2. Hence, the airgap can be described as a function of time and space as follows

$$airgap(\alpha,t)=\sum_{j=0}^{\infty}F_j\cdot\cos\left(j\cdot\alpha+j\cdot\varphi_j+j\cdot\omega_r\cdot t\right) \qquad (1)$$

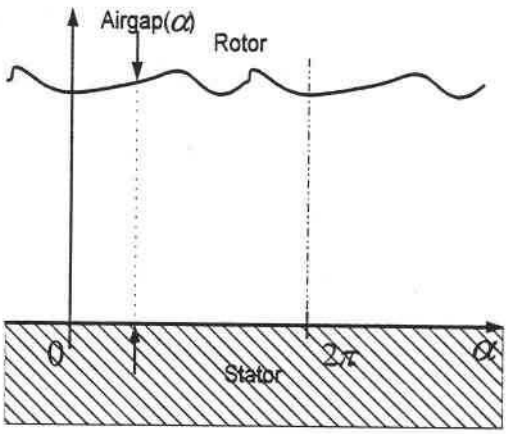


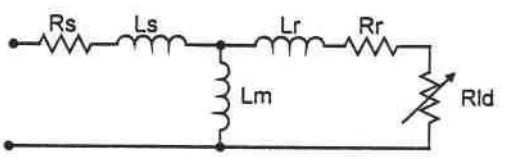
Figure 2: Dynamic eccentricity of higher order.

In (1) ω_r represents rotational speed of the rotor, α is the physical location around the stator, the integer j is the particular component of the Fourier series, and φ_j is the phase shift of that component. The stator-rotor mutual inductance is inversely proportional to the airgap length, with which the mutual inductance becomes a function of time and space as follows:

$$L_m(t)=\frac{k_m}{P}\cdot\sum_{u=1}^{P}\frac{1}{\sum_{j=0}^{\infty}F_j\cdot\cos\left(j\cdot\omega_r\cdot t+\frac{j\cdot\varphi_j}{P}+\frac{u\cdot j\cdot 2\pi}{P}\right)} \qquad (2)$$

where P is the pole pair number, and u is a second variable that ensures proper superposition for series wound motors.

Figure 3: Per-phase equivalent circuit of IM.



The modulated inductance is utilized in a simplified model, where it is put the per-phase equivalent circuit of an IM, as shown in Fig. 3.

The values for stator and rotor resistances, leakage inductances and stator-rotor mutual inducantces are taken from [11], page 190, for the 4-pole 50hp motor case. By changing *Rld*, different load conditions can be

set. The choice of *Rld* in conjunction with *Rr* define the steady state operating speed, which is set to be constant, assuming a large inertia load. The operating speed is needed to calculate the time variation of the magnitude of *Lm* (2).

C. Results of the model

A 1st order eccentricity is simulated first: Fig. 4 simulates a misalignment of the rotor's rotational axis with its symmetry axis. The magnitude of the 1st order Fourier dynamic eccentricity component is 5%. The resulting current signature is shown in Fig. 4.

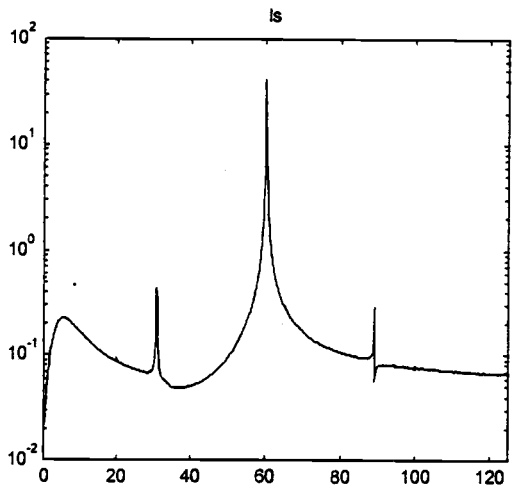


Figure 4: Predicted current signature for 1st order eccentricity, 4 pole motor.

The location of the side bands can be used to calculate the speed accurately, since, as stated in [7-9], they appear at exactly $fs \pm fr$, where fs and fr are the supply frequency and the rotational speed in *rps* respectively. For compatibillity, the units of f_s and f_r must be Hz and revolutions per second.

Fig. 5 shows the current signatures for a case with pure 2nd order eccentricities. This means, that the rotor symmetry axis and the rotor rotational axis coincide, but that the outer shape of the rotor has a slightly ellyptical shape. Again, the relative airgap excentricity is of 5%.

It is important to point out that a 1st order eccentricity will be the main component generated by bent shafts, and that the unbalanced pull generated through its own presence would aggravate its magnitude to some extent. This phenomenon will not occur with such significance for the mechanisms producing a 2nd order eccentricity.

The current signatures that have been gathered experimentally, some of which are presented in part III, show a significant 1st order eccentricity, and any 2nd

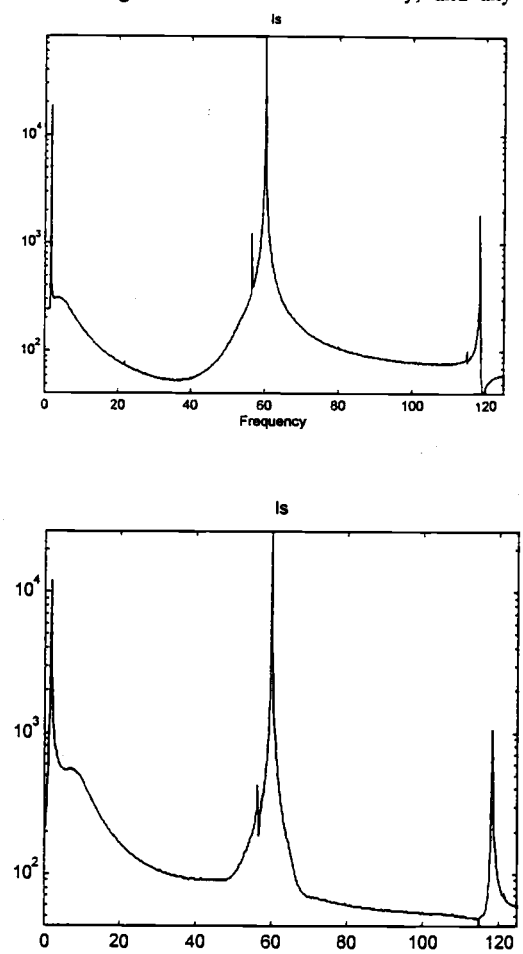


Figure 6: Predicted current signature for 1st order eccentricity, 2 pole motor.

Additionally, this model permits the simulation of any pole pair number. Fig. 6 shows the results of a 1 pole pair case, and Fig. 7 of a 3 pole pair case, both with a pure 5% 1st order eccentricity and the result is again a set of prevailing current signature side bands located at $fs \pm fr$ in both cases.

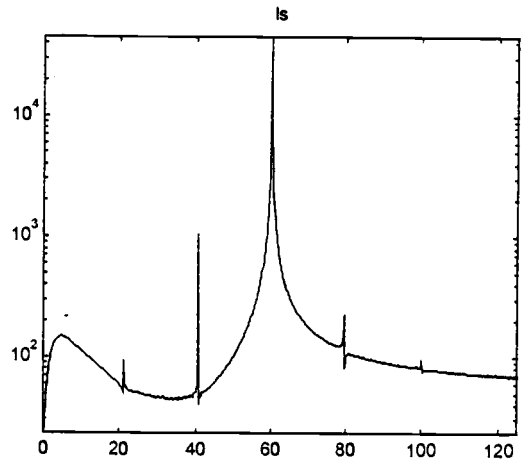


Figure 7: Predicted current signature 1st order eccentricity, 6 pole motor.

Interestingly the current signature created by a 2 pole machine, with pure 1st order eccentricity (Fig. 6), resembles closely the current signature that is created by a 4 pole machine with a pure 2nd order eccentricity (Fig. 5). The reason resides in the fact that the eccentricity itself, namely the frequency with which the offset sine wave travels with respect to the stator, is the same for these two cases. A sine wave of one period around the rotor, on the 2 pole machine, is rotating with twice the speed than a sine wave of two periods around the rotor for the 4 pole machine. Hence, in both cases, a stator coil would see a modulation of the same frequency on the airgap.

III. Experimental Results:

A. Experimental Setup:

Two different designs of .75hp three phase IMs were tested on a full load motor test system, manufactured by Baker Instrument Company. The motor test system permits loading down the IM under test to any steady state load level. The data acquisition system that was used was developed on LabWindows/CVI, by National Instruments. The data acquisition hardware was comprised of a National Instruments multi-channel A/D card with 12 bit resolution, and Hall effect based current sensors by Nana. Tektronix differential voltage probes are used for generating input voltage signals, which are not used in the instantaneous current analysis, but for displaying rms based operating conditions. The parameters that the operator can set are sampling rate and number of samples. The signals are low pass filtered, using a 5th order Bessel switched capacitor low pass, prior to reaching the A/D card. The acquired data is used to calculate rms stator quantities, which is the last part of the standard data acquisition process.

B. 4 Pole machine:

In order to ensure that the spectral calculations are done with the same algorithms, the acquired data is stored into a file, which is read by Matlab. The experimental results show that the simulation predicted the current side bands in the correct locations. Fig. 8 shows the results obtained for a 4 pole .75 hp Reliance IM, which was running at 80% load.

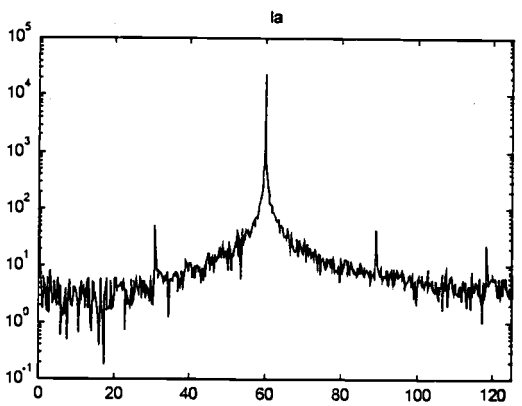


Figure 8: Experimental current signature: 4 pole motor.

It becomes apparent, that the side bands from Fig. 8 and the side bands from Fig. 4 coincide to the largest extent. The relative amplitude difference from the fundamental of each graph to its amplitude of the side bands entails information on the severity of that particular order of eccentricity. The ratio from fundamental to side band of Fig. 4 is of roughly 2 decades, and was calculated for a 5% eccentricity. In Fig. 8 one sees over 2.5 decades, which means that the severity of eccentricity should be a little less than 5%. In order to achieve more quantitative accurate numbers it would be necessary to write a simulation model which includes stator-stator mutual inductances, and which would not neglect the time-modulation caused to the leakage inductances by the constantly changing air gap size.

C. 6-pole machine:

A .75hp 6-pole Reliance motor was also tested on the motor test system, in order to corroborate the model by changing the parameter of pole pair number. Fig. 9 shows the current signatures obtained for the tested 6-pole machine.

Obviously, in the current signature the $fs \pm fr$ side bands are clearly visible. In this case the side bands of interest

are located close to 40Hz and 80Hz. But, comparing with Fig. 8, they are much smaller for this machine. This points to the fact that the evaluated 6-pole machine has a smaller level of dynamic eccentricity.

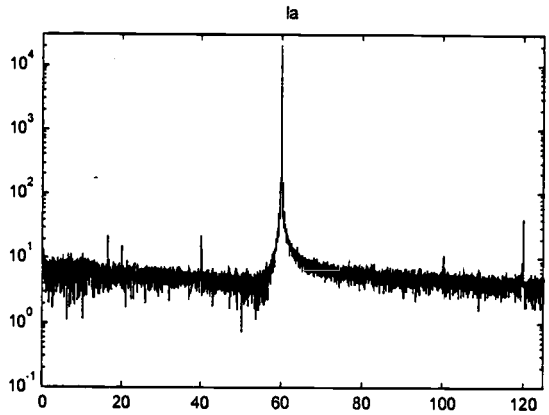


Figure 9: Experimental current signature: 6 pole motor.

The size of the side bands, even in this logarithmic representation, is sufficient to obtaine consistent, reliable and accurate speed reading.

IV. CONCLUSIONS

A simple, comprehensive yet accurate dynamic eccentricity model has been developed. The model entailed a per-phase equivalent circuit of an IM, where all components of the circuit apart from the mutual inductance, were constant. The mutual inductance was modulated according to an infinite Fourier series, which represented the time varying airgap size as a function of time and location. The results of the simulations show, when put into perspective with experimental data, gathered from 4-pole and 6-pole IMs, that the mechanisms creating the current side bands can be modeled as dynamic eccentricities. It has also been shown that the different orders of eccentricity can be modeled independently from each other, and also for any number of pole pairs. This feature is key to the ability of identifying which type of dynamic eccentricity occurs predominantly on real IMs.

A main finding of the presented work is that, when comparing different orders of eccentricity in the simulation with actual test results, the prevailing order of eccentricity found in physical machines is the 1st order. Additionally it has been shown that an IM with an even number of pole pairs still shows 1st order eccentricity. The reason for this occurrence is that the mutual inductance is not linearly proportional to the airgap. If this was the case, then diametrically opposing poles would negate the effect of 1st order of eccentricity respectively.

Future work will search for a quantitative measure of eccentricity by analyzing the gravity of the current side bands, which might be a useful tool for base line production testing QA measures. A more elaborate IM model is currently being developed, which will make more detailed calculations possible. This model will not neglect stator-stator mutual inductances, nor the time variance of stator and rotor leakage as a function of the fluctuating airgap.

V. REFERENCES:

1. K. D. Hurst, T. G. Habetler, G. Griva and F. Profumo, '*Speed sensorless field-oriented control of induction machines using current harmonic spectral estimation*', IEEE IAS proceedings 1994 vol. I, pp. 601.

2. K. D. Hurst and T. G. Habetler, '*A comparison of spectrum estimation techniques for sensorless speed detection in induction machines*', IEEE IAS proceedings 1995 vol. I, pp. 553.

3. J. F. Brudny and D. Roger, '*Induction machine speed sensor based on stator current measurement*', Power electronics and variable speed drives, Conference publication IEE 1996.

4. A. Ferrah, P. J. Hogben-Laing, K. J. Bradley, G. M. Asher and M. S. Woolfson, '*The effect of rotor design on sensorless speed estimation using rotor slot harmonics identified by adaptive digital filtering using the maximum likelihood approach*', IEEE IAS proceedings 1997, session 3-3.

5. A. Barbour and W. T. Thomson, '*Finite element study of rotor slot designs with respect to current monitoring for detecting static airgap eccentricity in squirrel cage induction motors*', IEEE IAS proceedings 1997, session 3-3.

6. M. Brandford, '*Unbalanced magnetic pull in a 6-pole induction motor*', IEE Proceedings 1968, Vol. 115 No. 11, pp. 1619.

7. D. Dorrell, '*Calculation of unbalanced magnetic pull in cage induction machines*', PhD thesis 1993, University of Cambridge.

8. D. G. Dorrell, W. T. Thomson and S. Roach, '*Combined effects of static and dynamic eccentricity on airgap flux waves and the application of current monitoring to detect dynamic eccentricity in 3-phase induction motors*', Electrical machines and drives, IEE 1995.

9. D. G. Dorrell, W. T. Thomson and S. Roach, *'Analysis of airgap flux, current and vibration signals as a function of the combination of static and dynamic airgap eccentricity in 3-phase induction motors'*, IEEE IAS proceedings 1995, vol. I., pp. 563.

10. P. Pillay and Z. Xu, *'Motor current signature analysis'*, IEEE IAS proceedings 1996, vol. I., pp. 587.

11. Paul C. Krause, Oleg Wasynczuk, Scott D. Sudhoff, *'Analysis of Electric Machinery'*, IEEE Press, New York, 1995.

Ernesto J. Wiedenbrúg holds a Dipl. Ing. from Aachen, Germany, and is currently working as an R&D engineer at Baker Instrument Company, CO. He has worked as an intern for Siemens SA in Buenos Aires, Argentina and as a power engineer for ISCOR, in VanderBjilpark, Rep. of South Africa. He is finishing his doctoral degree from Oregon State University, during which he had a one year fellowship from Volkswagen AG Germany, and was subsequently employed as the general manager of the Motor Systems Resource Facility, an EPRI funded center.

Alan K. Wallace received the degrees of B.Eng (Hons) and Ph.D. from the University of Sheffield, UK. His industrial experience over 11 years includes positions with Imperial Chemical Industries (UK), Spar Aerospace (Toronto), and the Urban Transportation Development Corporation of Canada. He has taught in the University of Nottingham (UK) and is currently Professor and Head of Department of Electrical & Computer Engineering at Oregon State University. Dr. Wallace is a member of the IEE (London) and a Senior Member of IEEE.

MOTOR EFFICIENCY DETERMINATION: FROM TESTING LABORATORY TO PLANT INSTALLATION

Alan K. Wallace
Oregon State University
wallace@ece.orst.edu

Ernesto J. Wiedenbrüg
Baker Instrument Company
ernesto@peakpeak.com

Abstract: IEEE 112b has become the method of choice for laboratory environment efficiency testing. However, no strategy for low level intrusion testing of field applications has been able to achieve the same level of widespread acceptance. This paper compares the goals, techniques employed and key points of interest of the standardized IEEE 112b laboratory test with a technique using minimum level of intrusion for field testing of Induction Machines. Development of the latter has been completed and is currently being tested.

Key words: Induction machines, Testing, Measurement, Measurement standards, Laboratories.

VI. INTRODUCTION

The IEEE 112b Standard is currently the baseline for laboratory testing of Induction Machines (IMs). It was designed to accurately measure the performance of IMs vs. load. As such it provides loss segregation and other performance values including operating temperature, power factor, speed and line currents as a function of load. The method was designed to be transportable, i.e. it is possible for evaluations performed in different facilities to yield very comparable results on the same motors, as demonstrated in "round robin" series of tests.

The level of intrusion that is required for performing an IEEE 112b test is too high for field testing, since controllable voltage sources and controllable loads are required. These conditions are very rarely obtained in the field, hence IEEE 112b does not represent an option for industrial site testing of IMs. However, there remains a need for knowledge of operating efficiency of machinery for proper plant management and good predictive maintenance programs in an industrial environment.

In an attempt to identify suitable field efficiency estimation methods, several techniques have been tested in a study conducted by Bonneville Power Administration and Pacific Gas and Electric. The first part of this study involved identification and assessment of expected accuracies [1]. The second part of the project entailed an extensive testing program at the Motor Systems Resource Facility MSRF, an EPRI funded laboratory [2-3]. The testing of those efficiency estimation methods investigated the level of intrusion they required, and their achieved level of accuracies. The deficiencies identified became the driving force in developing a new efficiency estimation method, with the intention of decreasing the level of intrusion, while increasing the accuracies of the predicted efficiencies.

This paper compares methodology, instrumentation, testing procedures and the information obtained using the laboratory IEEE 112b method of measuring the motor efficiency with the viable field method of efficiency estimation. The first part of this paper describes the differences in the required instrumentation and physical setup of both methods. In the second part, the different questions faced by a laboratory test with regard to a field test are evaluated. The third part investigates the particular methods by which the efficiency results are obtained, as well as the theoretical background on which they are based.

VII. REQUIRED INSTRUMENTATION

It has been mentioned that the methods compared in this paper are suitable for different environments. On one hand, a laboratory environment with advanced and substantial instrumentation, and with controllable loads and voltage conditions is required. On the other a field application, where the intent is to disrupt production as little as possible before, during and after measurements. These two implementations differ widely in their physical setup. Typical examples are shown in the following figures.

D. IEEE 112b

Fig. 1 is a schematic of a typical configuration for a test bed designed to perform IEEE 112b tests. A micro-ohmmeter can be used to gather the stator resistances, and a controllable voltage source is needed to set balanced operating voltages from 125% down to roughly

10% of rated conditions. The electrical input has to be measured, either with rms voltage, current and power transducers, or via instantaneous voltage and current transducers, the signals from which then can be used to calculate real power and power factor digitally. Torque and speed measurements are also needed, since the data required to calculate the results of a 112b is input and output based. Ambient temperature sensing is also needed.

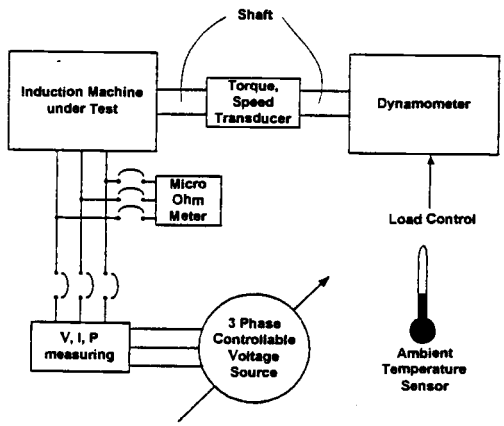


Figure 1: Physical setup of a test bed suitable for 112b tests

E. Field efficiency testing:

Fig. 2 shows a typical configuration for the field application of a proposed efficiency estimator. Neither motor connection, nor operating condition have to be changed.

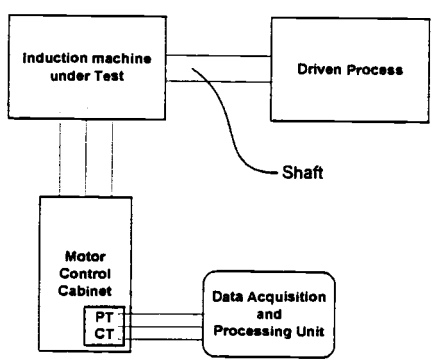


Figure 2: Typical implementation for the proposed field efficiency estimator

No sensors are required other than voltage and current monitoring, which, for other cases than in low voltage applications, ought to be connected on the low power side of potential transducers (PT) and current transducers (CT). This may impose obvious restrictions for the resulting accuracy caused by these transformations. For low voltage applications, the connections are done at the motor control center, or at the terminals of the machine, depending on accessibility. No other special instrumentation is required for the physical setup, and no uncoupling is needed either. Output power is not measured as this technique is based on input power measurement and output power estimation.

VIII. Differences in Sought Information

Both methods that are currently analyzed share the goal of efficiency measurement. However the difference in the environments in which the information is looked for is the source of deep differences, not only in the setup and expected accuracy, but also in the set of questions that are asked by each method.

A. IEEE 112b:

IEEE 112b is a laboratory environment testing method, designed to have a maximum of accuracy and repeatability. According to [4], a well calibrated laboratory, with a well performed IEEE 112b may reach overall accuracies within 1%. As with any loss segregation method, assumptions have to be made when separating the different losses. These assumptions are an integral part of the testing standard, ensuring portability. Since a laboratory is, by definition, a measurement environment, it permits both, a complex instrumentation setup and also the performance of tests at operating conditions other than the restricted options available in an industrial environment.

The method IEEE 112b has been designed to characterize the capability of the machine under test, at rated voltage conditions and as a function of load for steady state operation. The standard describes acceptable unbalance conditions, and acceptable levels of harmonic distortions. These specifications demand a clean voltage supply, which translates into both, a best case and also a repeatable scenario. In order to achieve repeatability, all the calculations must be corrected to a base line ambient temperature, which has been chosen to be 40°C.

The tables with results from a 112b test (Appendix A) include the following information, printed for every one of the operating points 25%, 50%, 75%, 100%, 125% and 150% of rated load:

- Efficiency, power factor, speed and line currents.
- Loss segregation and operating stator temperature.
- Operating torque and total loss.

The achieved information characterizes the capabilities of the tested machine, and describes the allocation of the losses into the classes of stator copper losses, core losses, friction and windage losses, rotor copper losses and stray load losses. The loss segregation allows some insight into the tradeoffs chosen by the designer, and, in some instances, permit identification of abnormal loss distributions, which may point to possible machine damage.

B. Field efficiency testing:

The field efficiency testing method that is presented here does not share identical goals of investigation with IEEE 112b method. Since it is a field viable method, it is not a realistic option to perform a test on a machine with a similar level of intrusion and complex testing setup as is performed with the IEEE 112b. This fact obviously translates into concessions that have to be made in terms of accuracy. Additionally, it can be proven difficult for some implementations to reach the machine that is to be tested, a sunken pump being a prime example.

The questions asked in industrial settings focus on the motor's actual performance under the given conditions, including voltage unbalances, harmonic distortions, over and under voltage conditions and torque variations under actual ambient temperature conditions. Typically, a plant manager's interest resides in whether that motor is actually performing according to expectations. The main issues to be identified in an industrial setting, are the following:

- Voltage condition
- Operating efficiency
- Operating mechanical power
- Operating speed
- Operating torque

1. Voltage condition:

With the knowledge of voltage condition it is possible to identify the maximum percentage of rated mechanical output, suggested by NEMA, for a particular supply unbalance [5]. Information on voltage symmetrical components should also be used, since a substantial negative sequence component has a significant heating effect on the motor, reducing its life significantly. Additionally, total harmonic distortion, harmonic bar charts and crest factor information permit identifying further problems in the supply condition and allow for corrective actions, which is part of adequate preventive maintenance programs.

2. Operating efficiency:

Obviously, knowing the operating efficiency of a motor allows comparison with acceptable baseline efficiency numbers for that particular implementation. The rising cost of energy makes exchange of low performance machinery economically advantageous. Pay back periods clearly below a year are possible in some applications which now can be identified.

3. Operating mechanical power:

The importance of knowing the operating power is twofold. First, knowing the power output of an IM enables comparison with its rating, allowing the identification of motor-load mismatches, which can waste large amounts of energy. Second, if field efficiency testing is performed as a part of plant maintenance, it is possible to build up historic files on performance of the load of any process. Sudden increase in requested load of any process which has not changed may point to problems with the process being driven. This particular asset allows identification of some problems in the mechanical systems driven by the machines, expanding predictive maintenance capabilities.

4. Operating speed:

The knowledge of operating speed can be used in conjunction with a history file. If the load driven did not change substantially since the last maintenance period, but the operating speed dropped, then closer attention has to be paid to the machine. It is possible that the rotor cage of the driven machine is showing degradations.

5. Operating Torque:

With the proposed method it is possible to calculate the operating torque, not only in a steady state, but also as a function of time. This feature allows for identification of Electro-mechanical oscillations, which put undue stress on the mechanical components, as well as on the machine. Misalignments, bent shafts and coupling problems can be identified by experienced operators from the frequency spectrum of the torque signal.

IX. Methods and Theoretical Background

While the IEEE 112b method is based upon rms calculations and identification of particular losses of the machine, the field efficiency estimation method is a result of instantaneous signal processing.

A. IEEE 112b:

1. Required Measurements

The set of measurements to be performed for a IEEE 112b, if based on a setup similar to the one described in Fig. 1, are the following:

- Cold stator resistance
- Cold no load performance
- No load saturation characteristic
- Load run data

The no load saturation test is performed with descending voltages. This process is performed in steps, so that there are at least ten voltage conditions available from the rated voltage conditions down to the lowest voltage, which are used to segregate friction and windage losses from core losses, as described in [6].

The load run consists of seven operating points. Standard are the uncoupled (no load), and the 25%, 50%, 75% and 100% operating points. Additionally two more operating points are required above 100% rated operation. Frequently 125% and 150% rated are chosen.

2. Data Processing

Operating temperature for the stator is calculated with the gathered data for each load point. The stator I^2r_s loss is calculated from the input currents at the different loads, and the average of the stator resistances. The power crossing the airgap is equal to the stator input power minus core loss and stator copper loss. The rotor I^2r loss is found by multiplying the power across the airgap with slip. Adding core loss, stator and rotor copper loss to friction and windage loss, results in the total conventional loss, also printed as a function of load. Stray load losses are obtained by subtracting the total conventional losses from the subtraction of input electrical power minus output mechanical power. A spreadsheet with results from a performed IEEE 112b is shown in the appendix of this paper.

B. Field efficiency testing:

1. Required measurements

The measurements necessary for the field efficiency testing are very simple if compared to the IEEE 112b method. Input currents and input voltages are the only quantities that need to be measured, but this data is required on an instantaneous basis for future calculations.

2. Data Processing

The field efficiency estimation method is based on an input output approach. Input power is calculated by the measured electrical input voltages and currents, while the electrical output power is found by the multiplication of operating speed times torque.

Speed is found by looking at modulations which occur in the frequency spectrum of the measured input currents. These modulations are caused by inherent imperfections of the rotor, which is always going to be slightly out of round. These imperfections modulate the airgap size as a function of the speed, effectively varying the mutual and stray inductances of the equivalent circuit diagram. [7] shows, in a single phase diagram model, how the speed dependent modulations of airgap size introduce side bands in the frequency domain of the stator currents, which' location is used for very accurate speed extraction. Sidebands of typical stator currents can be seen in Fig. 3.

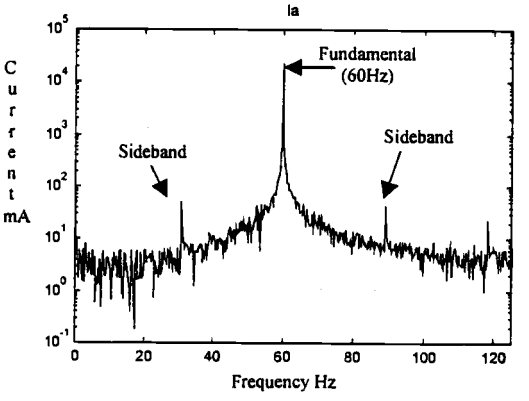


Figure 3: Stator current Spectrum of a 4-pole IM.

The torque calculations are based upon the set of equations that are commonly used in vector drive theory, in which the IM is modeled as a two axis machine in the *dq* domain. Torque can be calculated by the cross multiplication of the stator flux vector and the current vector [8].

Friction, windage and stray load losses also have to be estimated. However it has been found that these losses do not vary to a large extent from design to design, when compared among machines of similar rating and pole number. In this case, clearly, concessions are made with regards of accuracy when compared with the IEEE 112b method. However, the accuracies achieved by this field method clearly exceed the operator's needs, permitting proper identification of the subjects of interest in the field previously mentioned. Extensive test performed on

a laboratory environment showed that the employed method of speed extraction reaches consistent accuracies within 1r/min, while the estimated torque predictions commonly achieve accuracies of 2% or better for 50% or higher loading. At loads of 25%, however, the inaccuracies double.

Finally, if it was felt that more accurate measurements or loss segregation were required for field applications, then the only option available would be to send the machine to a laboratory, to perform an IEEE 112 test on it.

X. CONCLUSIONS AND FUTURE WORK

This paper presents the differences when comparing laboratory testing of IMs to state of the art field testing. It is important to point out that the questions of interest for field personnel differ sharply from the questions asked to laboratories, and so do the requested accuracies of the results obtained.

While the laboratory environment is faced with the challenge of achieving supreme accuracies and transportable data, the main challenge in the field resides in balancing the tradeoff between low intrusion measurements and achievable accuracies within that particular premise. At the same time, while the laboratory environment seeks to quantify the capabilities of a particular motor, the field is interested in the performance of that motor under particular given conditions. Consequently the methods tend to complement each other's scope.

Alpha and beta testing sites are currently being identified for testing the field efficiency estimator approach that has been presented in this paper. The results obtained from field testing of this method will be published as soon as the data is available.

XI. REFERENCES:

[1] J. S. Hsu, J. D. Kueck, M. Olszewski, D. A. Casada, P. J. Otaduy, L. M. Tolbert, *'Comparison of Induction Motor Field Efficiency Evaluation Methods'*, IEEE IAS Proceedings 1996 vol. 1, pp. 703-712.

[2] A. Wallace, A. von Jouanne, E. Wiedenbrüg, J. Douglass, C. Wohlgemuth, G. Wainwright, *'A Laboratory Assessment of In-Service Motor Efficiency Testing Methods'*, IEEE IEMDC Proceedings 1997, WC 1.7-1.

[3] John G. Douglass, *'Efficacy of Methods for Estimating In-Service Motor Efficiency'*, Washington State University, Cooperative Extension Energy Program, June 1997.

[4] EASA Technical Manual, © 1996, pp. 8-15, 8-16.

[5] NEMA *'Motors and Generators'*, MG 1, Part 14, page 8.

[6] IEEE *'Standard Test Procedure for Polyphase Induction Motors and Generators'*, IEEE Std 112.

[7] Ernesto J. Wiedenbrüg, Alan K. Wallace, *'Steady State Speed Extraction for Induction Machines Analyzing Instantaneous Stator Current Signatures'*, IEEE PES, Electrical Manufacturing Coil Winding Association 1998.

[8] Krause, Wasynczuk, Sudhoff, *'Analysis of Electrical Machinery'*, IEEE Press, New York, 1995, pp. 169-177.

APPENDIX A:

Type:	Y802-4	Design:	0	Frame:	0	hp:
Freq:	60	Volts:	460	Synch. r/min:	1800	Serial No:
Deg C Temp Rise:	90.1			Model No:	y-802-4	Phase:

Stator Winding Resistance Between Terminals		19.7000	@	24.9 deg C			
Specified Temperature for Resistance Correction (ts) =		115.0 deg C					
Item	Load, in % rated	150	125	100	75	50	25
1	Ambient Temperature, in C	25.3	25	25	24	24	24
2	(tt) Stator Winding Temperature, in C	98.5	78.7	63.8	55.9	48.8	42.3
3	Slip, in /min	168	124	87	61	36	20
4	Speed, in r/min	1632	1676	1713	1739	1764	1780
5	Line-to-Line Voltage, in V	460	460	460	460	460	460
6	Line Current, in A	2.42	2.01	1.72	1.45	1.23	1.14
7	Stator Power, in W	1600	1280	1025	765	495	320
8	Core Loss, in W	50	50	50	50	50	50
9	Stator I^2*R Loss, In W, at (tt) C	222	145	100	69	49	41
10	Power Across Air Gap, in W	1328	1085	875	646	396	229
11	Rotor I^2*R Loss, in W	124	75	42	22	8	3
12	Friction and Windage Loss, in W	20	20	20	20	20	20
13	Total Conventional Loss, in W	416	189	212	161	127	113
14	Torque, in Nm	6.6	5.32	4.235	3.203	1.96	1.063
15	Dynamometer Correction, in Nm	0.0	0.0	0.0	0.0	0.0	0.0
16	Corrected Torque, in Nm	6.60	5.32	4.24	3.20	1.96	1.06
17	Shaft Power, in W	1128	934	760	583	362	198
18	Apparent Total Loss, in W	472	346	265	182	133	122
19	Stray-Load Loss, in W	56	157	53	21	6	9
Int.	3.58 Slope 2.084	Corr. Factor		0.88		Del. Pt.	Y
20	Stator I^2*R Loss, in W, at (ts) C	233	161	117	84	60	51
21	Corrected Power Across Air Gap, in W	1317	1069	858	631	385	219
22	Corrected Slip, in r/min	176	138	102	73	44	25
23	Corrected Speed, in r/min	1624	1662	1698	1727	1756	1775
24	Rotor I^2*R Loss, in W, at (ts) C	129	82	49	26	9	3
25	Corrected Stray-Load Loss, in W	90.8	59.0	37.4	21.4	8.0	2.4
26	Corrected Total Loss, in W	523	372	273	201	147	126
27	Corrected Shaft Power, in W	1077	908	752	564	348	194
28	Shaft Power, in hp	1.4	1.2	1.0	0.8	0.5	0.3
29	Efficiency, in %	67.3	70.9	73.3	73.7	70.3	60.5
30	Power Factor, in %	82.9	79.6	74.4	66.0	50.0	35.0

Summary of Characteristics

Load, in % of rated	25	50	75	100	125	150
Power Factor, in %	35.0	50.0	66.0	74.4	79.6	82.9
Efficiency, in %	60.5	70.3	73.7	73.3	70.9	67.3
Speed, in r/min	1775	1756	1727	1698	1662	1624
Line Current, in A	1.14	1.23	1.45	1.72	2.01	2.42

Using the New EPRI/BPA Testing Facility to Evaluate the Latest Technologies in Motors and Drives

Ernesto Wiedenbrug Alan Wallace Annette von Jouanne

Oregon State University
Motor Systems Resource Facility
Elect. & Compt. Engr. Dept.
Corvallis, OR 97331-3211

Abstract - This paper describes the capabilities of a new facility that has been developed on the campus of Oregon State University by support from the Electric Power Research Institute (EPRI) and the Bonneville Power Administration (BPA). The purpose of the Motor Systems Resource Facility (MSRF) is to promote the effective and efficient use of electric motors, drives and generators by providing consulting and analysis services backed by test and evaluation capabilities rated up to 300 hp (225 kW). The laboratory is designed to be both highly flexible in its capabilities and to be of maximum efficiency through the use of a bi-directional, unity power factor, active rectifier dynamometer converter. Extracts from recent test projects are given including a comparison assessment of switched-reluctance and induction motor drives and an investigation of in-service efficiency measurement techniques.

1. Introduction

With the increased emphasis on improved efficiency and control of motor drives, many industries are investigating alternative energy saving approaches such as high efficiency motors and Adjustable Speed Drives (ASDs) for reasons of economic competitiveness. However, with increased application of ASDs, several issues must be considered such as the thermal effects of non sinusoidal power supply to the motor, the effect of high dv/dt, and the effect of the ASD on the service power quality [1-8].

In addition, switched reluctance motors (SRMs), permanent magnet motors (PMMs), hybrid forms of SRMs and PMMs [9], and several forms of doubly-fed motors [10,11] are being developed for niche applications where more conventional drives are not appropriate or are not competitively priced. Several unconventional machine configurations are also currently being investigated for use with renewable energy resources such as wind power [12,13]. This paper describes a high power motor systems resource facility capable of evaluating the performance of these advancing technologies and investigating problems arising from their adoption and application. To illustrate the range of capabilities of the MSRF, two recent test projects are described briefly: first a comparison assessment of SRMs with ASD/induction motor drives and second an investigation of in-service efficiency measurement techniques.

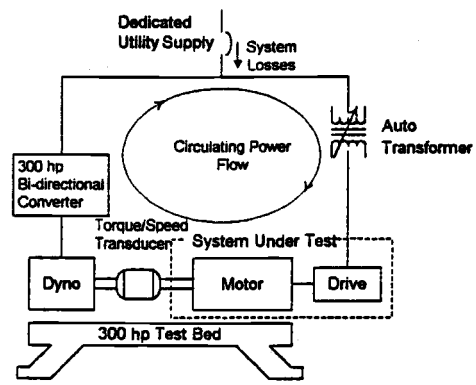


Fig. 1 Regenerative system design with circulating power flow.

2. Requirements of a Contemporary Testing Facility

The purpose of the Motor Systems Resource Facility is to promote the effective and efficient use of electric motors, drives and generators by providing:

- test and evaluation expertise and facilities
- assessment of alternative technologies
- research on advanced maclines and converters
- assessment of related power quality issues
- training in contemporary equipment and techniques
- compilation of a knowledge base
- consultations and communication of the above

The focus of the facility is a laboratory for testing machines and converters up to 300 hp (225 kW). The laboratory is designed to be both highly flexible in its capabilities (both motors and generators from 1 hp to 300 hp can be evaluated) and to be of maximum efficiency (only the inherent losses of the system under test and the dynamometer are dissipated). The key to this capability is a bi-directional, unity power factor, active rectifier dynamometer converter.

The schematic of Fig. 1 indicates how the fully line-regenerative converter enables the most efficient form of testing possible. The greater part of the energy being processed circulates between the system under test and the dynamometer system. Thus, only the total losses are

drawn from the utility, which are typically 10 to 20% of the rating of the machine being tested.

When operating in this mode it is essential to obtain good power quality at the terminals of the dynamometer converter in order to prevent any interference with the instrumentation or the system being tested. Pollution of the supply by harmonics will otherwise corrupt the performance of the system under test. Fig. 2 shows the overall laboratory schematic and Fig. 3 shows typical voltage and current waveforms observed at the converter terminals at full load. Since completion of the test facility, a variety of tests have been conducted indicating that the regenerated power contains less than 5% Total Harmonic Distortion (THD), and that the dynamometer does not cause unstable oscillations with the unit under test.

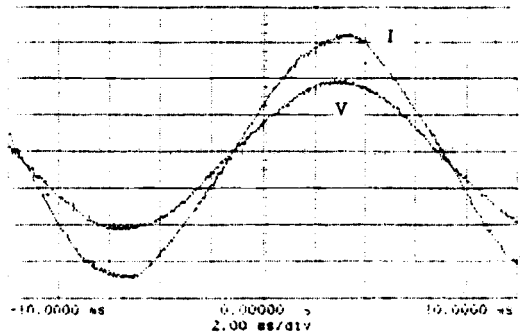


Fig. 3 Full load voltage and current at converter terminals.

In addition to clean, stable efficient operation, the testing facility provides the following:

- industrially relevant ratings
- wide range of service voltages, with capability for under-, over- and unbalanced voltages
- wide range of machine frame sizes, with the capability to accommodate unusual geometries
- wide range of machine speeds, with capability for electrically locked-rotor and programmable dynamic speed changes
- ability to test both motors and generators, while minimizing the dissipated energy
- ability to test power electronic converters, with investigations of nonsinusoidal and high frequency effects
- ability to make accurate measurements of the above, with provision for redundant instrumentation techniques where possible

Appendix I details how each of these specifications are met in the new MSRF. The following two sections are extracts from recent test projects conducted in the MSRF.

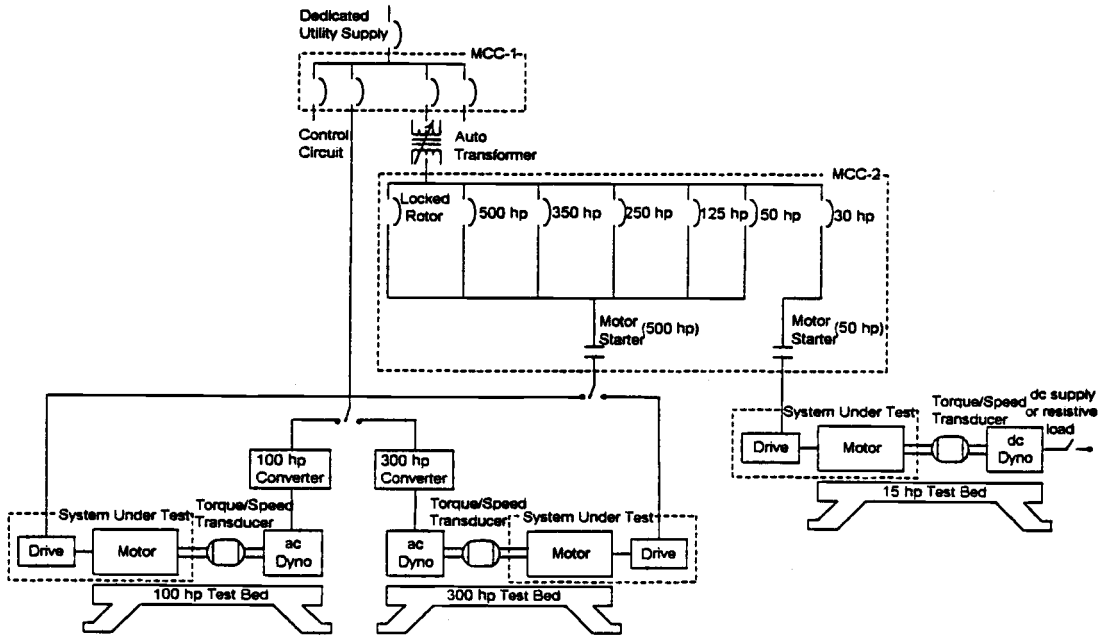


Fig. 2 Complete laboratory controls.

Table 1 Simulated Pump Profile Test Points

Speed, r/min	900	1100	1300	1400	1500	1600	1700	1800	2000
Torque, lb.in.	88	131	183	218	243	277	312	350	432
Power, hp	1.3	2.3	3.8	4.7	5.8	7.0	8.4	10.0	13.7

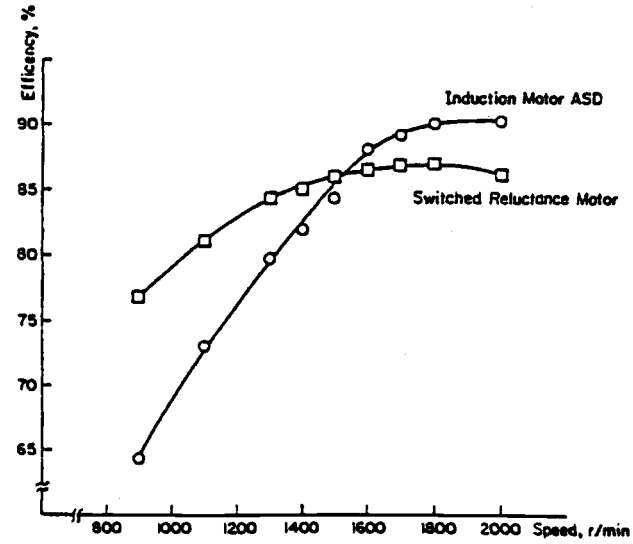


Figure 4. Efficiency Comparison (Pump Profile Test).

3. A Comparative Assessment of Switched-Reluctance and Induction Motor Drives

Based on many years of refinement in design, construction and reliable service, the induction motor is the industrial choice for fixed speed applications. In the last two decades power electronic adjustable frequency converters have enabled adjustable speed operation of squirrel-cage induction motors to an extent that they now operate in many functions previously only met by dc motors [1,2]. The refinements of power electronics have caused some investigators to propose a system approach to ASD and motor design which has led to several unconventional motor developments [9,11,14]. The most commercially advanced of these 'new' motors is the switched reluctance motor (SRM) which has now been under development for more than twenty years, and has been examined for its potential for application in industry on several occasions [15].

Recently a major US manufacturer has begun development work on industrial SRMs and has supported Bonneville Power Administration (BPA) and Seattle City Light in a project to assess the potential of these motors. As part of this exercise the MSRF performed a comparison of the performance of an SRM drive with that of an induction motor ASD. The complete findings are to be published by BPA in 1997. This test program investigated the performance of the prototype SRM with that of an insulated gate bipolar transistor (IGBT), pulse-width modulated (PWM), ASD and high efficiency induction motor to represent the best that is commercially available. For part of this project a test profile was developed, as given in Table 1, to simulate the loads anticipated by a small centrifugal pump, requiring 10 hp at 1800 r/min. Performance features investigated include input power quality and complete system (terminal to shaft) efficiency. Details of the latter are shown in Fig. 4. As anticipated, the induction motor efficiency is very good close to its rated condition but decreases rapidly with reducing speed (and load). In contrast the SRM efficiency holds up much better at lower loads. Consequently, if significant periods of operation are anticipated at lower speeds the SRM may be an appropriate choice from an energy perspective.

4. Investigation of In-Service Efficiency Measurement Techniques

It is estimated that more than 60% of the electrical energy being used in the U.S. is consumed by motors [16]. In addition, in 1992 the National Energy Policy Act (EPACT'92) was passed to promote increased industrial efficiency worldwide; this includes a provision that after October of 1997, all newly constructed motors must be high efficiency [17]. Note that motors running at full load most of the time will incur an annual energy cost far exceeding their initial price [18]. It is therefore essential for industries to ensure that their motors are operating in an effective and energy efficient manner.

The motor efficiency value, calculated as the ratio of the mechanical output to the electrical input, provides the basis for operating cost comparisons that could support

replacing a motor with a more efficient unit offering a major reduction in operating cost. In order to determine the true motor operating efficiency, it is necessary to measure the electrical input power from the terminal voltages and currents and the mechanical output power from the shaft torque and speed. However, the mechanical output power of installed motors can be difficult to obtain in a non-intrusive manner, and even more difficult to verify [18,19].

In response to the need of estimating installed motor efficiencies, several different in-service testing methods and special measurement tools have been developed. In March of 1996, a team at the Oak Ridge National Laboratory (ORNL), under contract to the Bonneville Power Administration (BPA) and Pacific Gas and Electric (PG&E), reviewed 28 of these proposed "estimate methods" and evaluated them according to the invasiveness and cost of equipment. Based on this review, the Washington State University Co-Operative Energy Extension Program (WSUCEEP), under contract with BPA and PG&E, subcontracted with the MSRF to test the 12 most promising in-service motor efficiency estimation methods.

The 12 selected methods can be classified into three categories: those that use dedicated, purpose developed instruments; those that take test data from generally available instruments and employ software models; basic principle or generic methods. Table 2 shows the short-listed candidate methods which are in the process of being compared to laboratory input/output based measurements that have been obtained.

4.1 Dedicated Instrument Methods

The Vogelsang & Benning (V&B) method I requires testing at three conditions: uncoupled, normally loaded and unpowered (off). A reflector must be attached to rotating equipment to allow the speed to be recorded. In Option II, testing is accomplished without uncoupling, and motor nameplate data is substituted, but the accuracy is assumed to be reduced.

The Vectron method was developed for the Electricity Corporation of New Zealand (ECNZ). It requires testing at a load <10% and a load > 50%, and unpowered. It also uses an optical tachometer based on an attached reflective strip. The manufacturer claims that the efficiency at full load conditions can be determined from testing at a load >50% but less than full load. The manufacturer also claims this method can correct for off nominal voltage conditions up to 5%.

The MAS-1000 method is based upon a tester developed by Niagara Instruments. The use of a magnetic reluctance speed sensor is preferred with this method, although a strobe tachometer can be substituted if readings are manually entered. The system is based on an Intel 486 processor and the manufacturer claims that Motor Master Plus (MM+) software developed by the U.S. Department of Energy (DOE) could be loaded and used with the tester for additional convenience in motor systems management.

4.2 Software Methods

The Esterline Angus method requires only generic equipment and custom software. Tests are required while uncoupled, at normal load, and unpowered. Surface and ambient temperatures are required.

The three Motor Master Plus (MM+) methods are based upon nameplate and normal load operation. They allow efficiency to be computed/estimated at the normal load. Electrical and/or speed readings are required at normal load. No uncoupled or unpowered readings are required.

The Oak Ridge Motor Efficiency and Load 96 (ORMEL96) method requires only nameplate information and a speed measurement. The speed can be taken with a strobe tachometer. A computer program is used to process the data.

4.3 Generic Methods

The three slip methods require only a speed reading and, in one case, a voltage reading. All readings are at normal load.

Note that the more accurate estimate methods require shaft speed measurements, some of which may be difficult to obtain in an industrial environment.

Table 2 Motor Efficiency Testing Methods

Motor Efficiency Testing Methods	Tests Required			
	No Load	Normal Load	Off	Speed Meas.
Dedicated Instruments				
V&B Option I	X	X	X	X
V&B Option II		X	X	X
Vectron (ECNZ)	X	X	X	X
MAS-1000	X	X	X	X
Software				
Esterline Angus	X	X	X	X
MM+ Power		X		
MM+ Current		X		
MM+ Slip		X		X
ORMEL96		X		X
Generic				
Standard Slip		X		X
O.H. Comp. Slip		X		X
Upper Bound Slip		X		X

4.4 Motors Evaluated

Four different motors have been used for the evaluation (50 hp, 100 hp, 150 hp, and 300 hp). The 50 hp and 300 hp motors are "perfect" motors which are used for laboratory set-up calibration. The 150 hp motor had a rotor eccentricity problem. The 100 hp motor had deliberately introduceable defects of a "dropped" turn on any one of all three stator phases (i.e. a turn of a winding that can be deliberately disconnected from the circuit).

The reason for introducing the defective motors is that some of the methods have built-in assumptions about performance characteristics that are approximations of "perfect" motor parameters. The dynamometer method should enable an assessment of the accuracies regardless of the condition of the motor.

The load points where efficiency has been documented include 25%, 50%, 75%, and 100% of rating for the 300 hp motor. For the smaller and intermediate hp motors, additional load points were performed at 125% and 150% of rating, which were not possible on the 300 hp motor due to the initial rating limitations of the lab. Tests were also conducted during 10% over-voltage, 10% under-voltage, and 1% and 2.5% unbalanced voltage conditions for all the motors.

Thus, with four motors, one of which can have balanced or unbalanced turns per phase, each evaluated at five voltage conditions, with no-load, 25%, 50%, 75%, 100% (for 300 hp) plus 125% and 150% of rated load (for the 150 hp, 100 hp [in both configurations] and 50 hp) 175 load points were determined and recorded.

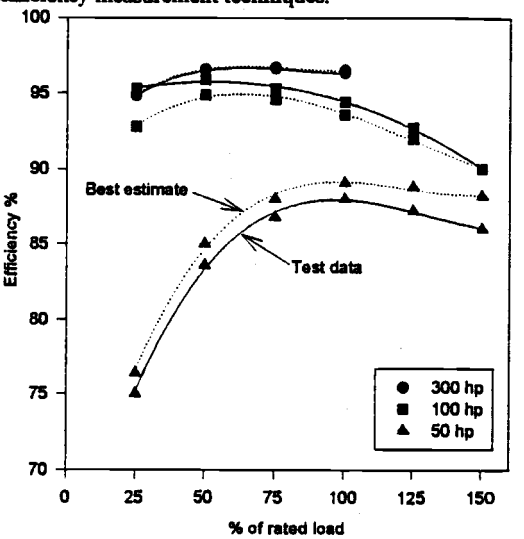
4.5 Preliminary Findings

The initial results of the test program are very mixed in their findings. In some cases the correlation between laboratory test data and efficiency estimations is good but, depending upon the level of accuracy sought, these tend to be exceptions rather than a general rule. An example of the best correlations obtained for each of the three motors (300 hp, 100 hp, and 50 hp) is shown in Fig. 5: it should be emphasized here that it is not the same estimation method for these three examples that produces the best correlation. As a general trend, however, it appears that the estimation methods perform better with the larger motors than the smaller ones, and with new motors rather than repaired ones. A far more thorough investigation is in process by WSUCEEP to isolate these specific details.

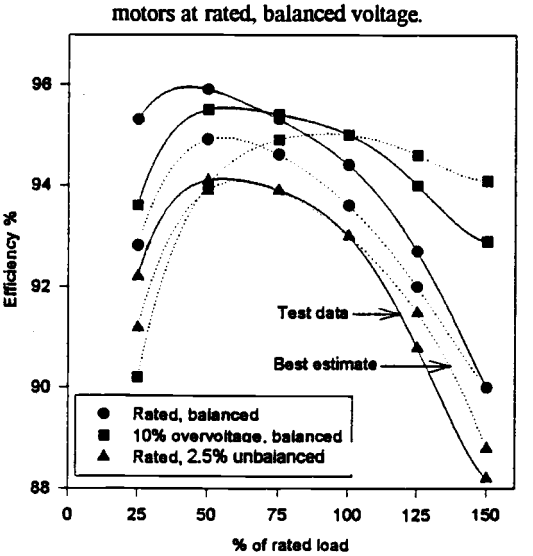
Fig. 6 provides comparable data for the 100 hp motor with balanced windings operating for three different voltage conditions (rated balanced, 10% overvoltage balanced, and rated 2.5% unbalanced). As only the data for one motor is being presented here a significantly expanded scale is possible. For some purposes the correlation between the test data and the best estimate data may be adequate. Again it is emphasized that the best correlation, as presented in Fig. 6, is not for the same estimation method for all three conditions. However, in general the correlation for balanced operation, and 2.5% unbalanced operation appear better than for the 10% overvoltage operation.

5. Conclusions

The need to provide an accurate, flexible and highly energy efficient laboratory for the testing of motors, drives and generators can be met by the new Motor Systems Resource Facility (MSRF) at Oregon State University described in this paper. Wherever possible redundant systems provide cross checks of the measurements. For example, power analyzers confirm the validity of permanently installed current, voltage and power transducers and the dynamometer control system can confirm the torque/speed transducer measurements. It was shown that the high overall efficiency of the system is obtained by recirculating the useful power within the laboratory, thus requiring the utility service only to make up the losses of the equipment under test and the dynamometer system. Finally, the capabilities of the MSRF are illustrated by extracts from recent test projects: a comparison assessment of switched-reluctance and induction motor drives and an investigation of in-service efficiency measurement techniques.



5 Comparison of test data and best estimates for three motors at rated, balanced voltage.



6 Comparisons of test data and best estimates for 10(hp motor .

Acknowledgments

Financial support from the Electric Power Research Institute (EPRI) and the Bonneville Power Administration (BPA) for the development of the test facilities is acknowledged, in addition to the support provided by BPA, Pacific Gas & Electric (PG&E) and U.S. Motors/Emerson for the development of the presented projects.

References

[1] A. von Jouanne, P. Enjeti, W. Gray, *Application Issues for PWM Adjustable Speed AC Motor Drives*, IEEE Industry Applications Magazine, Sept./Oct. 1996, pp. 10-18.

[2] A. Bonnett, "Analysis of the Impact of PWM Inverter Waveforms on AC Induction Motors", IEEE Trans. on IA, Vol. 32, No. 2, March/April 1996, pp. 386-392.

[3] J. M. Erdman, R. J. Kerkman, D. W. Schlegel, G. L. Skibinski, "Effect of PWM Inverters on AC Motor Bearing Currents and Shaft Voltages", *IEEE Trans. Ind. Appl.*, vol. IA-32, no. 2, pp. 250-259, 1996.

[4] E. Zhong, T. A. Lipo, "Improvements in EMC Performance of Inverter-Fed Motor Drives", *IEEE Trans. Ind. Appl.*, vol. IA-31, no. 6, pp. 1247-1256, 1995.

[5] R. H. Daugherty, C. H. Wennerstrom, "Need for Industry Standards for AC Induction Motors Intended for Use with Adjustable-Frequency Controllers", *IEEE Trans. Ind. Appl.*, vol. IA-27, no. 6, pp. 1175-1185, 1991.

[6] Electric Power Research Institute, Adjustable Speed Drive Office, 1996, "ASD Master," computer software system

[7] H. P. Nee, "Rotor Slot Design of Inverter-Fed Induction Motors," *IEE Electric Machines & Drives*, No 412, pp. 52-56, 1995.

[8] S. Williamson, C. I. McClay, "Optimisation of the Geometry of Closed Rotor Slots for Cage Induction Motors," *IEEE IAS 30th Annual Meeting Conference Record*, pp. 507-514, 1995.

[9] B. Sarlioglu, Y. Zhao, T. A. Lipo, "A Novel Doubly-Salient Single-Phase Permanent-Magnet Generator," *IEEE IAS 29th Annual Meeting Conference Record*, pp. 9-15, 1994.

[10] A. Masmoudi, A. Toumi, M. B. A. Kamoun, M. Poloujadoff, "A Numerically Controlled Test Bench for Experiments with a Doubly-Fed Synchronous Machine," *International Conference on Electrical Machines*, pp. 617-622, 1994

[11] R. Li, A. K. Wallace, R. Spée, C. Alexander, "Synchronous Drive Performance of Brushless Doubly-Fed Motors," *IEEE Transactions Industrial Applications*, pp. 963-970, 1994.

[12] Z. Chen, E. Spooner, 1995, "A Modular, Permanent-Magnet Generator for Variable-Speed Wind Turbines," *IEE Electrical Machines & Drives*, No 412, pp. 453-457, 1995.

[13] M. Boger, A. K. Wallace, "Performance Capability Analysis of the Brushless Doubly-Fed Machine as a Wind Generator," *IEE Electrical Machines & Drives*, No 412, pp. 458-461, 1995.

[14] A. K. Wallace, R. Spee, "Performance Evaluation of AC Adjustable Speed Drives", Conf. Rec., IEEE IAS Annual Meeting, pp. 463-467, 1987.

[15] W. F. Ray, P. J. Lawrenson, R. M. Davis, J. M. Stephenson, N. N. Fulton, R. J. Blake, "High-Performance, Switched-Reluctance, Brushless Drives", IEEE Trans. Ind. Appl., vol IA-22, No 4, pp. 722-730, 1986.

[16] A. D. Little, Inc. "Efficiency standards in commercial and industrial electric motors and equipment", Contract CO-04-50127-00, Case #78537, Jan. 1976.

[17] United States House of Representatives, "Energy Policy Act of 1992", report 102-1018.

[18] R. L. Nailen, "Finding true power output isn't easy", *Electrical Apparatus*, Feb. 1994.

[19] A. Bonnett, "An Update on AC Induction Motor Efficiency", IEEE Trans. on IA, Vol. 30, No. 5, Sept./Oct. 1994, pp. 1362-1372.

Appendix 1
MSRF Capabilities

Able to test devices up to 300 hp

- Motors
- Generators
- Converters and Controllers
- Instrumentation

Fully regenerative system

- Meets IEEE 519
- Dynamometer >95% efficient at rated load

Input power

- 750 KVA supply
- 0 to 600 VAC
- 3 phase, balanced or unbalanced

Mechanical specifications

- 300 hp
- 15,000 lb.in. torque
- 0 to 4000 rpm
- Bi-directional rotation
- Full load testing over entire speed range

- 15hp
- 550 lb.in. torque
- 1750/2300 rpm
- bi-directional rotation

Four Quadrant Dynamometer Converter

- Motor or Generator action
- Vector control for full load testing over entire speed range
- Programmable Torque and Speed Modes
    1. Steady state
    2. Torque vs. Speed profiles
    3. Torque vs. time profiles
    4. Speed vs. time profiles

Instrumentation

- Steady-state RMS and instantaneous
- Redundant electrical and mechanical measurements
- 3 independent data acquisition systems
  - Steady state, RMS
  - Local instantaneous (transient analysis)
  - Voltech PM 3300 based
- wide range of hall effect current sensors
  - LEM: 2000A, 1000A, 500A, 300A, 200A, 100A
  - Ohio Semitronics: 4000A, 1000A, 600A, 100A
- wide range of speed/torquemeters
  - 200, 500, 1k, 2k, 5k, 10k lb.in.
  - 22000 rpm
- calibrator unit for in-house instrument calibration

Special Projects

- Experimental and unique configurations

A Laboratory Assessment of In-Service Motor Efficiency Testing Methods

A. Wallace* A. von Jouanne* E. Wiedenbrug* J. Douglass** C. Wohlgemuth*** G. Wainwright****

Oregon State University*
Motor Systems Resource Facility
Elect. & Compt. Engr. Dept.
Corvallis, OR 97331-3211
Tel: (541) 737-1867
Fax: (541) 737-0771

Washington State** University
Co-Operative Extension Energy Program
P.O. Box 43165
Olympia, WA 98504

Bonneville Power*** Administration
P.O. Box 3621-MPMT
905 N.E. 11 AVE.
Portland, OR 97208

Pacific Gas & Electric**** Company
2303 Camino Ramon, Suite 200
San Ramon, CA 94583

Abstract - Determining in-service motor efficiencies is important to industries concerned with energy conservation and cost savings. However, non-intrusive efficiency measurements of installed motors can be difficult to obtain and even more difficult to verify. This paper describes an evaluation and comparison of twelve motor efficiency methods through laboratory testing to assess their accuracy and precision. Analysis of the methods and implementation procedures is discussed.

1. Introduction

It is estimated that more than 60% of the electrical energy being used in the U.S. is consumed by motors [1]. In addition, in 1992 the National Energy Policy Act (EPACT'92) was passed to promote increased industrial efficiency worldwide; this includes a provision that after October of 1997, all newly constructed motors much be high efficiency [2]. Note that motors running at full load most of the time will incur an annual energy cost far exceeding their initial price [3]. It is therefore essential for industries to ensure that their motors are operating in an effective and energy efficient manner. Fig. 1 illustrates how electric energy is distributed for various driven equipment [4].

The motor efficiency value, calculated as the ratio of the mechanical output to the electrical input, provides the basis for operating cost comparisons that could support replacing the motor with a more efficient unit offering a major reduction in operating cost. In order to determine the true motor operating efficiency, it is necessary to measure the electrical input power from the terminal voltages and currents and the mechanical output power from the shaft torque and speed. However, the mechanical output power of installed motors can be difficult to obtain in a non-intrusive manner, and even more difficult to verify [3,5-7].

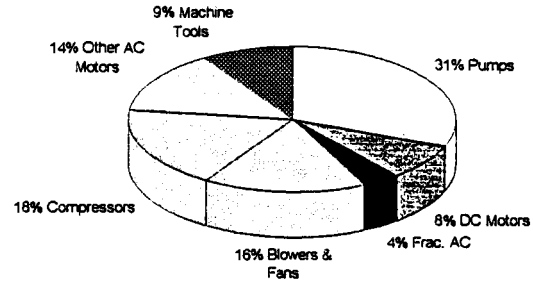


Fig. 1 Distribution of electric motor equipment.

In response to the need of estimating installed motor efficiencies, several different in-service testing methods and special measurement tools have been developed. In March of 1996, a team at the Oak Ridge National Laboratory (ORNL), under contract to the Bonneville Power Administration (BPA) and Pacific Gas and Electric (PG&E), reviewed 28 of these proposed "estimate methods" and evaluated them according to the invasiveness and cost of equipment [8,9]. Based on this review, the Washington State University Co-Operative Energy Extension Program (WSUCEEP), under contract with BPA and PG&E, subcontracted with the Motor Systems Resource Facility (MSRF) at Oregon State University (OSU) to test the 12 most promising in-service motor efficiency testing methods.

This paper discusses an impartial laboratory evaluation that has been conducted to assess the accuracy and precision of the final 12 methods. With the equipment available to the Motor Systems Resource Facility (MSRF) at OSU, the necessary measurements can be made with a high degree of accuracy and confidence by electric power analyzers and non-contact torque/speed transducers [10] in a manner similar to that employed by many motor manufacturers. The 12 in-service motor efficiency methods will be tested and the estimated efficiencies will be compared with the results of the laboratory baseline efficiencies found according to IEEE 112B. In addition, analysis of the methods and implementation procedures are discussed. From these results, the most appropriate method, or methods, for in-service motor efficiency estimation can be determined by the WSUCEEP team. This paper describes the test program and presents some of the preliminary findings.

2. In-Service Motor Efficiency Testing Methods

Table I Motor Efficiency Testing Methods

	Tests Required			
Motor Efficiency Testing Methods	No Load	Normal Load	Off	Speed Meas.
Dedicated Instruments				
V&B Option I	X	X	X	X
V&B Option II		X	X	X
Vectron (ECNZ)	X	X	X	X
MAS-1000	X	X	X	X
Software				
Esterline Angus	X	X	X	X
MM+ Power		X		
MM+ Current		X		
MM+ Slip		X		X
ORMEL96		X		X
Generic				
Standard Slip		X		X
O.H. Comp. Slip		X		X
Upper Bound Slip		X		X

Table I presents the 12 efficiency measurement methods recommended for testing at the MSRF. These represent nine different processes (with derivatives), three of which have specialized measurement equipment that is not already part of the system of sensors used in the IEEE 112B standard. Most of the methods involve different algorithms or computer programs to perform calculations on nameplate data and data gathered with generic instruments.

2.1 Dedicated Instrument Methods

The Vogelsang & Benning (V&B) method I requires testing at three conditions: uncoupled, normally loaded and unpowered (off). A reflector must be attached to rotating equipment to allow the speed to be recorded. In Option II, testing is accomplished without uncoupling, and motor nameplate data is substituted, but the accuracy is assumed to be reduced.

The Vectron method was developed for the Electricity Corporation of New Zealand (ECNZ). It requires testing at a load <10% and a load > 50%, and unpowered. It also uses an optical tachometer based on an attached reflective strip. The manufacturer claims that the efficiency at full load conditions can be determined from testing at a load >50% but less than full load. The manufacturer also claims this method can correct for off nominal voltage conditions up to 5%.

The MAS-1000 method is based upon a tester developed by Niagara Instruments under the direction of Vern Nielsen who was one of the designers of the motor test platforms in the Ontario Hydro Technology Lab (OHT) in Toronto, Canada and in the Industrial Electrotechnology Laboratory (IEL) in Raleigh, NC. A magnetic reluctance speed sensor is used, although a strobe tachometer can be substituted if readings are manually entered, but the former is preferred. The system is based on an Intel 486 processor and the manufacturer claims that Motor Master Plus (MM+) software developed by the U.S. Department of Energy (DOE) could be loaded and used with the tester for additional convenience in motor systems management.

2.2 Software Methods

The Esterline Angus method requires only generic equipment and custom software. Tests are required while uncoupled. at normal load, and unpowered. Surface and ambient temperature are required.

The three Motor Master Plus (MM+) methods are based upon nameplate and normal load operation. They allow efficiency to be computed/estimated at the normal load. Electrical and/or speed readings are required at normal load. No uncoupled or unpowered readings are required.

The Oak Ridge Motor Efficiency and Load 96 (ORMEL96) method requires only nameplate information and a speed measurement. The speed can be

taken with a strobe tachometer. A computer program is used to process the data.

2.3 Generic Methods

The three slip methods require only a speed reading and, in one case, a voltage reading. All readings are at normal load.

Note that the more accurate estimate methods require shaft speed measurements, some of which may be difficult to obtain in an industrial environment.

3. Laboratory Equipment and Testing

The baseline for the comparisons of the in-service testing methods outlined in Section 2, are determined from the MSRF laboratory dynamometer. At the input terminals, this consists of a power analyzer to measure the electrical quantities of voltage and current and, hence, calculate power and power factor. At the motor shaft the torque and speed were measured by the most appropriate of a range of mechanically interchangeable non-contact transducers.

Four different motors have been used for the evaluation (50 hp, 100 hp, 150 hp, and 300 hp). The 50 hp and 300 hp motors are "perfect" motors which are used for laboratory set-up calibration (see Appendix 1). The 150 hp motor has a rotor eccentricity problem. The 100 hp motor has deliberately introduceable defects of a "dropped" turn on any one of all three stator phases (i.e. a turn of a winding that can be deliberately disconnected from the circuit). The reason for introducing the defective motors is that some of the methods have built-in assumptions about performance characteristics that are approximations of "perfect" motor parameters. The dynamometer method should enable an assessment of the accuracies regardless of the condition of the motor.

The motors have been tested on the motor/dynamometer platform in the MSRF laboratory which is summarized in Appendix 2, and described in detail in a previous paper [10]. The load points where efficiency has been documented include 25%, 50%, 75%, and 100% of rating for the 300 hp motor. For the smaller and intermediate hp motors, additional load points were performed at 125% and 150% of rating, which were not possible on the 300 hp motor due to the initial rating limitations of the lab. Tests were also conducted during 10% over-voltage, 10% under-voltage, and 2% and 5% unbalanced voltage conditions for all the motors.

Thus, with four motors, one of which can be balanced or unbalanced turns per phase, each evaluated at five voltage conditions, with no-load, 25%, 50%, 75%, 100% (for 300 hp) plus 125% and 150% of rated load (for the 150 hp, 100 hp [in both configurations] and 50 hp) approximately 165 load points were determined and recorded.

4. Preliminary Findings

The initial results of the test program are very mixed in their findings. In some cases the correlation between laboratory test data and efficiency estimations is good but, depending upon the level of accuracy sought, these tend to be exceptions rather than a general rule. An example of the best correlations obtained for each of the three motors (300 hp, 100 hp, and 50 hp) is shown in Fig. 2: it should be emphasized here that it is not the same estimation method for these three examples that produces the best correlation. As a general trend, however, it appears that the estimation methods perform better with the larger motors than the smaller ones, and with new motors rather than repaired ones. A far more thorough investigation will be completed by WSUCEEP to isolate these specific details.

Fig. 3 provides comparable data for the 100 hp motor with balanced windings operating for three different voltage conditions (rated balanced, 10% overvoltage balanced, and rated 5% unbalanced). As only the data for one motor is being presented here a significantly expanded scale is possible. For some purposes the correlation for all three conditions may be adequate. Again it is emphasized that the best correlation, as presented in Fig. 3, is not for the same estimation method for all three conditions. However, in general the correlation for balanced operation, and 5% unbalanced operation appear better than for the 10% overvoltage operation.

5. Conclusion

With the increased emphasis on energy/cost savings, it is important for industries to be able to conduct motor efficiency measurements in the field. In this paper, 12 in-service motor efficiency methods have been presented and tested in the Motor Systems Resource Facility at Oregon State University to assess their accuracy and precision. From these results, the most appropriate means for in-service motor efficiency estimation can be

determined. Note that the required tolerance for the efficiency estimate varies with the application, i.e. for a plant operator to be within 3% would be acceptable, whereas a manufacturer or rewind shop may need the efficiency estimate to be within 1%.

In some cases the findings of the efficiency estimation studies indicate that the correlation with test data is somewhat irregular and an instrument, or generic method, may perform better for some motors than for others. This obviously presents a dilemma for users at present but indicates that the estimation techniques have potential to give adequate indication with more development or refinement. However, the need to keep the techniques simple and user friendly, for use in an industrial environment, should not be forgotten. In summary, the findings of this study reinforce the opinions of Reference [3], entitled, "Finding True Power Output Isn't Easy". Although the results are promising enough to indicate that reliable efficiency estimation methods can be developed.

Acknowledgments

Financial support from the Bonneville Power Administration and Pacific Gas & Electric for the development of this work is acknowledged, as is the support from the Electric Power Research Institute in the development of the test facilities. The authors also wish to thank Mr. Page Andrews for his endurance while conducting the testing program.

References

[1] A. D. Little, Inc. "Efficiency standards in commercial and industrial electric motors and equipment", Contract CO-04-50127-00, Case #78537, Jan. 1976.

[2] United States House of Representatives, "Energy Policy Act of 1992", report 102-1018.

[3] R. L. Nailen, "Finding true power output isn't easy", *Electrical Apparatus*, Feb. 1994.

[4] A. Bonnett, "An Update on AC Induction Motor Efficiency", IEEE Trans. on IA, Vol. 30, No. 5, Sept./Oct. 1994, pp. 1362-1372.

[5] C. Becnel, J. Kilgore, E. Merrill, "Determining Motor Efficiency by Field Testing", IEEE Trans. on IA, Vol. 23, No. 3, May/June 1987, pp. 440-443.

[6] P. Cummings, W. Bowers, W. Martiny, "Induction Motor Efficiency Test Methods", IEEE Trans. on IA, Vol. 17, No. 3, May/June 1981, pp. 253-272.

[7] S. Chen, S. Yeh, "Opmimal Efficiency Analysis of Induction Motors Fed by Variable-Voltage and Variable-Frequency Source", IEEE Trans. on Energy Conversion, Vol. 7, No. 3, Sept. 1992, pp. 537-543.

[8] J. Kueck, J. Gray, R. Driver, J. Hsu, "Assessment of available Methods for evaluating in-service motor efficiency", Oak Ridge National Laboratory report, ORNL/TM-13237 (3-96).

[9] J. Hsu, J. Kueck, M. Olzewski, D. Casada, P. Otaduy, L. Tolbert, "Comparison of Induction Motor Field Efficiency Evaluation Methods", IEEE IAS Conf. 1996.

[10] A. K. Wallace, T. E. Rollman, "High Efficiency Testing Laboratory for Motors, Drives, & Generators", PEVD96 Conf. Proc., pp. 220-225.

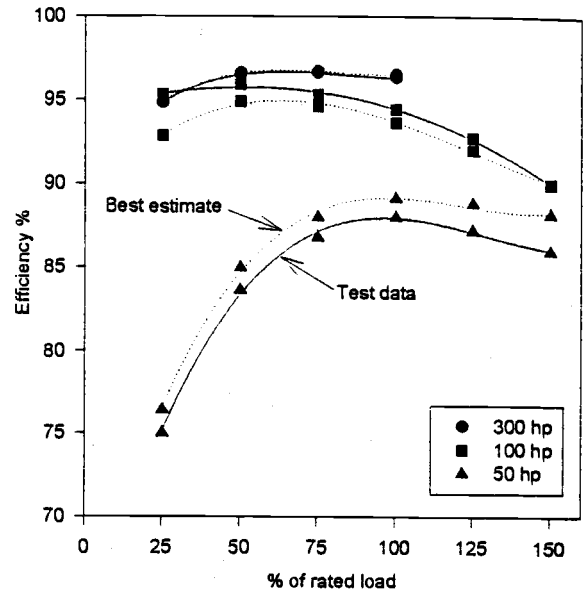


Fig. 2 Comparison of test data and best estimates for three motors at rated, balanced voltage.

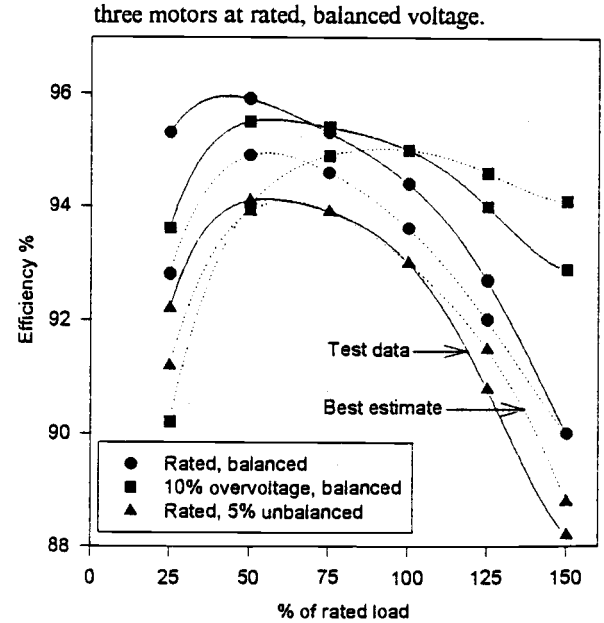


Fig. 3 Comparisons of test data and best estimates for 100 hp motor.

Appendix 1

Table 2. "Perfect" Test Motors

	Motor #1	Motor #2	Motor #3
Rating, hp	300	100	50
Speed, r/min	1780	1770	3550
Full-Load Current (A)	332	121.5	61.5
Line Volt. (V)	460	440	460
Nom. Eff. (%)	95.8	NA	NA
Nom. PF (%)	88.0	NA	NA

Appendix 2

Motor/Dynamometer Platform in MSRF Laboratory

Four Quadrant Dynamometer Converter

- Motor or Generator action
- Vector control for full load testing over entire speed range
- Programmable Torque and Speed Modes
    1. Steady state
    2. Torque vs. Speed profiles
    3. Torque vs. time profiles
    4. Speed vs. time profiles

Able to test devices up to 300 hp

- Motors
- Generators
- Converters and Controllers
- Instrumentation

Fully regenerative system

- Meets IEEE 519
- Dynamometer >95% efficient at rated load

Input power

- 750 KVA supply
- 0 to 600 VAC
- 3 phase, balanced or unbalanced

Mechanical specifications

- 300 hp
- 15,000 in-lb torque
- 0 to 4000 rpm
- Bi-directional rotation
- Full load testing over entire speed range

THE MEASURED EFFECTS OF UNDER-VOLTAGE, OVER-VOLTAGE AND UNBALANCED VOLTAGE ON THE EFFICIENCY AND POWER FACTOR OF INDUCTION MOTORS OVER WIDE RANGES OF LOAD

A K Wallace, A R von Jouanne, E J Wiedenbrüg, P S Andrews
(Oregon State University, USA)
C W Wohlgemuth (Bonneville Power Administration, USA)
J Douglass (Washington State University Cooperative Extension Energy Program, USA)
G Wainwright (Pacific Gas & Electric, USA)

This paper investigates the effects of over-voltage, under-voltage and unbalanced voltage on the performance of induction machines in terms of line currents, power factor and efficiency. A set of three motors of 50hp, 100hp and 300hp respectively, has been tested thoroughly at five different voltage conditions: rated voltage, 10% over-voltage, 10% under-voltage, 1.25% unbalance, and 2.5% unbalance. At each one of these voltage conditions, a load run series was performed for every motor. The data obtained shows how the different motors behave similarly, in terms of line current and power factor and differently in terms of efficiency, when subject to the particular voltage conditions.

INTRODUCTION

Increased energy costs and concerns over pollution and possible global warming are promoting efforts for increased industrial efficiency worldwide. In the USA the passing of the National Energy Policy Act of 1992, EPACT '92 (1), has prompted manufacturers of all forms of energy conversion equipment to investigate improved efficiency. As specifically mentioned in EPACT '92, the efficiencies of electric motors have been reviewed and improved where economically appropriate resulting in the introduction of the new "Design E" motors for October 1997. The nameplate efficiencies for newly manufactured induction machines which are being addressed in EPACT '92 are, of course, related to operation at full-load, with ideal terminal voltage conditions, a combination which almost never occurs in an industrial environment. Prior to EPACT '92 and Design E, the National Electric Manufacturers Association (NEMA) published guidelines for the effects of load on efficiency and power factor, NEMA (2), (Fig. 1) as well as guidelines for performance expectations when motors are subject to over-/under-voltage (Fig. 2) and recommendations for derating when subject to voltage unbalance (Fig. 3). These figures are now somewhat out of date and need to be revised and expanded to include Design E motors. NEMA (3).

In order to assess the efficiency and effectiveness of the manner in which induction motors are used in industry, the U.S. Department of Energy and the Bonneville Power Administration (BPA) instigated a three-phase study into the estimation of the efficiency of in-service induction motors in 1994. Phase one of the study was conducted by the Oak Ridge National Laboratory (ORNL), investigating the available methods of "in-service" efficiency estimation which appeared sufficiently unintrusive or undisruptive of industrial processes to be useful in practice. The results of the ORNL study are now available. Kueck et al (4), Hsu et al (5). Phase two of the BPA program called for evaluation of the estimation techniques which appear most promising for practical implementation and comparison of their results with strictly controlled laboratory tests. This work has been undertaken by Washington State Cooperative Extension Energy Program and the Motor Systems Resource Facility (MSRF) at Oregon State University. Phase three of the program will take those methods that appear most appropriate, following laboratory testing, and apply them in industry.

Three motors were tested as part of the MSRF "in-service" methods process: 50hp, 2-pole; 100hp, 4-pole; 300hp, 4-pole. Data was obtained over wide ranges of load, and several voltage conditions, intended to replicate the diversity of operating scenarios which might be experienced in practical industrial applications.

TEST PROGRAM

The Test Motors

Three different 460V rated cage rotor induction motors were successfully tested: 50hp, 2-pole; 100hp, 4-pole; 300hp, 4-pole. The 100hp machine had special connections enabling its windings to be deliberately unbalanced if desired, but only the balanced winding configuration will be discussed in the present paper. Details of the motors are

(i) 50hp; 2-pole; severe duty; totally enclosed fan cooled; design B; frame 326TS.
(ii) 100hp; 4-pole; open frame drip-proof; frame 444U.
(iii) 300hp; 4-pole; open-frame drip-proof; premium efficiency; design B; inverter duty; frame 447T.

The Laboratory Facilities

The MSRF test Laboratory has been described in detail in a previous paper, Wallace and Rollman (6), and is shown in the schematic of Fig. 4. The 225kW regenerative dynamometer system, Lewis et al (7), enables a circulation of energy between the motor under test and the dynamometer, such that only the system losses need to be provided from the utility supply. Motor output power is determined by the use of the most appropriate of three, mechanically interchangeable, torque/speed transducers rated upto 20,000 r/min and 2,000 lb·in (226 N·m), 5,000 lb·in (565 N·m), and 10,000 lb·in (1,130 N·m) respectively.
The motors under test are mounted on a custom built platform designed for ease of accommodation of machines of a wide range of aspect ratios (6), Wallace et al (8). Power is supplied to the motors under test via a motor starter and the most appropriately rated of a range of breakers. The voltages applied to the motors under test are controlled by an autotransformer, rated at 600A, with phase-to-neutral voltages independently controllable between 0 and 350V.

The Test Conditions

The adjustability of the dynamometer system and the controllability of the autotransformer are essential for the establishment and maintenance of the range of test conditions as follows.

1) Supply variables: balanced rated voltage (460V); balanced 10% over-voltage (506V); balanced 10% under-voltage (414V); 1.25% unbalanced voltage (460V); 2.5% unbalanced voltage (460V).
2) Load conditions: no-load (decoupled); 25%; 50%; 75%; 100%; 125% (50hp and 100hp only due to dynamometer limitations); 150% (50hp and 100hp only) of name-plate rating.

These conditions were intended to simulate the ranges that industrial motors may experience. In practice, plant managers tend to operate induction machines at over rated voltage (say 480 to 500V for a 460V motor) (a conservative measure to avoid under-voltage operation) and at a load estimated between 60% and 80% of nameplate rating. Supplies are invariable somewhat unbalanced; 2.5% is not uncommon and is frequently experienced in the Utility Supply to the MSRF Laboratory. The test data showed that 1.25% voltage unbalance only marginally affects induction motor performance and consequently the corresponding data is not included in the next section for ease of presentation.

THE TEST DATA

The Effects of Over- and Under-Voltage

For balanced rated voltage, the 50hp, 100hp and 300hp motors draw, respectively 60A, 119A and 330A. Using these values to normalize (to 100%) the currents drawn at other conditions results in the characteristics of Fig. 5. This shows, as would be expected, that lower currents are drawn by the motors when under-voltage conditions apply at low loads and over-voltage conditions apply at high loads. It is significant for the users of induction motors in industry to note that the cross-over of the under- and over-voltage and rated voltage characteristics occurs in the 50% to 60% rated load region. The similarity of these normalized characteristics is very marked except at substantial overloads.
Similarly, the power factors of the 50hp, 100hp and 300hp motors are, respectively 0.88, 0.88, and 0.89 at rated load and terminal voltage conditions. Using these to normalize the data for the other operating conditions results in Fig. 6. Again the similarity of the performance of the three motors is evident. As expected, improved power factor is obtained by use of under-voltage at low loads and over-voltage at high loads. Cross-over of these characteristics of the voltage conditions, however, occurs closer to 100% of rated load.
The efficiency curves of the three motors tested exhibit very different forms as shown in Figs. 7, 8 and 9. The value of low voltage at low load and high voltage at high load is again evident for all three motors. The cross-over condition occurs at 75% load for the 50hp motor, between 50% and 60% load for the 100hp motor, and approximately 40% for the 300hp motor. The load at which the efficiencies peak decreases with increasing motor size.

The Effects of Voltage Unbalance

Unbalance is defined according to NEMA (9) as

$$U = \frac{A_{max} - A_{mean}}{A_{mean}}$$

where A is either voltage or current, A_{mean} is the mean of the three line quantities and A_{max} is the line quantity which gives maximum deviation from the mean.
All induction motors have some inherent phase unbalance due, primarily, to mass-production tolerances. For the three motors examined in this study it was shown that the inherent motor unbalance produces current unbalance of 5% to 8% at low loads, reducing to 2% to 3% at high loads, with balanced applied voltage. The current unbalance increases rapidly with applied voltage unbalance such that, for a 2.5% voltage unbalance, the current unbalances have

grown to the range of 30% to 35% at low loads reducing to a 12% to 18% range at high loads. Surprisingly, in spite of these current unbalances, the efficiencies of the 300hp and 50hp motors are virtually unaffected from the balanced case. The 100hp motor was observed to suffer efficiency reduction resulting from the voltage (and current) unbalance, as shown in Fig. 10. At the rated load the motor losses increase from 5% to 7% of output necessitating a derating to about 80% load to maintain the same losses as balanced operation.

CONCLUSIONS

An extensive series of tests of three different induction machines was performed at the MSRF, running at five different voltage conditions and five (or seven, depending on the investigated machine) different load points per motor. Care has been taken to choose machines of different rating, duty factor, pole-number and design. The large quantity of recorded measurements makes it possible to isolate similar behaviors for each one of the three differently rated machines. It has been shown, after appropriate per-unitization, that the three investigated machines display a very similar performance characteristic in terms of power factor and line current as a function of the load and rated-, over- and under-voltage conditions. A result of the performed testing and evaluation is that the proposed NEMA derating recommendations for motors subject to over-/under-voltage do not coincide with the obtained results. The evaluation of the unbalanced cases, particularly regarding thermal I^2R losses in the stator, shows that the derating curve proposed by NEMA for unbalanced voltages agrees with the data obtained in the test series for the worst case. For the other two machines, the derating curve was conservative, since the percentage of unbalance of the currents changes strongly for every machine.
A further point of interest is that although the motors display very similar behaviors in terms of line current and power factor as a function of the voltage condition and the applied load, they do differ substantially from each other in the shape of their efficiency as a function of the load.

ACKNOWLEDGMENTS

Financial support from the Bonneville Power Administration and Pacific Gas & Electric for the development of this work is acknowledged, as is the support from the Electric Power Research Institute in the development of the test facilities.

REFERENCES

1. United States House of Representatives. 1992 "Energy Policy Act of 1992", report 102-108.

2. National Electrical Manufacturers Association. "MG1-2.59", NEMA Standards, Table 12-10.

3. NEMA, "MG1-12.60" Standards (Table 12-11), ibid.

4. J. Kueck, J. Gray, R. Driver, J. Hsu, 1996, "Assessment of available Methods for evaluating in-service motor efficiency", Oak Ridge National Laboratory report, ORNL/TM-13237 (3-96).

5. J. Hsu, J. Kueck, M. Olzewski, D. Casada, P. Otaduy, L. Tolbert, 1995, "Comparison of Induction Motor Field Efficiency Evaluation Methods", IEEE IAS Conf. 1995.

6. A. K. Wallace, T. E. Rollman, 1996 "High Efficiency Testing Laboratory for Motors, Drives, & Generators", PEVD'96 Conf. Proc., pp. 220-225.

7. T. Lewis, D. Heberle, A. Wallace, 1995, "Development of a Fully Regenerative 300hp Motor and Drive Test Facility", IEEE KPH Stockholm Power Technology Conf. Sweden.

8. A. Wallace, A. von Jouanne, E. Wiedenbrug, J. Douglass, C. Wolgemuth, G. Wainwright, 1997, "A Laboratory Assessment of In Service Motor Efficiency Testing Methods", IEEE IEMDC '97 Milwakee Wisconsin,.

9. NEMA Standards "MG1-14.35" (Fig. 14.1).

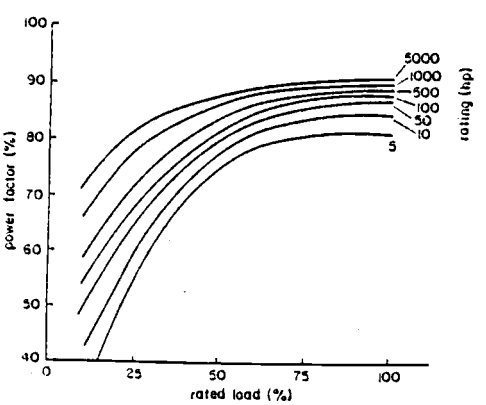


Figure 1-a: Typical Power factors versus load (NEMA)

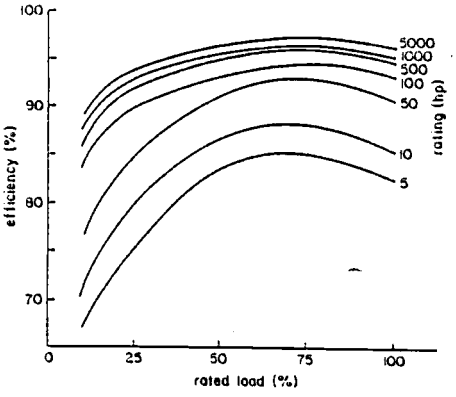


Figure 1-b: Typical Efficiency versus load (NEMA)

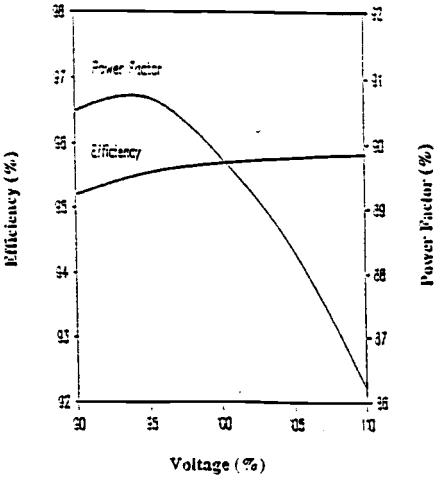


Figure 2: Efficiency and Power Factor versus over-/ under voltage conditions (NEMA)

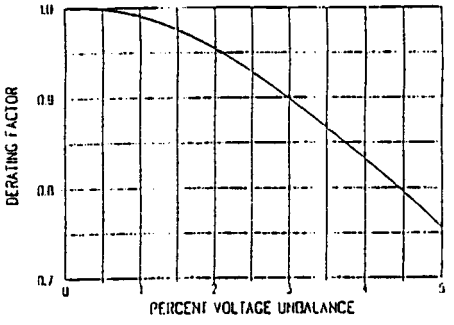


Figure 3: NEMA proposed derating against voltage unbalance in percent

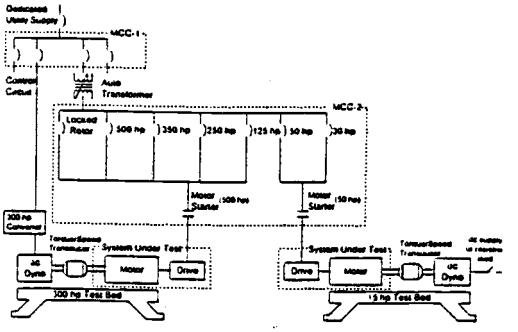


Figure 4: MSRF test center schematic

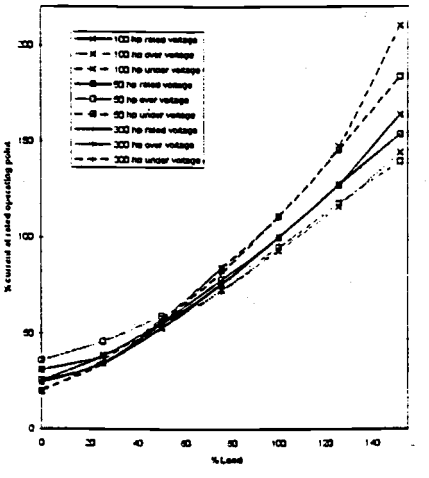


Figure 5: Line current in percent versus load

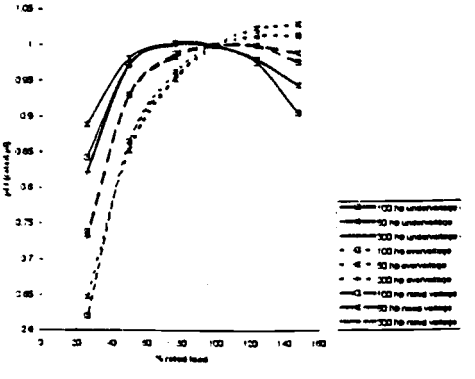


Figure 6: Per-unitized power factor versus load

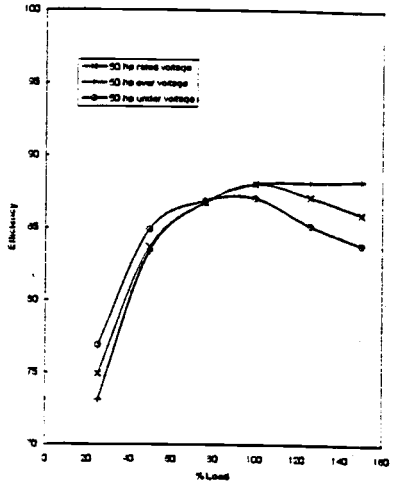


Figure 7: Efficiency versus load for the 50hp motor

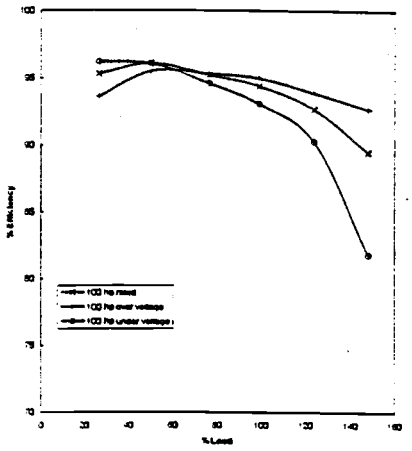


Figure 8: Efficiency versus load for the 100hp motor

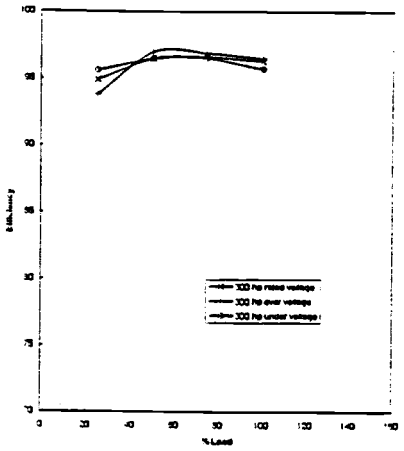


Figure 9: Efficiency versus load for the 300hp motor

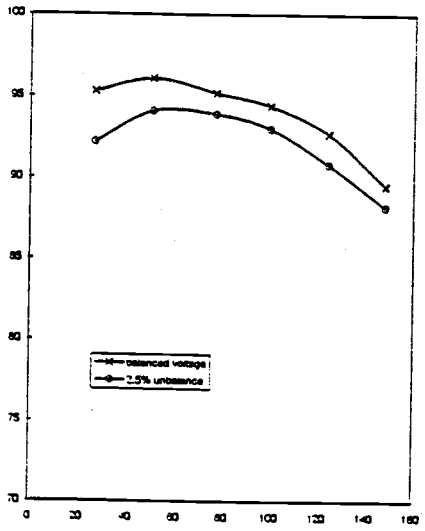


Figure 10: Efficiencies for balanced and 2.5% unbalanced voltage condition

Appendix E:

E.1 DFLL Matlab script:

```
tend    = 1;
f1      = 30.123;
fstart  = 28;
fend    = 32;
fstep   = 0.0003;
B       = 5;            %my second favorite number
a = [0:tend/5000:tend];    %generates the time vector
f = B * sin(2*pi*f1.*a);    %simulates the data acquired signal
w = .5*((1-cos(2*pi.*a/max(size(a)))).*(1-cos(2*pi.*a/max(size(a)))));
cnt = 0;
for freq = fstart: fstep: fend %frequency steps to be calculated
  cnt = cnt + 1;            %counter
  s = sin(2*pi.*a*freq).*w;  %generates compare sin signal
  c = cos(2*pi.*a*freq).*w;  %generates compare cos signal
  cres = f * c';          %inner product of acquired and generated cos
  sres = f * s';          %inner product of acquired and generated sin
  result(cnt) = sqrt(cres*cres + sres*sres);
end
f = [fstart:fstep:fend];
plot(f,result)
[x,y] = max(result);
frequency = fstart+y*fstep;
```