University of Cincinnati

Date: 7/12/2016

I, Achyut Vemuri, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:

Estimation of Instantaneous Speed for Rotating Systems: New Processing Techniques

Student's name: Achyut Vemuri

This work and its defense approved by:

Committee chair: Randall Allemang, Ph.D.

Committee member: David L. Brown, Ph.D.

Committee member: Allyn Phillips, Ph.D.

22200

Estimation of Instantaneous Speed for Rotating Systems: New Processing Techniques

A thesis submitted to the Graduate School of the
University of Cincinnati
in partial fulfilment of requirements for the degree of
Master of Science
in the Department of Mechanical and Materials Engineering
of the College of Engineering & Applied Sciences
July 2016

By
Achyut Vemuri
Bachelor of Technology, Jawaharlal Nehru Technological University,
Hyderabad, India
May 2013

Committee Chair: Dr. Randall J. Allemang
Committee Members: Dr. David L. Brown, Dr. Allyn W. Phillips

ABSTRACT

One of the most important parameters in the analysis of rotating systems is the instantaneous speed or RPM. Dedicated tachometers like photo probes, magneto-resistive, magnetic and Hall effect sensors are typically used along with post processing for RPM estimation at any arbitrary time. The current work contains an extension of the well documented process for a square wave output as applied to sinusoidal pulse train signals obtained from variable reluctance tachometers.

The estimates from tachometer data generally contain rapid unacceptable fluctuations. Post processing is done by fitting a spline to the initial speed estimate to eliminate any fluctuations that are not possible due to the inertia of the system. As such, it is not a solution to the problem but just a workaround. A new technique called the **Adaptive n^{th} Pulse Algorithm (AnPA)** has been developed as part of this thesis which aims to improve the estimation procedure by addressing the fluctuations in the data through a root cause analysis. A process by which the signal to noise ratio of the curve is greatly improved resulting in a profile that is more conducive to spline fitting than the traditional raw estimate method (no pulse skipping) is described. An averaging technique for a smooth instantaneous speed profile without the need for a spline fit is also proposed. In addition to significantly improving the noise characteristics, this method has the advantage of transforming the rotation speed from being a function of angle of rotation to a function of time.

The hardware (tachometer) based approach to instantaneous speed estimation presents several disadvantages such as cost, space constraints and unsuitable operating conditions in addition to the post processing required. An alternative approach has been developed by Brüel & Kjær (B&K) called the **Autotracker Algorithm**. The instantaneous speed is estimated using Bayesian statistics from measured accelerometer or microphone data, thereby avoiding the

need for tachometers. In the current work, a **Smart Bayesian Algorithm (SBA)** built upon this approach which effectively extracts the instantaneous RPM by tracking the dominant harmonics (orders) in the data has been developed. Analysis of data with crossovers between orders and dominant natural frequencies is also possible using a reprocessing step in this algorithm. Data with very high slew rate and speeds reaching 25000 RPM has been shown to be processed in under 5 seconds. Some prior knowledge of the system and its operating conditions is required to employ this method.

For both the algorithms, a detailed discussion of the procedure and the parameters involved is presented with applications to data sets from different sources. Examples include data from a rotor rig, a 6 cylinder engine powertrain and a vacuum cleaner. Comparison is made between the *Smart Bayesian Algorithm* (SBA) and *Adaptive n^{th} Pulse Algorithm* (AnPA) estimates for these data sets to show that the results from these two methods are consistent and support each other.

TABLE OF CONTENTS

Abstract	ii
Table of Figures	viii
Nomenclature	xiv
1. Introduction	1
1.1 Conventional Method for Instantaneous RPM Estimation	1
1.2 New Method – Adaptive <i>nth</i> Pulse Algorithm	2
1.3 An Alternative Approach to Tachometers	2
2. Theoretical Background	4
2.1 Bayesian Inference	4
2.2 Spectral Maps	10
2.3 Different Types of Tachometers	13
3. RPM Estimation from Variable Reluctance Tachometer Signals	15
3.1 Setup for Data Acquisition	15
3.2 Characteristics of a Variable Reluctance Tachometer Signal	16
3.3 Algorithm to Extract the Instantaneous Speed Curve	17
3.4 Algorithm Applied to Data from a Magneto-resistive Tachometer	22
4. The Smart Bayesian Algorithm (SBA)	24
4.1 Problems Faced During the Application of the Bayesian Formulation	24

4.2 Iterative Improvements to the Procedure	27
4.3 Procedure for the Improved Bayesian Algorithm	33
4.4 Model with Two or More Dominant Frequencies	36
4.5 Smart Algorithm	37
4.6 Comparison to Spline Fit Estimate:	39
5. The <i>nth</i> Pulse Algorithm (nPA)	41
5.1 Need for a Better Instantaneous Speed Estimation Process	41
5.2 Analysis of the Cause for Fluctuations in the Instantaneous Speed Curve	43
5.3 Concept of the <i>nth</i> Pulse Algorithm	45
5.4 Error analysis	47
5.5 Reasons for High Fluctuations	49
5.6 Combined <i>nth</i> Pulse Curve	50
5.7 Averaging Technique	58
6. The Adaptive <i>nth</i> Pulse Algorithm (AnPA)	62
6.1 Need for an Adaptive Algorithm	62
6.2 Concept of the Adaptive Algorithm	63
6.3 Development of the Algorithm	64
7. Comparison	69
8. Summary	83
8.1 Conclusion	83

8.2 Recommendations for future work	84
References	86
Appendix A	88
Applying FFT Principles to Non – Stationary Data:	88
Demonstration of the Smart Bayesian Algorithm - Reprocessing	89
Appendix B	91
MATLAB Script	91

TABLE OF FIGURES

Figure 1: Effect of Variance and Mean on the Normal Distribution

Figure 2: Demonstration of Difference in Weightage for Surrounding Points with Different Variance Values

Figure 3: Time Spectral Map – 6 Cylinder Engine Data

Figure 4: Top View of Time Spectral Map – 6 Cylinder Engine Data

Figure 5: RPM Spectral Map – 6 Cylinder Engine Data

Figure 6: Square Wave Output from an Infrared Tachometer

Figure 7: Sinusoidal Pulse Output from a Variable Reluctance Tachometer

Figure 8: Setup for Data Acquisition with Different Tachometers

Figure 9: Spacing between VR Tach and Ferromagnetic Feature

Figure 10: Amplitude Variation in Signal Due to Variation in Rotational Speed

Figure 11: Actual and Normalized Signals from a Variable Reluctance Tachometer

Figure 12: Flow Chart for RPM Estimation from Sinusoidal Pulse Train Signal

Figure 13: (a) Contamination Due to Noise (b) Contamination Due to Oscillations (c) Uncertainty in Amplitudes of the Peaks

Figure 14: Instantaneous RPM Estimates from Tachometers (a) VR1 (b) VR2

Figure 15: Comparison to Estimates from Infrared Tachometer (a) VR1 (b) VR2

Figure 16: (a) Magneto-resistive Tachometer Signal (b) Single Pulse

Figure 17: RPM Estimation for Magneto-resistive Tachometer Data Using the Algorithm Developed for Variable Reluctance Tachometers

Figure 18: Flow Chart for the Improved Bayesian Algorithm

Figure 19: Time Spectral Map for a Test Run from a Vacuum Cleaner

Figure 20: Initial (Raw) RPM Processed from Tachometer Data

Figure 21: RPM Profile Obtained with Uniform Prior

Figure 22: Time Spectral Map Demonstrating the Dominant Frequencies

Figure 23: Dominant Natural Frequency Filtered Out

Figure 24: Bayesian Estimates from (a) Filtered Data (b) Original Data

Figure 25: Bayesian Estimate with Prior Function Applied

Figure 26: Bayesian Estimate with Mid-Block as Initial Block, Prior Applied

Figure 27: Bayesian Estimate Overlapped on Time Spectral Map

Figure 28: Effect of Variance on Bayesian Estimate (a) $\sigma^2 = 1$ (b) $\sigma^2 = 0.15$

Figure 29: Top – Time Spectral Map and Initial Bayesian RPM Estimate; Bottom – Initial Estimate Overlapped on the Time Spectral Map

Figure 30: User Input for Reprocessing

Figure 31: Reprocessed Curve after User Input

Figure 32: Need for Curve Fit at the Crossing with Dominant Natural Frequency

Figure 33: Spline Fit to Interpolate the Estimate over a Small Region

Figure 34: Bayesian Estimate with 2 Dominant Harmonics Model

Figure 35: SBA Estimate vs Spline Fit Estimate

Figure 36: Comparison of SBA Estimate vs Assigned Tach Estimate (a) Absolute Difference
(b) Percentage Difference

Figure 37: Initial RPM Estimate Using Conventional Formulation

Figure 38: Initial Raw Estimate vs (a) Spline Fit (b) SBA Estimates

Figure 39: Example: Fluctuations in the Initial Estimate at ~ 25000 RPM

Figure 40: Range of Fluctuations in Speed Curve at ~ 25000 RPM

Figure 41: Instantaneous Speed Estimate with 18 Pulses Skipped

Figure 42: Reduction in Fluctuations at ~ 25000 RPM – 18 Pulses Skipped

Figure 43: Range of Fluctuations in the Speed Curve – 18 Pulses Skipped

Figure 44: Initial Raw Speed Curve – 6 Cylinder Engine Powertrain

Figure 45: Range of Fluctuations at ~ 1900 RPM – 6 Cylinder Engine Powertrain

Figure 46: Raw Instantaneous RPM – No Pulses Skipped

Figure 47: Starting at Pulse 1 – 40 Pulses Skipped

Figure 48: Starting at Pulse 2 – 40 Pulses Skipped

Figure 49: Starting at Pulse 3 – 40 Pulses Skipped

Figure 50: Starting at Pulse 4 – 40 Pulses Skipped

Figure 51: (Clockwise from above) Instantaneous Speed Estimate with the ***nth*** Pulse Process
Starting from 1st, 21st and 40th pulses

Figure 52: From Top to Bottom: Combination of Assigned Speeds at Different Time Instances;
Combined ***nth*** Pulse Curve with 40 Pulses Skipped; Zoomed in Version Showing Reduction
in Fluctuations

Figure 53: Process Starting at Different Pulses – Same No. of Pulses Skipped

Figure 54: Interpolated Estimates – Process Starting at Different Pulses, Brought to the Same Time Reference

Figure 55: Averaged *nth* Pulse Estimate – 40 Pulses Skipped

Figure 56: Differences in Time Resolution with Constant No. of Pulses Skipped

Figure 57: Consistent Time Resolution with AnPA

Figure 58: AnPA - Pulses Skipped vs Time

Figure 59: Duty Cycle

Figure 60: AnPA Estimate vs. Raw Estimate

Figure 61: Vacuum cleaner – Comparison SBA Estimate vs AnPA Estimate

Figure 62: Vacuum cleaner – Comparison SBA Estimate vs AnPA Estimate (a) Absolute Difference (b) Percentage Difference

Figure 63: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution at Various Locations in the Estimate

Figure 64: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

Figure 65: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

Figure 66: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped on the Time Spectral Map

Figure 67: 6 Cylinder Engine Powertrain – Comparison SBA Estimate vs AnPA Estimate

Figure 68: 6 Cylinder Engine Powertrain – Comparison SBA Estimate vs AnPA Estimate (a) Absolute Difference (b) Percentage Difference

Figure 69: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution at Various Locations in the Estimate

Figure 70: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

Figure 71: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

Figure 72: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped on the Time Spectral Map

Figure 73: SM Lab Rig Data Set 1 – Comparison SBA Estimate vs AnPA Estimate

Figure 74: SM Lab Rig Data Set 1 – Comparison SBA Estimate vs AnPA Estimate (a) Absolute Difference (b) Percentage Difference

Figure 75: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution at Various Locations in the Estimate

Figure 76: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

Figure 77: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

Figure 78: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped over the Time Spectral Map

Figure 79: SM Lab Rig Data Set 2 – Comparison SBA Estimate vs AnPA Estimate

Figure 80: SM Lab Rig 2 – Comparison SBA Estimate vs AnPA Estimate (a) Absolute Difference (b) Percentage Difference

Figure 81: (a) Stepped and (b) Smooth Bayesian Estimates

Figure 82: Microphone Data – (a) Final and (b) Initial SBA Estimates over the Time Spectral Map

Figure 83: Microphone Data – (a) Final and (b) Initial SBA Estimates

NOMENCLATURE

Symbols

Description

$\{\theta\}$

Set of parameters to be estimated

ω

Frequency in rad/second

D

Acquired data

σ^2

Variance

M_i, I

Model chosen

e_n

Error in estimation of the n^{th} data point

t

Time recording or sample

B_{2n-1}, B_{2n}

Coefficients corresponding to frequency ω_n

d_i

The i^{th} data point

N, BS

Data block size

ω^l

Frequency estimate of the l^{th} data block

T

Total time sampled

Δf

Frequency resolution

Δt

Discrete sampling time

n

No. of pulses skipped

F_{samp}

Sampling frequency

F_{nyq}

Nyquist frequency

FFT

Fast Fourier Transform

<i>FRF</i>	Frequency response function
<i>DSP</i>	Digital signal processing
<i>RPM</i>	Rotations per minute
<i>SBA</i>	Smart Bayesian Algorithm
<i>ANPA</i>	Adaptive n^{th} Pulse Algorithm

1. INTRODUCTION

1.1 Conventional Method for Instantaneous RPM Estimation

Instantaneous RPM or speed estimation was traditionally performed using infrared, magnetic, magneto-resistive, Hall effect or other position sensors. The signal obtained is a waveform of varying frequency (and amplitude sometimes) depending on the rotating speed. This signal needs to be processed, typically in the time domain, in order to extract the speed at any arbitrary time. The processing involves capturing zero crossings in the waveform, obtaining an initial estimate and then fitting a curve (spline) to obtain a more physically compatible speed profile [1]. While this process has been well documented for a square waveform, the application to sinusoidal pulses (output from variable reluctance tachometers) is yet to be investigated. In Chapter 3, an algorithm which can extend the application to a sinusoidal pulse train will be explained along with examples.

Shortcomings of the Conventional Method:

Curve fitting might generate a speed profile consistent with the inertia of the system but it is a mathematical approximation relying heavily on the processing parameters used and not always governed by the physics of the system. The inertia control is applied through brute force by shaving the data, a term used in the processing of rotational speed data to mean that the outlier data points have been removed. Depending on the data set, it might be very difficult to attain a curve fit which stays within bounds of the fluctuating initial estimate.

1.2 New Method – Adaptive n^{th} Pulse Algorithm

The *Adaptive n^{th} Pulse Algorithm* (AnPA) was developed as an attempt to minimize the fluctuations in the estimate while using the same conventional formulation. On analysing the formulation, it is clear that the fluctuations in speed stem from small but significant errors in the interpolation for zero crossings.

In order to minimize the effect of these errors (zero crossing uncertainty), several pulses can be skipped so as to maintain a larger time interval between samples. The underlying assumption is that there is very little variation in the true speed of the system over the duration of the pulses skipped. The error analysis and subsequent processes that have been developed are given in Chapters 5, 6 and 7. The results clearly show an improvement in the signal to noise ratio.

1.3 An Alternative Approach to Tachometers

The use of dedicated hardware in industrial applications presents many challenges. While variable reluctance tachometers are usually more temperature resistant and physically robust than photo probes, the sensors have to be mounted and held steadily during the operating period. This could prove to be challenging sometimes considering the operating conditions. Moreover, they require a dedicated channel in the data acquisition system (or sometimes a specific tachometer channel) along with other sensors like accelerometers for order analysis. Using tachometers with such acquisition systems could be impractical and expensive in the oil industry for example, where the facilities might have hundreds of operating pumps.

It would therefore be very beneficial if the need for a tachometer can be avoided. It is well known that the orders or harmonics of a rotating system, which are caused due to geometry and/ or imperfections in the systems are responses related to the rotating speed which is the

forcing function. By observing a time spectral map, it can be clearly seen that the path traced by the dominant harmonics through the data blocks is similar to the RPM profile. Hence with prior knowledge of dominant orders in the system and operating speed conditions, the instantaneous speed can be extracted from measured vibration/ acoustic data by tracking these orders.

Bayesian statistics provides an intuitive way of estimating parameters from spectral data based on the laws of probability. In the present context, the dominant frequencies in a given time series can be estimated using Bayesian inference. In the past, this method was not favoured due to computational intensity and complexity of the formulation. Bayesian inference has come back into prominence as the computational power increased over recent decades. The *Autotracker Algorithm* for instantaneous speed estimation developed by B&K is based on Bayesian statistics and works on the principle described above.

A *Smart Bayesian Algorithm* (SBA) for RPM estimation has been illustrated in Chapter 4 which is the result of an effort to improve the computational efficiency and robustness of the *Autotracker Algorithm* [1]. Chapter 7 contains comparisons between the results from SBA and AnPA for different data sets.

2. THEORETICAL BACKGROUND

2.1 Bayesian Inference

Bayesian statistical inference is based on Baye's rule in probability theory which states that for two events ' A ' and ' B ',

$$P(A|B) * P(B) = P(B|A) * P(A) \quad (1)$$

The law follows intuitively from the definition of conditional probability and can be very helpful in updating prior belief in the distribution of a random variable based on newly obtained data. As applied to data blocks and parameter estimation, the Baye's rule is given by [3],

$$P(\theta|D, M_i) = \frac{P(D|\theta, M_i) * P(\theta| M_i)}{P(D|M_i)} \quad (2)$$

Note that ' M_i ' is the model chosen, ' D ' is the data obtained and $\{\theta\}$ are collectively the parameters to be estimated. Any prior knowledge about the system can be included in the model as appropriate (e.g. Dominant harmonics, rate of change of speed etc.) [1].

The expressions involved in the formulation are:

- Posterior distribution $P(\theta|D, M_i)$
- Likelihood function $P(D|\theta, M_i)$
- Prior distribution $P(\theta|M_i)$
- Marginalizing factor $P(D|M_i)$

Thus, referring to Equation (2), the posterior probability of a parameter is obtained from the product of a prior probability distribution and the likelihood of obtained data based on the

assumed model divided by the marginalization factor. The objective of the analysis would be to determine the set of parameters which would maximize the posterior probability across the total parameter distribution range. Calculation of the marginalizing factor can be very complex but fortunately, it is a constant for a given data set and hence can be safely ignored for the purpose of identifying the maximum posterior probability.

Likelihood function comprises the formulation of the chosen model for the problem. If ' M_i ' is the model that is chosen to define the system, then the likelihood of obtaining a given data with the assumed parameters will be the same as the probability of obtaining an error ' e_i ' which is the difference between obtained data and the estimate from chosen model definition ' M_i ' with assumed parameters $\{\theta\}$.

For a single dominant frequency model [3], ' M_i ' or ' I ' can be defined by,

$$I \Rightarrow f(t) = B_1 \cos(\omega t) + B_2 \sin(\omega t) \quad (3)$$

Then,

$$P(e_1, e_2, \dots, e_N | \sigma, I) \propto \prod_{i=1}^N \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) \right] \quad (4)$$

If B_1, B_2, ω and σ collectively comprise $\{\theta\}$, then,

$$\begin{aligned} P(D | B_1, B_2, \omega, \sigma, I) &\propto L(B_1, B_2, \omega, \sigma) = \prod_{i=1}^N \sigma^{-1} \exp\left\{-\frac{1}{2\sigma^2} [d_i - f(t_i)]^2\right\} \\ &\Rightarrow L(B_1, B_2, \omega, \sigma) = \sigma^{-N} \times \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^N [d_i - f(t_i)]^2\right\} \end{aligned} \quad (5)$$

$$P(\omega|D, \sigma, I) \propto \exp \left\{ \frac{C(\omega)}{\sigma^2} \right\} \quad (6)$$

Note that $C(\omega) = \frac{1}{N} * [R(\omega)^2 + I(\omega)^2]$ is the Schuster Periodogram [4] and ' σ ' is the noise variance.

Marginalization is a very powerful tool in probability theory by virtue of which the complexity of the model can be greatly reduced as all the parameters in the model need not be known to estimate those of interest. Equation (6) was obtained from Equation (5) by marginalization of ' B_1 ' and ' B_2 ' which is done by integrating the probability density functions over these variables.

If noise variance is unknown, ' σ ' too can be marginalized. Then,

$$P(\omega|D, I) \propto \left[1 - \left(\frac{2C(\omega)}{N\overline{d^2}} \right) \right]^{\frac{2-N}{2}} \quad (7)$$

Where,

$$\overline{d^2} = \frac{1}{N} \sum_{i=1}^N d_i^2 \quad (8)$$

$$R(\omega) = \sum_{i=1}^N d_i \cos(\omega t_i) \quad (9)$$

$$I(\omega) = \sum_{i=1}^N d_i \sin(\omega t_i) \quad (10)$$

This process is repeated for all frequencies ' ω ' from 0 to F_{Nyq} . The set $\{\theta\}$, in this case the single frequency ' ω ' which gives the maximum $P(\omega|D, I)$ is taken as the dominant frequency

for the data block ' D '. In the present context, this frequency can be divided by the dominant harmonic number to give the rotation speed of the system. A smart algorithm will be discussed later that significantly reduces the amount of data being used for computation by taking advantage of the inertia of the system.

As an extension, for multiple dominant frequencies, the model [4] can be described as,

$$d_m(t) = a_{m,0} + \left[\sum_{k=1}^K a_{m,k} \cos(\omega_k t) + b_{m,k} \sin(\omega_k t) \right] + e_m(t) \quad (11)$$

In matrix form,

$$D = G * b + e \quad (12)$$

Where,

$D = [d(t_0), \dots, d(t_{N-1})]$ – $N \times 1$ vector of data points

$G = [1, \cos(t.\Omega^T), \sin(t.\Omega^T)]$ – $N \times 2K$ matrix whose columns are the basis vectors
evaluated at each point in the time series

$T = [t_0, \dots, t_{N-1}]$ – Vector containing time samples

$\Omega = [\omega_1, \dots, \omega_K]$ – The dominant harmonics

The likelihood function for this model can be given by,

$$P(d_m | \omega_0) \propto \frac{(d^T d - d^T G (G^T G)^{-1} G^T d)^{-\frac{(N-2K-1)}{2}}}{\sqrt{G^T G}} \quad (13)$$

Characteristics of the Prior Distribution:

- The *prior* is a modelling assumption about the nature of the parameters which generally takes the form of a probability distribution expressing prior belief in the system

- *Non informative priors*: No assumption is made to suggest that some values of a parameters are more probable than others. E.g. Flat prior (same probability from $-\infty$ to $+\infty$), Jeffrey's prior [6] etc.
- *Informative priors* can be used to include prior knowledge about the system
- Non informative priors can be used as a parallel analysis with informative prior to estimate the sensitivity of analysis to the prior

The prior distribution used for a given block of time data in the following work will be a normal distribution whose mean would be determined by the RPM estimates of the previous data blocks. Choosing this prior provides a smooth tracking of the speed as it weighs the frequency bins surrounding a previous estimate more than a frequency that is a few bins away. The function to be used and the number of previous estimates to be used depend upon the system being analysed. A linear function involving two previous block estimates has been discussed in [1]. A variation of this expression as given in Equation (14) has been used in the following analysis.

$$P(\omega_0^l | \omega_0^{l-1}, \omega_0^{l-2}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(\omega_0^l - 2\omega_0^{l-1} + \omega_0^{l-2})^2}{2\sigma^2} \quad (14)$$

To understand the effect of this prior, it is necessary to briefly review the concepts of a normal distribution. Figure 1 demonstrates the effect of the mean and variance on the shape of the normal distribution.

Figure 1: Effect of Variance and Mean on the Normal Distribution

The effect of variation in mean is to shift the position of the peak while that of variance is to increase or decrease the magnitude of the peak. When multiplied by a likelihood function, this prior distribution essentially weighs the values of the likelihood with the corresponding values of the normal distribution. Hence, it can be said that the parameter which affects the weighting of frequencies in the Bayesian analysis is the variance of the prior distribution.

This is demonstrated in Figure 2 with two distributions of variances 0.25 and 0.15. In the case of $\sigma^2 = 0.15$, the point which is 0.2 units away from the peak is weighed at around 41% of the peak value (59 % reduction). When $\sigma^2 = 0.25$, the point which is 0.2 units away is weighed at 72% (28% reduction) of the peak value. Thus, variance is to be chosen according to the operating conditions and the physical characteristics of the system. Generally, trial and error can be used to arrive at the appropriate variance for the analysis quite easily.

Figure 2: Demonstration of Difference in Weightage for Surrounding Points with Different Variance Values

Bayesian Estimation as a Special Case of Traditional Statistical Methods:

- Least squares and maximum likelihood approaches can be perceived as special cases of Bayesian estimation [3]
- If the error function is minimized, it is equivalent to the Least Squares method
- If a uniform prior is used along with the criterion of maximum posterior probability, it would be equivalent to the Maximum Likelihood function
- The informativeness of a prior can be assessed by how close the estimate is to a Maximum Likelihood estimate

2.2 Spectral Maps

For non-stationary data, it is not appropriate to transform the time data into the frequency domain using an FFT and then to average using various techniques. The information contained

in each data block is unique and has to be treated differently. Spectral maps are used to visually inspect the changes in the frequency content of the system as a function of time [7]. The entire time data is divided into blocks and the FFT of each block is plotted as a function of time (usually corresponding to the mid-point time of the data block). The resulting three-dimensional plot is given for a 6 cylinder engine data in Figure 3 and the top view is given in Figure 4.

Figure 3: Time Spectral Map – 6 Cylinder Engine Data

Figure 4: Top View of Time Spectral Map – 6 Cylinder Engine Data

If the speed of the rotating member during a given data block (usually mean speed) is assigned to the FFT, the plot of *Instantaneous Speed vs Frequency* called the RPM spectral map is obtained. This is shown in Figure 5.

Figure 5: RPM Spectral Map – 6 Cylinder Engine Data

Note: FFT assumes a stationary system (one whose frequency content does not change with time) and so one needs to be careful while making inferences from the frequency domain while analysing rotating systems. Refer to Appendix A for further explanation.

2.3 Different Types of Tachometers

Variable reluctance (magnetic), magneto-resistive, Hall effect, laser and infrared tachometers are usually employed for estimation of instantaneous speed of a rotating member. While the end results are similar, the difference is primarily in the physical principal of operation for each tachometer sensor. Photo probes like laser and infrared tachometers work on the principal of reflection of light from the rotating shaft. Most of the other tachometers work on the principal of magnetism (magnetic or inductive field disturbance) wherein the characteristic pulses are generated due to a change in magnetic flux. The output of a majority of the sensors are square waves while the variable reluctance tachometer generates a sinusoidal pulse train. Noise or other complicated characteristics (such as Gibb's oscillations) are typically present in the data.

There are advantages and limitations to each type of sensor. The photo sensors can be used at a distance from the rotating member and so mounting does not become an issue. However, they are expensive, not resistant to high temperatures and require clean operating conditions. Hence they are more suitable for laboratory purposes but not for application near the car wheels for example. The variable reluctance tachometers are inexpensive, robust and can be used in relatively harsh operating conditions. However, they can only be used with ferromagnetic material and are to be mounted very close to the rotating member (close to a millimetre) which becomes a major problem while mounting. Also the distance has to be maintained throughout the operation which becomes a challenge.

The square wave output and sinusoidal pulse output from infrared (IR) and variable reluctance (VR) tachometers respectively are given in Figure 6 and Figure 7.

Figure 6: Square Wave Output from an Infrared Tachometer

Figure 7: Sinusoidal Pulse Output from a Variable Reluctance Tachometer

3. RPM ESTIMATION FROM VARIABLE RELUCTANCE TACHOMETER SIGNALS

3.1 Setup for Data Acquisition

Extraction of instantaneous RPM from a square wave has been explained in detail in [1]. The objective of this section of the thesis is to introduce this technique and extend the application to signals from variable reluctance tachometers (sinusoidal pulses).

The set up used for data acquisition is shown in Figure 9. Sensors used: 2 VR (variable reluctance), 1 infrared, 1 laser and 1 magneto-resistive tachometer

Figure 8: Setup for Data Acquisition with Different Tachometers

Figure 9: Spacing between VR Tach and Ferromagnetic Feature

DAC Express TM, a trademark of VXI Technology, Inc. was used for data acquisition. The tachometers could not be mounted with prescribed proximity but were placed as close as possible to the rotating ferromagnetic feature as shown in Figure 9.

3.2 Characteristics of a Variable Reluctance Tachometer Signal

Unlike the output from photo or magneto-resistive tachometers, variable reluctance signal is characterised by varying amplitude as well as frequency. This could become a problem at high speeds as the amplitude might exceed the dynamic range of the acquisition system. The characteristics of a VR signal are shown in Figure 10 and Figure 11.

Figure 10: Amplitude Variation in Signal Due to Variation in Rotational Speed

Figure 11: Actual and Normalized Signals from a Variable Reluctance Tachometer

The signal is normalized by the maximum amplitude of the signal so that it is bounded between 1 and -1. This is done for convenience of choosing a standard amplitude range during user input to avoid recording peaks due to oscillations and noise. This will be discussed in detail when the algorithm is explained.

3.3 Algorithm to Extract the Instantaneous Speed Curve

The general outline of the algorithm is given below in Figure 12.

Figure 12: Flow Chart for RPM Estimation from Sinusoidal Pulse Train Signal

Zero Crossings:

Choosing a standard point corresponding to each pulse is necessary for processing the RPM from the tachometer signal. The formulation is given in Equation (15).

$$\text{Instantaneous RPM} = \frac{60}{t_1 - t_2} \quad (15)$$

Note that t_1 and t_2 are the corresponding points from two consecutive sinusoidal pulses. In theory, any arbitrary point can be chosen in a pulse as long as the same corresponding point is chosen for every pulse. However, it is best to choose the zero crossing when the pulse changes from positive to negative or negative to positive amplitude. Other points might not be good choices as they can be contaminated by noise and sensor characteristics like oscillations due to inertia. Also, the rotating speed might alter the number of points across a given pulse. This is illustrated in Figure 13.

Choosing the point where the pulse begins can be problematic as there is no clear initial point. Similarly the end point of the pulse is contaminated by noise and the oscillations till steady

state is achieved. Any other point in between cannot be arbitrarily chosen as the amplitude and frequency of the pulse (height and width of pulse) changes rapidly with speed. Therefore, the choices are limited to the peak, valley or the zero crossing. Due to the discrete nature of the signal, it is not appropriate to choose the peaks as there is no guarantee that the true peak is recorded during data acquisition.

**Figure 13: (a) Contamination Due to Noise (b) Contamination Due to Oscillations
(c) Uncertainty in Amplitudes of the Peaks**

It is therefore necessary to use the zero crossings for the processing. Again, due to the discrete nature of the signal, the zero crossing will most likely not be recorded accurately by the

acquisition system. From a strictly statistical point of view, the probability of recording a zero crossing exactly is zero. Hence, zero crossings are interpolated using Equation (16).

$$x_{crossing} = \left((0 - y_1) \times \frac{x_2 - x_1}{y_2 - y_1} \right) + x_1 \quad (16)$$

Note that (x_1, y_1) and (x_2, y_2) are the points before and after the zero crossing respectively that are used for interpolation and $x_{crossing}$ is the time instance at the crossing to be determined.

A method is needed to differentiate the zero crossing of interest from those that occur due to noise and other oscillations. This can be easily done by first finding the peaks in the signal and then finding the zero crossing just before or after the peak as appropriate. If the pulse has a peak – valley characteristic, the zero crossing of interest will be the one occurring after the peak and if it is a valley – peak characteristic, it would be the one before the peak. In order to accommodate for this possibility, the user is prompted to choose the characteristic of the signal.

Also, the oscillation amplitude at the end of a pulse might sometimes be significant enough to be detected as a peak. The solution for this is to utilize additional arguments in the function '*findpeaks(X, MINHEIGHT)*' in MATLAB which then sets a minimum amplitude for peaks to be recorded. It is solely for this purpose that the signal is normalized so that the user has a standard range and a convenient way of specifying the minimum amplitude.

Now, Equation (15) can be used for RPM estimation. Trial runs have been performed on the setup shown in Figure 9. The results from a coast down trial are shown in Figure 14 and Figure 15. Spline fitting has not been performed to show the characteristics of the initial RPM estimate.

(a)

(b)

Figure 14: Instantaneous RPM Estimates from Tachometers (a) VR1 (b) VR2

(a)

(b)

Figure 15: Comparison to Estimates from Infrared Tachometer (a) VR1 (b) VR2

It can be seen that there is less uncertainty and fewer oscillations in the initial instantaneous RPM estimate from the VR tachometers as compared to the IR tachometer. The singular glitch in VR 2 estimate at around 30 seconds is assumed to be due to noise.

3.4 Algorithm Applied to Data from a Magneto-resistive Tachometer

A magneto-resistive tachometer works on the principle of magnetism and involves an internal circuitry which finally gives a square waveform as the output signal. The oscillations to reach steady state are sustained longer and at higher amplitudes than for photo probes (See Figure 16). The algorithm that was developed for the sinusoidal waveform in a broad sense is just a method of finding the correct zero crossing for use in Equation (15). As the signal exhibits a peak – valley characteristic, the same algorithm as discussed has been applied out of curiosity although the traditional method for square waves is simpler. Results are shown in Figure 17.

Figure 16: (a) Magneto-resistive Tachometer Signal (b) Single Pulse

Figure 17: RPM Estimation for Magneto-resistive Tachometer Data Using the Algorithm Developed for Variable Reluctance Tachometers

4. THE SMART BAYESIAN ALGORITHM (SBA)

4.1 Problems Faced During the Application of the Bayesian Formulation

The following estimation procedure for instantaneous RPM has been developed in iterative improvements to overcome several problems discovered along the way. Straight forward application of the Bayesian formulation as explained in Chapter 2 might not be appropriate for all systems and operating conditions.

Some problems that might arise are:

- Orders not dominant at low speeds
- Contamination from dominant natural frequencies of the system
- Inaccurate estimation due to noise
- RPM profile has different parts with different rates of change of speed
- Computational intensity

For proper understanding of these issues and how they are addressed, the algorithm will be explained in steps as it has been developed. The general outline of the improved algorithm is given in Figure 18.

Figure 18: Flow Chart for the Improved Bayesian Algorithm

The data obtained is divided into blocks of N points each. To account for the non – stationarity of the system, the data records or blocks are overlapped for analysis. Block size N is also chosen such that there is an optimum combination of sufficient resolution ($< \Delta f$) and retention of stationarity within the time block. A single dominant frequency model will be used for analysis.

While computing for the posterior distribution in the Bayesian procedure, the frequency steps are incremented by a fraction of Δf or maybe even an increment of the user's choice (usually about 0.1 Hz) instead of Δf . For further details, refer to Appendix A. Also, the likelihood function given in Equation (7) involves the exponent $(1 - N/2)$ which might cause computational problems. This can be modified as required without loss of proportionality.

Windows are not employed for the present analysis. Windows are used in order to make the data more periodic in the time block by bringing the values at the beginning and ending of the data block closer to zero. This is not appropriate in this case as the initial and end values of a data block are important for accurate frequency estimation. For further details, refer to [2].

The time spectral map for the data from a vacuum cleaner is shown in Figure 19.

Figure 19: Time Spectral Map for a Test Run from a Vacuum Cleaner

This data has been chosen for demonstration as it presents several of the challenges that inspired the different capabilities included in the algorithm. It can be seen that one order is dominant at low frequency ranges and there is one fairly dominant order at very high frequency ranges. Initially, the use of a single dominant frequency model will be illustrated. A uniform prior is applied and Equations (3) – (10) are used for the estimation. The results are shown in Figure 21 and the initial tachometer estimate (prior to spline fit) is shown in Figure 20.

Figure 20: Initial (Raw) RPM Processed from Tachometer Data

Figure 21: RPM Profile Obtained with Uniform Prior

4.2 Iterative Improvements to the Procedure

The results suggest that the order of interest may not be the dominant frequency in all the time data blocks. This is correct, as can be observed from the time spectral map shown in Figure 22.

Figure 22: Time Spectral Map Demonstrating the Dominant Frequencies

To address this contamination, one approach considered was to filter out a band containing the natural frequency. However, this would also filter out the order (and hence the rotational speed) information which would then have to be interpolated in this band. The time spectral map with

this filter and the corresponding Bayesian estimates are shown in Figure 23 and Figure 24 respectively.

Figure 23: Dominant Natural Frequency Filtered Out

Figure 24: Bayesian Estimates from (a) Filtered Data (b) Original Data

Although the dominant frequency has been removed, some other frequency may become dominant in the corresponding data block and the process needs to be repeated. It may not be practical even if this approach can be improved further for the user to identify and filter out all such dominant features from the time spectral map.

Hence, it is important to use a *prior* which sufficiently weighs the frequencies around the estimates of the previous blocks to track the orders correctly. A similar form of the prior function given in Equation (14) involving two previous block estimates has been shown to work sufficiently well for this purpose in [1]. The first two blocks of data are processed with a uniform prior. Results after application of the prior are shown in Figure 25.

Figure 25: Bayesian Estimate with Prior Function Applied

The reason for such a bad estimate is due to the fact that the *prior* would weigh the surrounding frequencies of the previous block estimates for the estimation of the current block. If an incorrect frequency is estimated by the formulation in the first two blocks, the tracking algorithm then is set to this frequency. In this case, the dominant frequency in the data is around 120 Hz or 7200 RPM. Once the prior is set to this frequency which is highly dominant, it does not track any other frequency.

It is therefore important for the user to choose a block to initiate the process such that the order of interest is the most dominant frequency. The initiation block is one of the most important parameters in the algorithm. The processing will then be started from the initial block and proceed backwards to the first block and forwards to the end block.

The processing time and complexity can be significantly reduced by limiting the Bayesian analysis to the first dominant order of interest. Thus, the range of analysis will be $(\text{first dominant order number}) \times (\text{maximum RPM for the run})/60$. This will ensure that there is only one dominant harmonic in the analysis range which is consistent with the model chosen. The irregular spikes in the speed profile as can be seen from Figure 21 can be avoided by using this approach as it reduces contamination from higher frequencies. This also optimizes the analysis range thereby reducing the computation required.

Figure 26 shows the results with the following parameters: Initial block – 296th block, maximum frequency – 467 Hz (2.8×10^4 RPM), Overlap – 87.5%, Variance for the prior – 1.

Figure 26: Bayesian Estimate with Mid-Block as Initial Block, Prior Applied

The Bayesian estimate now starts to approach the true profile as processed from the tachometer. Comparison is made to the raw tachometer estimate for now to show that the profile is within the fluctuating curve obtained. Notice that the Y-axis describing the number of overlapping blocks has been converted to time for the comparison. This was done by assigning the Bayesian estimate to the central time point of the corresponding time data block.

It can be seen that the Bayesian RPM profile deviates on a tangent at certain blocks and does not recover due to the applied prior. This is quite clear when the Bayesian estimate (black curve) is projected upon the time spectral map as shown in Figure 27.

Figure 27: Bayesian Estimate Overlapped on Time Spectral Map

This could be due to one or more of the following reasons:

- More previous blocks required for analysis in the prior function
- More complex (higher order) prior functions required to track the orders
- Different variance required at different regions of the profile
- The tracker is deviated by the modulation effect at points where the order crosses a natural frequency of the system

Manipulating the prior function by involving more blocks or higher order functions would cost significant computation time. At this point, the estimation process takes around 104 seconds with a block size of 2048, 87.5% overlap and 5120 Hz sampling rate for 30 seconds of test data. It is therefore not efficient to complicate the function further.

The spread of a normal distribution is determined by the variance used. In the present context, the variance determines the weighting for the region around a particular estimate used for the next data block. The effect of varying the variance is given in Figure 28.

Figure 28: Effect of Variance on Bayesian Estimate (a) $\sigma^2 = 1$ (b) $\sigma^2 = 0.15$

It is quite clear that variance has a significant influence in tracking the dominant order. However, at around 7000 RPM (6 seconds), the curve is deviating again. Further change in variance does not improve the result. It can be seen from the spectral map that this point is close to a dominant natural frequency of the system. Hence it might be a case of contamination due to the modulation effect of the harmonic versus the natural frequency. To overcome this problem, the blocks before the system reaches 7000 RPM need to be analysed separately. Also, the frequency range of analysis must be less than $(7000/60)$ Hz to avoid problems with the algorithm tracking just the natural frequency.

The current algorithm enables the user to reprocess the data over a specified range with a different choice of parameters. The important features to note are the ability to reset the frequency range over which the analysis is done in a given data block and the option to

interpolate through a specified region in the RPM curve. Interpolation can be used as a last resort when no further amount of reprocessing can generate a proper estimate.

The first few data blocks in which the natural frequency is more dominant than the order can now be processed better without being affected by contamination. The reprocessing time required is very small when compared to the time taken for the entire curve processing and so does not add any significant analysis time. These features are demonstrated from Figure 29 to Figure 33. The step-wise application process is described below.

4.3 Procedure for the Improved Bayesian Algorithm

- Observe the spectral map and set up the first estimate

Figure 29: Top – Time Spectral Map and Initial Bayesian RPM Estimate; Bottom – Initial Estimate Overlapped on the Time Spectral Map

- Reprocess data with a different combination of parameters

Figure 30: User Input for Reprocessing

Figure 31: Reprocessed Curve after User Input

- At the block where the order crosses the dominant natural frequency, contamination cannot be avoided no matter what parameter choice has been made. An interpolation will be required in such cases. The need for a curve fit is demonstrated in Figure 32.

Figure 32: Need for Curve Fit at the Crossing with Dominant Natural Frequency

- If the dominant harmonic cannot be recovered over a small range, interpolate the curve

Figure 33: Spline Fit to Interpolate the Estimate over a Small Region

4.4 Model with Two or More Dominant Frequencies

Results from application of the conventional Bayesian formulation for a two dominant harmonics model are shown in Figure 34. There is no notable improvement in the estimation from that of a single frequency model.

Figure 34: Bayesian Estimate with 2 Dominant Harmonics Model

Further investigation using two or higher dominant frequencies models has not been done due to the following reasons:

- Care must be taken in choosing the dominant harmonics or else the estimation might not be accurate
- Computationally intense – Model with two dominant frequencies takes 5 – 10 minutes to process the same data which was done in 104 seconds with the earlier model
- Would have to deal with multiple problems like crossovers for each order being tracked if it has features similar to the data used for demonstration

4.5 Smart Algorithm

The computation time for the vacuum cleaner data with a single dominant frequency model as mentioned earlier was about 104 seconds for the chosen parameters on a computer with 8 GB RAM and 2.4 GHz processor. This is significantly higher than the processing time for a tachometer signal.

Hence it is necessary to develop a fast algorithm to make the entire process worth the effort and for this to be implemented in a software package. Without changing the processing techniques or the user interface for the algorithm, the computation has been considerably reduced by taking advantage of the inertia constraints of the system. While inertia constraints are something to be achieved in the spline fitting technique for tachometer data, it presents an opportunity to develop a smart algorithm in the Bayesian estimation process.

The maximum rate of change of speed is governed by the inertia of the system. This is the principle used to detect locking of car wheels in *anti – lock breakage systems* (ABS). As it is known that the speed can only change by a certain amount in a given time interval, the analysis can be reduced to a few frequency bins (whatever non Δf resolution is chosen) around the estimate of the previous block instead of considering the whole range $0 - F_{nyq}$.

Procedure for the Smart Bayesian Algorithm (SBA):

- For first two estimates (FFT blocks):
 - Use complete range of 0 to *frequency of interest* with a non – Δf resolution
 - Assign uniform prior distribution across the complete frequency range
- For further computing:
 - Use a short frequency range around the estimate of the previous block for likelihood estimation
 - User can choose the number of points on either side of the previous block estimate to involve in the analysis
 - Compute the *prior* function across 0 to *frequency of interest* with the same resolution
 - Consider *prior* for only the frequency range used for likelihood function
- While reprocessing:
 - Choose the block to begin the analysis
 - If the estimates of that block and the previous block seem to be correct, choose the non – uniform prior for the next estimate
 - Else, choose a uniform prior for the first two blocks during reprocessing and then apply the non – uniform prior for further blocks
- Repeat the above process till a satisfactory profile consistent with the spectral map is obtained

Advantages:

- Up to 100x faster: Run time significantly reduced

- 1.1 seconds with 10 side points as opposed to 104 seconds with earlier algorithm
- No contamination from frequencies out of surroundings of the dominant order
- Physical constraint on the rate of change of speed is naturally imposed
- Ability to include higher harmonics in the estimation which was not computationally viable earlier
- Results are identical to those from the previous algorithm

4.6 Comparison to Spline Fit Estimate:

The results from SBA are compared to those from the spline fit. The absolute difference and the percentage difference $\left(\frac{\text{Bayesian} - \text{Tach}}{\text{Tach}} \times 100\right)$ are presented.

Figure 35: SBA Estimate vs Spline Fit Estimate

(a)

(b)

Figure 36: Comparison of SBA Estimate vs Assigned Tach Estimate (a) Absolute Difference (b) Percentage Difference

From Figure 36, it is clear that there is a large error initially until about 5 seconds. The reason for this is a bad SBA estimate during that period which in turn is caused due to the order not being dominant. The amplitude based approach (most dominant frequency) for SBA has this limitation because the vibration / acoustic signal is buried in noise at very low speeds as the system is not excited to the required extent. Usually, this is seen within speeds of 500 RPM but may vary for each system. If the main objective is to analyse the system at higher speeds, this should not be a problem but if the analysis is to be carried out at low frequencies of about 10 Hz, either the signal to noise ratio needs to be increased or a tachometer must be employed. The SBA can be further developed in a future work to extract these buried signals. Also, the tachometer estimate may somehow be involved to keep track of the order. These are out of scope of this thesis and have been listed as recommendations for future work.

To test the algorithm further, it has been applied to data from a 6 cylinder engine powertrain and from the UC Structural Mechanics Lab rotor rig. The results are shown in Chapter 7 where a comparison is made with results from the *Adaptive n^{th} Pulse Algorithm* (AnPA).

5. THE n^{th} PULSE ALGORITHM (NPA)

5.1 Need for a Better Instantaneous Speed Estimation Process

Though the Bayesian estimate provides many benefits, there are several shortcomings due to which tachometers cannot entirely be avoided or replaced. For real time analysis (for example in a vehicle), a sophisticated computer is needed to analyse the data which might not be viable financially. Tachometers have become an integral part of the industry and do not require a skilled operator. As such, the Bayesian estimate would find more applications in the laboratory and academics for fault diagnosis or other similar applications. It would be beneficial to develop better estimation procedures from tachometers for the sake of practical applications in industry.

Conventionally, the final RPM profile from a tachometer data was generated by fitting a curve to the raw estimate. This is usually done because the raw estimate has rapid fluctuations which cannot occur in a real system due to inertia. The initial estimate from tachometer data using the conventional method (raw estimate) for a vacuum cleaner is shown in Figure 37.

Figure 37: Initial RPM Estimate Using Conventional Formulation

While curve fitting might be a good mathematical way of getting rid of fluctuations, the results may or may not be reliable. The approximation depends highly on the processing parameters chosen by the user and so can vary for each run. There is no absolute or consistent curve that can be generated as such. Also, it can be seen from Figure 38 that the Bayesian estimate seems to be a much better fit (smooth inflections) for the data than the mathematical curve fit itself.

(a)

(b)

Figure 38: Initial Raw Estimate vs (a) Spline Fit (b) SBA Estimates

Hence, it is necessary to develop a different method to address this issue for a better estimate. Order tracking methods use the RPM estimate to generate an amplitude of vibration versus rotation speed plot. This can then be used for operational mode shape analysis. Thus, any significant difference (might be small or large depending on the case and data set) from the true value would result in the estimation of a different amplitude during order tracking which might result in missing the peaks in the data. Noise could be tracked instead of a significantly steep peak if the speed estimate is off by a few frequency bins.

5.2 Analysis of the Cause for Fluctuations in the Instantaneous Speed Curve

To be able to understand the problem, it is necessary to revisit the formula used for instantaneous RPM extraction discussed in Chapter 3.

$$\text{Instantaneous RPM} = \frac{60}{t_2 - t_1}$$

It is important to note that the values t_2 and t_1 are approximated by linear interpolation between 2 points that are Δt apart. Hence the error that can occur in the estimation of a zero crossing ranges from $+\Delta t/2$ to $-\Delta t/2$. In this case, the sampling frequency $F_{\text{samp}} = 5120 \text{ Hz}$ and $\Delta t/2 \sim 0.0001 \text{ s}$. When this error is a significant percentage of the absolute value of time difference itself (say 2%), it causes the values to fluctuate. This is basically a sampling problem and the RPM curve when observed closely shows similar characteristics to bit noise.

Figure 39: Example: Fluctuations in the Initial Estimate at ~ 25000 RPM

Figure 39 is a zoomed in version of the vacuum cleaner data shown earlier. Consider a set of consecutive time instances recorded at zero crossings at around 24500 RPM. $t(2725) - t(2724) = 0.00244 \text{ s}$, $t(2726) - t(2725) = 0.00238 \text{ s}$ and so on. The corresponding RPM estimates are 24599 RPM and 25249 RPM assigned at 15.2629s and 15.2653 seconds respectively. Hence, in 0.002408 seconds, there has been a change of about 650 RPM which corresponds to a rate of change of 269933 RPM/second. This is impossible considering the inertial of the system. It is evident that the speed curve is not smooth because of the small variations in time recordings. A further zoomed in picture demonstrating the fluctuations is presented in Figure 40.

Figure 40: Range of Fluctuations in Speed Curve at ~ 25000 RPM

To increase the signal to noise ratio, one of the following can be done:

- Use better interpolation techniques to estimate zero crossings
- Increase the absolute value of time difference for numerical stability
- Increase the sampling rate to reduce the error in recording time samples

While increasing the sampling rate is the best option, increasing the absolute time difference between the zero crossings is less complicated and does not require sophisticated equipment.

5.3 Concept of the n^{th} Pulse Algorithm

When the time interval between successive pulses is so small that it is affected by the noise (at high speeds or very low rate of change of speed or a combination of these), it would make sense to skip ' n ' number of pulses to achieve a sufficiently large spacing that is not numerically as susceptible to noise. Thus the speed would now be estimated as,

$$Instantaneous\ RPM = \left(\frac{60}{t_2 - t_1} \right) * (n + 1) \quad (17)$$

Note that t_2 and t_1 are the time instances corresponding to the $(k + n + 1)^{th}$ and k^{th} pulses respectively and n is the number of pulses skipped. While in theory, the absolute time spacing would keep increasing with the increasing number of pulses skipped, it should be kept in mind that the underlying assumption is that the variation in the true rotating speed through the duration of the skipped pulses is much less. Hence, an optimum number of pulses to be skipped must be determined. This will be discussed further as part of AnPA in Chapter 6.

The result when 18 pulses were skipped is shown in Figure 41 and the reduction in fluctuations can be seen from Figure 42. A further zoomed in picture showing the range of fluctuations that is at the same scale as Figure 40 is shown in Figure 43. Clearly, the estimate has become much better with reduced fluctuations.

Figure 41: Instantaneous Speed Estimate with 18 Pulses Skipped

Figure 42: Reduction in Fluctuations at ~ 25000 RPM – 18 Pulses Skipped

Figure 43: Range of Fluctuations in the Speed Curve – 18 Pulses Skipped

5.4 Error analysis

To demonstrate the reduction in error by the use of this method, consider the example given earlier.

$$t(2725) - t(2724) = 0.00244 \text{ s} \quad (\text{A})$$

$$t(2726) - t(2725) = 0.00238 \text{ s} \quad (\text{B})$$

Since the error in each time estimate can be $\pm \Delta t/2$, the overall error in $t(k) - t(k - 1)$ for any 'k' can be $\pm \Delta t$. Now, the true value of instantaneous RPM must satisfy the following,

$$\frac{60}{t(k) - t(k - 1) + \Delta t} \leq \text{True value of RPM} \leq \frac{60}{t(k) - t(k - 1) - \Delta t} \quad (18)$$

Plugging in the values of time differences given in A and B into (18) gives,

$$22775.91 \leq \text{True A} \leq 26741.11$$

$$23331.37 \leq \text{True B} \leq 27510.06$$

The values might fluctuate anywhere within the specified range (almost 4000 RPM here). The time interval between these estimates is just 0.002408 seconds. The fluctuation would not usually lie to the extremities of the error range but as was calculated earlier might be around 650 RPM which is about 6.5 times lower than the maximum error possible. However, this uncertainty is still unacceptable and these fluctuations need to be minimized.

Now, consider 18 pulses skipped during the above given time period:

$$t(2745) - t(2725) = 0.048061 \text{ s} \quad (\text{C})$$

$$t(2746) - t(2726) = 0.048263 \text{ s} \quad (\text{D})$$

The corresponding speed ranges for C and D as estimated by Equation (18) are,

$$24766.77 \leq \text{True C} \leq 24968.06$$

$$24763.54 \leq \text{True D} \leq 24964.78$$

Here, the values fluctuate by a maximum of around 200 RPM. Again, the fluctuation usually does not lie in the extremities of this range and so, the change from one RPM to the next might

be well under this maximum value. In fact, the corresponding RPM values are 24867 RPM and 24863 RPM assigned at 15.2883 seconds and 15.2907 seconds respectively. Hence in a time interval of 0.00238 seconds, the speed changed by about 4 RPM which is a rate of change of about 1,680 RPM/second which is much more believable given that the system goes from 0 – 25000 RPM in 15 seconds.

5.5 Reasons for High Fluctuations

While it is established that the causes for fluctuations are sampling and the error in interpolation, it is important to investigate when such high fluctuations can be expected. It is clear that at higher speeds, the time difference between the pulses becomes less. Hence high speed is undoubtedly one of the reasons for high fluctuations. Now, looking at another data set taken from a six cylinder engine as shown in Figure 44, the system operates at speeds that are less than 2000 RPM and the slew rate is much slower compared to the vacuum cleaner data. Though the maximum fluctuation is only about 30 RPM, it is known that the system cannot change speed that quickly due to its inertia. The speed remains fairly constant over a significant period of time and so, during this period the pulses must ideally have about the same time difference. This does not happen due to the problems with sampling and interpolation which cause fluctuations that are significant for the scale of the problem.

Figure 44: Initial Raw Speed Curve – 6 Cylinder Engine Powertrain

Figure 45: Range of Fluctuations at ~ 1900 RPM – 6 Cylinder Engine Powertrain

5.6 Combined n^{th} Pulse Curve

Skipping ' n ' pulses provides a smooth instantaneous speed curve that can be seen to have less fluctuation. However, considerable data is being discarded in the process which can actually be used for a statistical advantage. There is no reason to believe that it is beneficial to start with one particular pulse and skip ' n ' pulses rather than to start with the next pulse and skip ' n ' pulses

and so on to starting with the $(n - 1)^{th}$ pulse and skipping 'n' pulses. Thus, there are $n + 1$ estimates of speed over the same data set when skipping the same 'n' number of pulses.

This is represented pictorially in the following figures (Figure 46 through Figure 50) with the corresponding speed curves. For convenience, the examples with square waves show 3 pulses skipped but in the corresponding estimate figures, 40 pulses were skipped.

Figure 46: Raw Instantaneous RPM – No Pulses Skipped

Figure 47: Starting at Pulse 1 – 40 Pulses Skipped

Figure 48: Starting at Pulse 2 – 40 Pulses Skipped

Figure 49: Starting at Pulse 3 – 40 Pulses Skipped

Figure 50: Starting at Pulse 4 – 40 Pulses Skipped

It can be seen that the curves with the process starting at different pulses have little to distinguish between them which reinforces our belief that there is no reason to start with one pulse as opposed to another. One use that immediately comes to mind is to average the estimates for a smooth RPM curve. However, this cannot be done straight away as the time instances associated with the estimated speeds are not the same for all the curves. This will be clear when considering the cases of starting with 1st, 21st and 40th pulses as shown in Figure 51.

Figure 51: (Clockwise from above) Instantaneous Speed Estimate with the n^{th} Pulse Process Staring from 1st, 21st and 40th pulses

Different estimates of speed are assigned to different time instances and so a logical way to use all of this information is to combine them as shown in Figure 52. It can also be seen that this provides the benefit of obtaining an instantaneous speed estimate at intervals consistent with the case where every pulse is used without skipping. This is very convenient as the instantaneous velocity will now be estimated at the same sampling times as any other response data that is recorded. This is due to the fact that the combinations all line up at consecutive pulse locations.

Figure 52: From Top to Bottom: Combination of Assigned Speeds at Different Time Instances; Combined n^{th} Pulse Curve with 40 Pulses Skipped; Zoomed in Version Showing Reduction in Fluctuations

Clearly, the curve from Figure 52 has a superior signal to noise ratio as compared to the raw estimate shown in Figure 39. Fitting a spline to this curve would be numerically much easier and physically more compatible than to the raw estimate.

Initial Time Lag in the Combined Instantaneous Speed Curve:

If too many pulses are skipped initially, the first data point in the RPM profile would not start at '0' but at some point which is at a significant time interval away. From there on, the estimates will be closely spaced again as desired. This initial loss of data was ignored in the present research work.

5.7 Averaging Technique

For averaging the different estimates, the RPM values need to correspond to the same time instances. This can be achieved by simply taking the time points associated with one estimate as the reference (for convenience the estimate starting from the 1st pulse) and then interpolating all other curves to the RPM values at the time instances corresponding to this reference curve. The function 'spline' can be used to do this in MATLAB. This process is shown in Figure 53 and Figure 54.

Figure 53: Process Starting at Different Pulses – Same No. of Pulses Skipped

Figure 54: Interpolated Estimates – Process Starting at Different Pulses, Brought to the Same Time Reference

Once all the curves are rearranged, they can be averaged. The resulting curve is shown in Figure 55. It can be seen that the curve does not need a spline fit. Any noise on the individual curves will be random and this technique will produce a sufficiently smooth curve. However, the number of pulses skipped must be optimum for this to happen. If in this particular example, 10 pulses are skipped instead of 40, the resulting combined as well as averaged curves will still have some significant noise that needs to be dealt with. Chapter 6 introduces an algorithm which can optimize the number of pulses to be skipped at various points in a given data.

Possible Shortcomings of the Averaging Technique:

If too many pulses are skipped, there is a significant decrease in resolution which might lead to missing any subtle inflections in the speed. No system with such rapid speed changes was analysed for this research work and so it was not possible to investigate this phenomenon further.

Figure 55: Averaged n^{th} Pulse Estimate – 40 Pulses Skipped

6. THE ADAPTIVE n^{th} PULSE ALGORITHM (ANPA)

6.1 Need for an Adaptive Algorithm

Though nPA gives superior signal to noise characteristics to that of the spline fit, there is an underlying assumption that during the period of every n fixed pulses throughout the span of the data, the speed remains fairly constant. This might not be the case throughout a given data or for all data sets. For example, at low speeds and high slew rates, though the conventional estimate generates a smooth enough curve, too many pulses would be skipped. This also causes varying time resolution throughout the curve. Moreover, the assumption of constant speed might not be satisfied. Any smooth inflections might be missed as a result of the skipping.

Figure 56 demonstrates the effect of skipping too many pulses. Almost 1 second of data is skipped in a total data set of 30 seconds when 40 pulses are skipped and the speed is below 5000 RPM. At about 26000 RPM, the interval between estimates is 0.1 seconds. This demonstrates the need for an adaptive algorithm which somehow takes this time resolution issue into account while skipping pulses.

Figure 56: Differences in Time Resolution with Constant No. of Pulses Skipped

6.2 Concept of the Adaptive Algorithm

If the time resolution of the curve is maintained fairly constant throughout the data, the instantaneous RPM curve can be transformed from a function of angle of rotation of the shaft (pulses and corresponding times) into a function of absolute time with a defined resolution. This would also provide an optimization for the number of pulses to be skipped.

Observing the raw estimate, it can be seen that the time resolution changes significantly for the *nPA* estimate at high speeds or at low rates of change of speed. In the raw tachometer estimate, during periods of high fluctuations, the resolution is small and the curve is inherently smoother during periods with low resolution.

Previously, the problem was viewed as a signal to noise ratio problem where the solution was to skip several pulses. To decide how many pulses are to be skipped, the problem should now be viewed from a different perspective, that is, from the time resolution perspective.

6.3 Development of the Algorithm

Consider the following rotating system characteristics:

- System rotating at ' R ' RPM.
- Number of pulses recorded by tachometer per second = $R/60$
- Therefore, time period for each pulse = $60/R$
- Time interval between ' n ' pulses = $60 \times n/R$

Now, the entire time data can be divided into a number of time blocks and for a desired time resolution, the number of pulses n to be skipped for each block can be identified as the integer number which provides the time interval between estimates ($60 \times n/R$) that is closest to the desired resolution. The smaller the block, the better will the estimate. This is because the assumption of fairly constant speed will easily hold for smaller blocks. However, if the block size becomes too small, it might not even be possible to skip a certain number of pulses as suggested by the above method. Hence, the user must be careful in choosing the number of blocks to divide the data into.

The result is shown in Figure 57 with consecutive points at different locations that are consistently 0.05 seconds apart. This was the user input on desired time resolution. The corresponding number of pulses skipped as a function of time is shown in Figure 58.

Figure 57: Consistent Time Resolution with AnPA

Figure 58: AnPA - Pulses Skipped vs Time

Effect of Duty Cycle:

A whole revolution of the system need not be counted for the instantaneous speed formulation.

For example, in the square wave output, a portion of the revolution can also be used by

considering the zero crossing at the end of the pulses. Thus, n need not necessarily be a whole number (E.g. 2.3 pulses or 7.8 revolutions can also be skipped). The advantage of this would be better control and accuracy in maintaining the time resolution.

The fraction of revolution completed from one end of a pulse to the other end can be easily found using the duty cycle for the output. Duty Cycle is defined as the percentage of time period of the output during which the pulse characteristic is active as shown in Figure 59.

$$\text{Duty Cycle} = \frac{\text{Time Duration of the Pulse}}{\text{Total Time Period of the Signal}} \times 100 \% \quad (19)$$

Figure 59: Duty Cycle

In case of an IR tachometer, the duty cycle is determined by the width of the reflective strip used. Note that this thought process can be adjusted if there is more than one pulse per revolution. This was not implemented in the current work but only a minor change to the code is required.

While the process of finding the optimum resolution can be automated to give the best speed profile, for the scope of this thesis, a user input is used for desired resolution to be maintained

throughout the profile. The resulting curve from the example shown in Figure 57 has no need for post processing and can be directly used for analysis. However, an option for spline fitting the result is provided in the code, to be used if the optimum resolution is not selected by the user. The resulting curve will generally have better noise characteristics as compared to the raw estimate and so curve fitting must only become less complicated and more appropriate.

For further improvement of the spline fit, an option for using different processing parameters for different regions of the curve has been provided in the final code (Refer to Appendix B). This improvement also has the effect of adapting to the change in speed and the rate of change of speed across various regions of the profile.

Checking for the Correctness of the Estimate:

The AnPA estimate is shown over the raw estimate in Figure 60.

Figure 60: AnPA Estimate vs. Raw Estimate

It can be seen that the curve is within bounds of the fluctuations throughout the data, something which can be a challenge to achieve with the spline fit. However, this alone cannot be taken as a check for correctness of the results. An independent estimate from a different method is required for comparison. SBA provides a method which is completely independent of tachometers. This is used for comparison in Chapter 7. For completeness, the results from AnPA and SBA for various data sets and systems are also provided.

7. COMPARISON

Comparison with absolute and percentage difference between AnPA and SBA has been made for several data sets. Examples include a vacuum cleaner, a 6 cylinder engine powertrain and a rotor rig setup that was shown in Figure 9. For completeness, individual results from the algorithms are also given for all cases except the vacuum cleaner.

Vacuum Cleaner:

Figure 61: Vacuum cleaner – Comparison SBA Estimate vs AnPA Estimate

(a)

(b)

Figure 62: Vacuum cleaner – Comparison SBA Estimate vs AnPA Estimate (a)

Absolute Difference (b) Percentage Difference

The difference is mostly during low speeds where the speed is so low that the vibration response is buried in noise. The error from there on mostly lies within 0.1 % and 25 RPM. Maximum error is about 300 RPM and 1.75 %.

6 Cylinder Engine Powertrain:

AnPA: 40 pulses maximum, 100 block divisions, 1 second time resolution

Figure 63: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution at Various Locations in the Estimate

(a)

(b)

Figure 64: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

(a)

(b)

Figure 65: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

SBA: BS – 4096, Initiation block - middle block, 0.05 Hz increment, Variance – 0.15, Overlap – 93.5%. No reprocessing was required for the data.

(a)

(b)

Figure 66: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped on the Time Spectral Map

Comparison:

Figure 67: 6 Cylinder Engine Powertrain – Comparison SBA Estimate vs AnPA Estimate

Figure 68: 6 Cylinder Engine Powertrain – Comparison SBA Estimate vs AnPA Estimate (a) Absolute Difference (b) Percentage Difference

There is very little difference between the two estimates. Maximum difference is around 2 RPM and the percentage difference lies within 0.1 % for most part of the data.

SM Lab Rotor Rig – Data set 1:

AnPA: 40 pulses maximum, 250 time blocks, 0.05 seconds time resolution

**Figure 69: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution
at Various Locations in the Estimate**

(a)

(b)

Figure 70: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

(a)

(b)

Figure 71: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

SBA: BS – 4096, Initiation block – 250, Variance – 0.15, Overlap – 93.5%. Reprocessing was required, refer to Appendix A for details.

(a)

(b)

Figure 72: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped on the Time Spectral Map

Comparison:

Figure 73: SM Lab Rig Data Set 1 – Comparison SBA Estimate vs AnPA Estimate

Figure 74: SM Lab Rig Data Set 1 – Comparison SBA Estimate vs AnPA Estimate

(a) Absolute Difference (b) Percentage Difference

SM Lab Rotor Rig – Data set 2:

AnPA: 40 pulses maximum, 250 time blocks, 0.05 seconds time resolution

**Figure 75: From Top to Bottom – Raw Estimate; AnPA Estimate; Time Resolution
at Various Locations in the Estimate**

(a)

(b)

Figure 76: (a) AnPA Estimate vs Raw Estimate (b) Combined Curve

(a)

(b)

Figure 77: (a) Combined Curve showing Reduced Fluctuations (b) No. of Pulses Skipped as a Function of Time

SBA: BS – 4096, Initiation block – middle block, Variance – 0.15, Overlap – 93.5%. No reprocessing was required.

(a)

(b)

Figure 78: (a) SBA Estimate vs Raw Estimate (b) SBA Estimate Overlapped over the Time Spectral Map

Comparison:

Figure 79: SM Lab Rig Data Set 2 – Comparison SBA Estimate vs AnPA Estimate

Figure 80: SM Lab Rig 2 – Comparison SBA Estimate vs AnPA Estimate (a)

Absolute Difference (b) Percentage Difference

The difference is mostly within 15 RPM and the percentage difference is within 0.2 %. Maximum difference is about 30 RPM and 0.5%.

From these illustrations it can be seen that both AnPA and SBA are capable of generating reliable results that closely agree with each other.

8. SUMMARY

8.1 Conclusion

The conventional way of estimating instantaneous speed involves fitting a curve (spline) to an initial estimate for which each consecutive pulse is considered in the formulation. The sampling process and error in interpolation give rise to unacceptable fluctuations in the instantaneous speed profile. The *Adaptive n^{th} Pulse Algorithm* (AnPA) is an effective new technique which can address this issue. The main idea behind the algorithm is to increase the absolute time interval over which a single speed estimate is being calculated so that it is not affected by the errors (zero crossing uncertainty). The number of pulses to be skipped for different parts of the data has to be regulated according to the speed and rate of change of speed. This optimization is achieved by matching the time interval between estimates to a desired time resolution. The averaging technique presented generates a smooth curve which does not require a spline fit. The combination technique generates a curve with high signal to noise ratio to which a spline can be fit easily. It is a fairly simple procedure to apply and the operator/ user does not have to be skilled in rotating systems analysis to understand it.

The Bayesian estimation approach provides a way of estimating the speed from vibration data without the need for a tachometer. This can prove highly advantageous for applications with constraints like space, cost or unfavourable operating conditions. The *Smart Bayesian Algorithm* (SBA) developed in this work greatly reduces the analysis required and thus brings down the processing time by as much as 100 times. The most important contribution of this algorithm is the ability to reprocess with different parameters over different parts of the data. The results have a high signal to noise ratio and can directly be used for generation of RPM spectral maps. There is no need for approximations and assumptions in the assignment of a

particular speed to a given data block [8]. Though the benefits from this method are attractive, there are several shortcomings like the requirement of a high speed computer and a skilled operator due to which it might not be viable for all practical real time applications.

The comparisons presented in Chapter 7 show that both AnPA and SBA arrive at similar speed profiles although the processes are independent of each other. Also, the curves always lie within bounds of the raw tachometer estimate and seem to be a better fit to the data than the mathematical spline fit. It should be noted however that there is no way to know the true value of instantaneous speed. As such, no error analysis can be done with respect to the true value. These two methods support each other and provide results that are more physically reasonable than the spline fit approach.

8.2 Recommendations for future work

Smart Bayesian Algorithm:

SBA requires further development to overcome several limitations. The model chosen for this thesis work assumes that there is only one dominant harmonic in the data. For many systems, this is not the case. More harmonics can be included to make the estimation procedure more robust. This will cause lesser contamination from frequencies surrounding the orders being tracked.

A model can be developed which estimates the speed from different orders simultaneously and then corrects them by comparison based on a predefined tolerance in case of any contamination. This will ensure that the curve does not deviate from the orders of interest and makes the process more robust.

The above two ideas were not pursued in this work due to the amount of computation involved. However, the SBA is more efficient at handling huge data sets and thus can enable the involvement of higher orders into the model.

The *prior* function in the Bayesian formulation is based on two previous block estimates. Other functions can be developed which consider either a higher order extrapolation or more number of previous block estimates. A central difference approach can also be investigated.

Tachometer results were not used at all in this process as the aim was to develop a method that was independent of the conventional estimation approach. It might be beneficial to include the tachometer result as a corrector for the Bayesian estimate. This might find applications in systems with orders crossing each other which is the case with automobiles due to the presence of a differential. Also during gear changes in a system, an abrupt change in orders might occur in which case the algorithm developed in this thesis is not sufficient.

Adaptive n^{th} Pulse Algorithm:

For AnPA, one major improvement that can be made is the automation of adaptive pulse skipping. For the scope of this thesis, user inputs are taken on the desired time resolution and number of blocks to divide the data into. This process can be automated through optimization algorithms so that the only input required would be the tachometer data file. Methods like the Bayesian probability theory can be used to assess and rank different models (combinations of time resolution and divisions) to achieve this.

The *Adaptive n^{th} Pulse Algorithm* and the processes involved constitute a new concept and has been investigated in this thesis for the first time. There could be improvements other than the ones suggested here that might be required to successfully adapt the procedure to a different data set.

REFERENCES

- [1] Blough, J. R., *Improving the Analysis of Operating Data on Rotating Automotive Components*, PhD Dissertation, University of Cincinnati, 1998.
- [2] Pedersen, T. F., Herlufsen, H., Konstantin-Hansen, H., *Order Tracking in Vibro-acoustic Measurements: A Novel Approach Eliminating the Tacho Probe*, SAE paper 01-2266, 2005.
- [3] Bretthorst, G. L., *Bayesian Spectrum Analysis and Parameter Estimation*, Vol. 48 of Lecture Notes in Statistics, Springer – Verlag, 1988.
- [4] Schuster, A., *The Periodogram and its Optical Analogy*, Proceedings of the Royal Society of London, 77, pp. 136, 1905.
- [5] Ruanaidh, J. J. K., Fitzgerald, W. J., *Numerical Bayesian Methods Applied to Signal Processing, Statistics and Computing*, Springer, 1998.
- [6] Jeffreys, H., *Theory of Probability*, Oxford University Press, London, 1939, (Later editions, 1948, 1961).
- [7] Allemang, R. J., Course Notes, *Rotating System Measurements*, Structural Dynamics Research Laboratory, University of Cincinnati, 2014.
- [8] Deshpande, S., *Improving Observability in Experimental Analysis of Rotating Systems*, Master's Thesis, University of Cincinnati, 2014.
- [9] Jaynes, E. T., *Marginalization and Prior Probabilities in Bayesian Analysis in Econometrics and Statistics*, A. Zellner, (ed.), North-Holland Publishing Company, Amsterdam, 1980.
- [10] Vold, H., Mains, M., & Blough, J. R., *Theoretical Foundations of High Performance Order Tracking with the Vold-Kalman Filter*, Noise and Vibration Conference, SAE Paper number 972007, 1997.
- [11] James, M. L., *Applied Numerical Methods for Digital Computation*, New York, New York: Harper & Row Publishers Inc, 1985.
- [12] Gade, S., Herlufsen, H., Konstantin-Hansen H., Vold, H., *Characteristics of the Vold – Kalman Order Tracking Filter*, Brüel & Kjær Technical Review No. 1 – 1999.

[13] Vemuri, A., *Estimation of Instantaneous Speed for Rotating Systems: New Processing Techniques*, Proceedings, International Modal Analysis Conference, 2016.

APPENDIX A

Applying FFT Principles to Non – Stationary Data:

For non – stationary data, it might not always be appropriate to make inferences from the frequency domain results obtained from a Fourier series approach. Though the results might be close to those inferred from the time domain in most cases, they might not be acceptable in some situations. An example of this is can be seen when analysing the dominant harmonics for Bayesian estimation following the frequency domain principals such as the Rayleigh criterion for Δf .

Figure 81: (a) Stepped and (b) Smooth Bayesian Estimates

The curve shown in Figure 81 was extracted from accelerometer data using the Bayesian algorithm. The frequency content changes so quickly in this example that the sweep is not captured when using the usual Δf spacing. When a spacing of about $\Delta f/10$ (no particular reason for the fraction chosen) is used, the estimate becomes much better.

Demonstration of the Smart Bayesian Algorithm - Reprocessing

The reprocessing step in the Smart Bayesian Algorithm developed in this thesis is especially useful when analysing noisy data. The data presented in Figure 72 is from a microphone. The signal is clearly contaminated by noise. This can cause problems when an initial block is to be selected. As expected, the initial results were bad but were improved during reprocessing as shown in Figure 82 and Figure 83. The option to limit the analysis to a certain frequency range was used here (5000 – 8000 RPM in this case). This ensured that the estimation process would be limited to the frequency region in which the order is the most dominant.

Figure 82: Microphone Data – (a) Final and (b) Initial SBA Estimates over the Time Spectral Map

Final estimate – reprocessed

(a)

Initial estimate

(b)

Figure 83: Microphone Data – (a) Final and (b) Initial SBA Estimates

APPENDIX B

MATLAB Script

```
% ACHYUT VEMURI
% MS Thesis
% Estimation of Instantaneous Speed for Rotating Systems: New Processing
Techniques
% Structural Dynamics Research Lab
% University of Cincinnati

%% Loading data
clear; clc; close all;
load('Data_File_Name.mat')
Fnyq = Fsamp/2;
deltaT = 1/(2*Fnyq);
% Data import has to be updated for each data problem manually
% Tach signal processing requires an Fsamp. If it is not directly given,
% be sure to obtain it from Fnyq or deltaT

%% Tach signal
figure, plot(Tach1(ceil(end/2)-1000:ceil(end/2)))
Tachtype = menu('Type of Tach output','Variable reluctance: Sine
output','Square output');
npulses = input('How many pulses per rev\n');
n = 0; m = 0;
if Tachtype == 1
    Tach1 = Tach1/max(Tach1);
    figure, plot(Tach1(ceil(end/2)-1000:ceil(end/2)))
    modTach = menu('Multiply by "-1" to get peak-valley
characteristic?','Yes','No');
    MPH = input('Give the minimum peak height (0 to 1)\n');
    if modTach ==1, Tachmod = -Tach1; else Tachmod = Tach1;end
    [peaks, locations] = findpeaks(Tachmod,'MINPEAKHEIGHT',MPH);
    for k = locations
        if k == size(Tach1,2)
            break
        end
        while true
            if Tachmod(k)>0 && Tachmod(k+1)<0
                n = n+1;
                t1(n) = ((0-Tachmod(k))*(1/Fsamp)/(Tachmod(k+1)-
Tachmod(k)))+(k*1/Fsamp);
                break
            else k = k+1;
            end
        end
    end
elseif Tachtype == 2
    while true
        modTach2 = menu('Preprocessing','Oscillate about 0','Negate a DC
value');
        if modTach2 == 1
            Tach1 = Tach1-((max(Tach1)+min(Tach1))/2);
```

```

        else DC = input('Give DC value\n');
            Tach1 = Tach1- DC;
        end
        figure, plot(Tach1(ceil(end/2)-1000:ceil(end/2)))
        plotok = menu('Is this ok?', 'Yes', 'No');
        if plotok == 1, break; end
    end
    for k = 1:size(Tach1,2)
        if k == size(Tach1,2)
            break
        end
        if Tach1(k)<0 && Tach1(k+1)>0
            n = n+1;
            t1(n) = ((0-Tach1(k))*(1/Fsamp)/(Tach1(k+1)-
Tach1(k)))+(k*1/Fsamp);
        elseif Tach1(k)>0 && Tach1(k+1)<0
            m = m+1;
            t2(m) = ((0-Tach1(k))*(1/Fsamp)/(Tach1(k+1)-
Tach1(k)))+(k*1/Fsamp);
        elseif Tach1(k)==0
            if Tach1(k-1)<0
                n = n+1;
                t1(n) = (k*1/Fsamp);
            elseif Tach1(k-1)>0
                m = m+1;
                t2(m) = (k*1/Fsamp);
            end
        end
    end
end
t1 = t1(1:npulses:end);
rawirpm = zeros(1,size(t1,2)-1); rawTime = zeros(1,size(t1,2)-1);
for k = 1:size(rawirpm,2)
    rawirpm(k) = 60/(t1(k+1)-t1(k));
    rawTime(k) = (t1(k+1)+t1(k))/(2);
end
figure
plot(rawTime,rawirpm)
grid
title('Raw RPM estimate from Tach signal')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')

%% Adaptive nth pulse algorithm
clear irpm1 Time1 Timeavgasgn Timecombstore Timerefstore Timestore
irpmavgasgn irpmavgstore irpmcombstore irpmstore ranking irpmavgstore
Timerefstore irpmrespline
pulses = input('Maximum no.of pulses to be skipped for Adaptive
algorithm\n')+1;
for pulses = 1:pulses
    clear irpmstore Timestore
    counterirpmstart = 0;
    for startingpulse = 1:pulses
        t2 = t1(startingpulse:pulses:end);
        irpm1 = zeros(1,size(t2,2)-1); Time1 = zeros(1,size(t2,2)-1);
        for k = 1:size(irpm1,2)
            irpm1(k) = 60/(t2(k+1)-t2(k));
            Time1(k) = (t2(k+1)+t2(k))/(2);
        end
        irpm1 = irpm1*pulses;
        if counterirpmstart == 0

```

```

    maxsizeirpm = size(irpm1,2);
    makenote = startingpulse;
end
if size(irpm1,2)>maxsizeirpm
    for iter = 1:startingpulse-1
        padding = -20000*ones(1,size(irpm1,2)-maxsizeirpm);
        irpmstore(iter,:) = [irpmstore(iter,:) padding];
        Timestore(iter,:) = [Timestore(iter,:) padding];
    end
elseif size(irpm1,2)<maxsizeirpm
    padding = -20000*ones(1,maxsizeirpm-size(irpm1,2));
    irpm1 = [irpm1 padding];
    Time1 = [Time1 padding];
end
if size(irpm1,2)>maxsizeirpm, maxsizeirpm = size(irpm1,2);makenote =
startingpulse;end
irpmstore(startingpulse,:) = irpm1;
Timestore(startingpulse,:) = Time1;
counterirpmstart = 1;
end
for k = 1:size(irpmstore,2)
    irpm1((k-1)*pulses+1:pulses*k) = irpmstore(:,k);
    Time1((k-1)*pulses+1:pulses*k) = Timestore(:,k);
end
irpm1 = irpm1(irpm1~-20000);
Time1 = Time1(Time1~-20000);
irpmcombstore(pulses,1:size(irpm1,2)) = irpm1;
Timecombstore(pulses,1:size(irpm1,2)) = Time1;
irpmavg = zeros(1,size(irpmstore,2));
Timeref = Timestore(makenote,:);
Timeref = Timeref(Timeref~-20000);
for k = 1:size(irpmstore,1)
    Timestore1 = Timestore(k,:); irpmstore1 = irpmstore(k,:);
    Timestore1 = Timestore1(Timestore1~-20000);irpmstore1 =
irpmstore1(irpmstore1~-20000);
    irpmavg = irpmavg + spline(Timestore1,irpmstore1,Timeref);
end
irpmavg = irpmavg/k;
irpmavgstore(pulses,1:size(irpmavg,2)) = irpmavg;
Timerefstore(pulses,1:size(irpmavg,2)) = Timeref;
irpmrespline(pulses,:) = spline(Timeref,irpmavg,Timerefstore(1,:));
end
pulseblocks = input('How many partitions to make for adaptive
algorithm?\n');
timeres = input('What is the desired time resolution (seconds)\n');
inpulseblocks = floor(size(Timerefstore,2)/pulseblocks);
timerespulses = zeros(pulses,pulseblocks+1); ranking =
zeros(1,pulseblocks+1);
for k = 1:pulseblocks+1
    if k == pulseblocks+1
        irpmmean = mean(irpmavgstore(1,inpulseblocks*(k-1)+1:end));
    else
        irpmmean = mean(irpmavgstore(1,(k-
1)*inpulseblocks+1:k*inpulseblocks));
    end
    timerespulses(:,k) = 60*((1:pulses).')/irpmmean;
    [discard, ranking(k)] = min(abs(timeres-timerespulses(:,k)));
end
irpmaveraged = []; Timeaveraged = []; Timeforpulseblocks =
zeros(1,pulseblocks+1);
for k = 1:pulseblocks+1

```

```

    Timetemp = Timerefstore(ranking(k),:); irpmttemp =
    irpmavgstore(ranking(k),:);
    Timetemp = Timetemp(Timetemp~=0); irpmttemp = irpmttemp(irpmttemp~=0);
    if k == pulseblocks+1
        [discard, in1] = min(abs(Timerefstore(1,(k-1)*inpulseblocks+1)-
    Timetemp));
        while true
            if k~=1 && Timetemp(in1) <= Timeaveraged(end)
                in1 = in1 + 1;
            else break
            end
        end
        [discard, in2] = min(abs(Timerefstore(1,end))-Timetemp);
    else
        [discard, in1] = min(abs(Timerefstore(1,(k-1)*inpulseblocks+1)-
    Timetemp));
        while true
            if k~=1 && Timetemp(in1) <= Timeaveraged(end)
                in1 = in1 + 1;
            else break
            end
        end
        [discard, in2] = min(abs(Timerefstore(1,k*inpulseblocks)-
    Timetemp));
        end
        irpmblock = irpmttemp(in1:in2); Timeblock = Timetemp(in1:in2);
        Timeforpulseblocks(k) = Timeblock(floor((1+size(Timeblock,2))/2));
        irpmaveraged = [irpmaveraged irpmblock]; Timeaveraged = [Timeaveraged
    Timeblock];
    end

figure
plot(Timeaveraged,irpmaveraged)
grid
title(['Averaged adaptive RPM estimate skipping upto ',num2str(pulses-1),'
pulses'])
xlabel('Time(sec)')
ylabel('Instantaneous speed (RPM)')
pulsechoice = input(['Show combined RPM for how many pulses skipped? (1 -
',num2str(pulses-1),')\n'])+1;
Timeforcombplot = Timecombstore(pulsechoice,:); Timeforcombplot =
Timeforcombplot(Timeforcombplot~=0);
irpmforcombplot = irpmcombstore(pulsechoice,:); irpmforcombplot =
irpmforcombplot(irpmforcombplot~=0);
figure
plot(Timeforcombplot,irpmforcombplot)
grid
title(['Combined RPM estimate skipping ',num2str(pulsechoice-1),' pulses'])
xlabel('Time(sec)')
ylabel('Instantaneous speed (RPM)')
figure
plot(Timeforpulseblocks,ranking-1)
grid
title('No. of pulses skipped as a function of time')
xlabel('Time (sec)')
ylabel('Pulses skipped')
figure
plot(rawTime,rawirpm,'green', Timeaveraged, irpmaveraged, 'red')
grid, legend('Raw Tach','Adaptive nth pulse')
title('Comparison of Raw Tach estimate vs. Adaptive nth pulse estimate')
xlabel('Time(sec)')

```

```

ylabel('Instantaneous speed estimate (RPM)')

%% Spline fit
splinefit = menu('Fit a spline?', 'Yes', 'No');
if splinefit == 1
restart = 1;
while restart == 1
forspline = menu('Choose', 'Combined nth pulse RPM', 'Adaptively averaged
RPM');
if forspline == 2, irpm = irpmaveraged; Time = Timeaveraged;
else
pulsechoice = input(['Show combined RPM for how many pulses? (1 -
', num2str(pulses), ')\n']);
Timeforcombplot = Timecombstore(pulsechoice,:); Timeforcombplot =
Timeforcombplot(Timeforcombplot~=0);
irpmforcombplot = irpmcombstore(pulsechoice,:); irpmforcombplot =
irpmforcombplot(irpmforcombplot~=0);
irpm = irpmforcombplot; Time = Timeforcombplot;
end
loop = 1; re = 0;
while loop == 1
if re == 0
BSt = input(['No.of points in raw Tach estimate are
', num2str(size(irpm,2)), '.\nGive the block size of divisions for spline
fitting\n']);
end
if re > 0
pick = menu('Pick regions to apply spline?', 'Yes', 'No');
if pick == 1
figure
plot(Time, irpm)
grid
title('Spline fit RPM(blue) vs Shaved RPM(red)')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
[Xpoint, Ypoint] = ginput(2);
diff1 = abs(Time - Xpoint(1)); diff2 = abs(Time - Xpoint(2));
diff3 = abs(Time1 - Xpoint(1)); diff4 = abs(Time1 - Xpoint(2));
[Xval1, point1] = min(diff1); [Xval2, point2] = min(diff2);
pointtemp = point2; if point1>point2, point2 = point1; point1 =
pointtemp;end
% [Xval3, point3] = min(diff3); [Xval4, point4] = min(diff4);
BSt = input(['No.of points in selected region (present
iteration: ', num2str(re+1), ') are ', num2str(abs(point1-point2)), '.\nGive
the block size of divisions for spline fit\n']);
else
BSt = input(['No.of points in updated estimate (present iteration:
', num2str(re+1), ') are ', num2str(size(irpm,2)), '.\nGive the block size of
divisions for spline fit\n']);
end
else pick = 2;
end
blockpoints = input('Give no.of points to fit in each block\n');
if pick == 2
Nt = floor(size(irpm,2)/BSt)-1;
counter = 1;
time = Time(1:BSt);
XX = linspace(time(1), time(end), blockpoints);
xx(counter:counter-1+size(XX,2)) = XX;
irpmt = irpm(1:BSt);
irpm2(counter:counter-1+size(XX,2)) = spline(time, irpmt, XX);

```

```

counter = size(xx,2)+1;
for k = 2:Nt
    time = Time(BSt*(k-1):k*BSt);
    XX = linspace(time(2), time(end), blockpoints);
    xx(counter:counter-1+size(XX,2)) = XX;
    irpmt = irpm(BSt*(k-1):BSt+(k-1)*BSt);
    irpm2(counter:counter-1+size(XX,2)) = spline(time,irpmt,XX);
    counter = size(xx,2)+1;
end
time = Time(Nt*BSt:end);
XX = linspace(time(2), time(end), blockpoints);
xx(counter:counter-1+size(XX,2)) = XX;
irpmt = irpm(BSt*Nt:end);
irpm2(counter:counter-1+size(XX,2)) = spline(time,irpmt,XX);
end
if pick == 1
    Nt = floor((abs(point1-point2))/BSt)-1;
    counter2 = 1;
    for k = 1:Nt
        time = Time(point1+BSt*(k-1):point1+k*BSt);
        XX = linspace(time(2), time(end), blockpoints);
        xxt(counter2:counter2-1+size(XX,2)) = XX;
        irpmt = irpm(point1+BSt*(k-1):point1+k*BSt);
        irpm2t(counter2:counter2-1+size(XX,2)) = spline(time,irpmt,XX);
        counter2 = size(xxt,2)+1;
    end
    time = Time(point1+Nt*BSt:point2);
    XX = linspace(time(2), time(end), blockpoints);
    xxt(counter2:counter2-1+size(XX,2)) = XX;
    irpmt = irpm(point1+BSt*Nt:point2);
    irpm2t(counter2:counter2-1+size(XX,2)) = spline(time,irpmt,XX);
    xx(1:point1) = Time(1:point1);
    xx(point1+1:point1+size(xxt,2)) = xxt;
    xxtemp = Time(point2+1:end);
    xx(point1+size(xxt,2)+1:point1+size(xxt,2)+size(xxtemp,2)) =
xxtemp;
    irpm2(1:point1) = irpm(1:point1);
    irpm2(point1+1:point1+size(xxt,2)) = irpm2t;
    irpm2temp = irpm(point2+1:end);
    irpm2(point1+size(xxt,2)+1:point1+size(xxt,2)+size(irpm2temp,2)) =
irpm2temp;
end
% Shaving the data
shave = menu('Shave data?', 'Yes', 'No');
if shave == 1
    if pick == 2
        irpm3expand = spline(xx,irpm2,Time);
        variation1 = input('Give the lower limit of variation allowed
in RPM\n');
        variation2 = input('Give the upper limit of variation allowed
in RPM\n');
        irpm3 = irpm3expand(abs(irpm3expand-irpm)>variation1 &
abs(irpm3expand-irpm)<variation2);
        xx3 = Time(abs(irpm3expand-irpm)>variation1 & abs(irpm3expand-
irpm)<variation2);
    end
    if pick == 1
        irpm3texpand = spline(xxt,irpm2t,Time(point1+1:point2));
        variation1 = input('Give the lower limit of variation allowed
in RPM\n');

```



```

        variation2 = input('Give the upper limit of variation allowed
in RPM\n');
        irpm3t = irpm3texpand(abs(irpm3texpand-
irpm(point1+1:point2))>variation1 & abs(irpm3texpand-
irpm(point1+1:point2))<variation2);
        xx3t = Time(point1+find(abs(irpm3texpand-
irpm(point1+1:point2))>variation1 & abs(irpm3texpand-
irpm(point1+1:point2))<variation2));
        xx3(1:point1) = xx(1:point1);
        xx3(point1+1:point1+size(xx3t,2)) = xx3t;
        xx3(point1+size(xx3t,2)+1:point1+size(xx3t,2)+size(xxtemp,2)) =
xxtemp;
        irpm3(1:point1) = irpm2(1:point1);
        irpm3(point1+1:point1+size(irpm3t,2)) = irpm3t;

irpm3(point1+size(irpm3t,2)+1:point1+size(xx3t,2)+size(irpm2temp,2)) =
irpm2temp;
    end
end
% Plots for user reference
figure
plot(Timeaveraged,irpmaveraged,'blue',xx,irpm2,'red')
grid, legend('Processed estimate','Spline fit')
title('Unshaved Spline fit RPM')
xlabel('Time(sec)')
ylabel('Instantaneous speed (RPM)')
if shave == 1
    figure
    plot(Timeaveraged,irpmaveraged,'blue',xx3,irpm3,'red')
    grid, legend('Processed estimate','Shaved spline fit')
    title('Shaved RPM(red)')
    xlabel('Time(sec)')
    ylabel('Instantaneous speed (RPM)')
end
loop = menu('One more iteration?','Yes','No');
if shave == 2
    irpm = irpm2;
    Time = xx;
    if loop == 1
        re = re + 1; clear xx xx3 xxt xx3t irpm2 irpm3 irpm2t irpm3t;
    end
else
    irpm = irpm3;
    Time = xx3;
    if loop ==1
        re = re + 1; clear xx xx3 xxt xx3t irpm2 irpm3 irpm2t irpm3t;
    end
end
end
restart = menu('Do you want to restart process?','Yes','No');
if restart ==1
    close all, clear irpm Time time irpmt xx xx3 xxt xx3t irpm2 irpm3
irpm2t irpm3t;
end
end
end
close all; clc

%% Bayesian
% Set - up
restart2 = 1; Ave = 1; Store = 1;

```

```

while restart2 == 1
accn = menu('Choose a source','1','2','3','4');
accel = data(accn,:);
setup = menu('Choose a block size','256','512','1024','2048','4096');
if setup == 1
    BS = 256;
elseif setup == 2
    BS = 512;
elseif setup == 3
    BS = 1024;
elseif setup == 4
    BS = 2048;
elseif setup == 5
    BS = 4096;
end

overlap = menu('overlap averaging','0%','50%','75%','87.5%','93.75%');
if overlap ~= 1
    if overlap == 2
        x1 = 50/100;
    end
    if overlap == 3
        x1 = 75/100;
    end
    if overlap == 4
        x1 = 87.5/100;
    end
    if overlap == 5
        x1 = 93.75/100;
    end
else x1 = 0;
end

t = 0:deltaT:(BS-1)*deltaT;
deltaF = 1/(BS*deltaT);
Ns = size(accel,2)/BS;
N = floor(((size(accel,2)-BS)/((1-x1)*BS))+1);
wcentralTime = (((2*((1:N)-1)*(1-x1)+1)*BS)+1)/2)*deltaT;
a0 = 1; a1 = 0.5;
hanning = a0 - (2*a1*cos(2*pi*t/(BS*deltaT)));
x = zeros(N,BS); Nd2 = zeros(1,N);
[B,A] = butter(5,[1/Fnyq 0.8],'bandpass');
accel = filter(B,A,accel);
for k = 1:N;
    x(k,:) = accel(BS*(1-x1)*(k-1)+1:BS+(k-1)*BS*(1-x1));
    Nd2(k) = sum(x(k,:).^2);
end

% FFT for spectral analysis
F1 = zeros(N,BS); F = zeros(N,BS/2+1); freq = (0:BS/2)*deltaF;
for k = 1:N
    F1(k,:) = fft(accel(BS*(1-x1)*(k-1)+1:BS+(k-1)*BS*(1-x1)));
    F1(k,2:BS/2) = 2*F1(k,2:BS/2);
    F(k,:) = F1(k,1:BS/2+1);
end
F = F/BS;
zcord = (max(max(abs(F)))+1)*ones(1,N);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Start Bayesian analysis

```

```

range = input('Give the operating speed range (upper limit in RPM)\n');
f = 0:deltaF/10:(range/60)+50; Nf = size(f,2);

% Time Spectral map
figure
mesh(freq,1:N,10*log10(abs(F)))
view(0,90), xlim([0 range/60+50])
colorbar
title(['Accel ',num2str(accn),' Time Spectral map with BS = ',num2str(BS),'
- ',num2str(x1*100),'% overlap'])
xlabel('Frequency (Hz)')
ylabel('Overlapping blocks')
zlabel('Amplitude (V)')

fprintf('Observe the Spectral map and make the following decisions\n\n');
dom = input('How many dominant frequencies? 1, 2 or 3\n');
if dom == 1, order = input('Which order is the dominant one? (not
necessarily integer)\n');end
if dom~=1, mul = input('What are the dominant harmonics\n');end
var = input('Give the variance\n');
start = input(['Choose the block to start analysis from. Block range : 2 to
',num2str(N),' \nCan also give in terms of fractions of total blocks (N)
E.g. N/2\n']);
sidepoints = input('How many points on either side of estimate to consider
for analysis?\n');
tic
start = floor(start);
f0 = 0; f1 = 0; index = zeros(1,N); Prior = ones(N,Nf);
for l = start:-1:start-1
    R_w = x(l,:)*(cos(2*pi*transpose(t)*(f))); I_w =
x(l,:)*(sin(2*pi*transpose(t)*(f)));
    C_w = (1/BS)*(R_w.^2+I_w.^2);
    C = (1-((2*C_w)/Nd2(1))).^(1-BS/20);
    A = C;
    I = find(A == max(A));
    if size(I,2) ~= 1
        I = I(1);
    end
    index(l) = I;
    f0 = f1; f1 = f(index(l));
    indexnew = index(l)-sidepoints:index(l)+sidepoints;
    if find(indexnew == 0), indexnew = 1:indexnew(end);
    elseif find(indexnew == size(f,2)), indexnew = indexnew(1):size(f,2);
    end
    fnew = f(indexnew);
end
for l = start-2:-1:1
    R_w = x(l,:)*(cos(2*pi*transpose(t)*(fnew))); I_w =
x(l,:)*(sin(2*pi*transpose(t)*(fnew)));
    C_w = (1/BS)*(R_w.^2+I_w.^2);
    C = (1-((2*C_w)/Nd2(1))).^(1-BS/20);
    Prior(l,:) = 1/sqrt(2*var*pi)*exp(-(f-2*f1+f0).^2/2*var);
    Priortemp = Prior(l,indexnew);
    A = Priortemp.*C/sum(Priortemp.*C);
    I = find(A == max(A));
    if size(I,2) ~= 1
        I = I(1);
    end
    index(l) = I+indexnew(1)-1;
    f0 = f1; f1 = f(index(l));

```

```

    indexnew = index(1)-sidepoints:index(1)+sidepoints;
    if find(indexnew == 0), indexnew = 1:indexnew(end);
    elseif find(indexnew == size(f,2)), indexnew = indexnew(1):size(f,2);
    end
    fnew = f(indexnew);
end
w(1:start) = f(index(1:start))*60;
f0 = w(start-1)/60;f1 = w(start)/60;
indexnew = index(start)-sidepoints:index(start)+sidepoints;
fnew = f(indexnew);
for l = start+1:N
    R_w = x(l,:)*(cos(2*pi*transpose(t)*(fnew)));I_w =
x(l,:)*(sin(2*pi*transpose(t)*(fnew)));
    C = (1/BS)*(R_w.^2+I_w.^2);
    C = (1-((2*C)/Nd2(l))).^(1-Bs/20);
    Prior(l,:) = 1/sqrt(2*var*pi)*exp(-(f-2*f1+f0).^2/2*var);
    Priortemp = Prior(l,indexnew);
    A = Priortemp.*C/sum(Priortemp.*C);
    I = find(A == max(A));
    if size(I,2) ~= 1
        I = I(1);
    end
    index(l) = I+indexnew(1)-1;
    f0 = f1; f1 = f(index(l));
    indexnew = index(l)-sidepoints:index(l)+sidepoints;
    if find(indexnew == 0), indexnew = 1:indexnew(end);
    elseif find(indexnew == size(f,2)), indexnew = indexnew(1):size(f,2);
    end
    fnew = f(indexnew);
end
w(start+1:N) = f(index(start+1:N))*60;
w = w/order;
toc
% Plots - Bayesian estimate
figure
mesh(freq,1:N,10*log10(abs(F)))
view(0,90), xlim([0 range/60+50])
colorbar
title(['Accel ',num2str(accn),' Time Spectral map over Bayesian with BS = ',num2str(BS),' - ',num2str(xl*100),'% overlap'])
xlabel('Frequency (Hz)')
ylabel('Overlapping blocks')
zlabel('Amplitude (V)')
hold on
plot3(w/60*order,(1:N),zcord,'white')
hold off

figure
plot(wcentralTime,w)
grid
title(['Bayesian RPM estimate Accel ',num2str(accn),' BS = ',num2str(BS),' - ',num2str(xl*100),'% overlap'])
xlabel('Time (sec)')
ylabel('Instantaneous RPM (RPM)')

reprocess = menu('Do you wish to reprocess data?','Yes','No');
if reprocess == 1
    restart3 = 1;
    while restart3 == 1

```

```

option = menu('Choose an option','Reprocess from one point to the
beginning','Reprocess from one point to the end','Reprocess over a range');
if option == 1 || option == 2
    [Xreprocess, Yreprocess] = ginput(1);
    [dump, block1] = min(abs(Xreprocess - wcentralTime));
    if option == 1
        block2 = 1;
        rerange = block1:-1:block2;
    else
        block2 = N;
        rerange = block1:block2;
    end
end
if option == 3
    [Xreprocess, Yreprocess] = ginput(2);
    [dump1, block1] = min(abs(min(Xreprocess) - wcentralTime));
    [dump, block2] = min(abs(max(Xreprocess) - wcentralTime));
    rerange = block1:block2;
end
assignprior = menu('Assign uniform prior to first two estimates?',
'Yes','No');
rangereset = menu('Do you wish to reset RPM range for this reprocessing
iteration?','Yes','No');
if rangereset == 1
    range1 = input('give the lower limit of range in RPM\n')/60*order;
    range2 = input('give the upper limit of range in RPM\n')/60*order;
    [dump2,In1] = min(abs(f-range1)); [dump3,In2] = min(abs(f-range2));
else In1 = 1;In2 = size(f,2);
end
sidepoints = input('How many points on either side of estimate to
consider for analysis?\n');
if assignprior == 2
    if option == 1
        f0 = w(block1+2)/60*order; f1 = w(block1+1)/60*order;
    end
    if option == 2 || option == 3
        f0 = w(block1-2)/60*order; f1 = w(block1-1)/60*order;
    end
end
if assignprior == 1
    f0 = 0; f1 = 0;
    if option == 1
        rerange1 = block1:-1:block1-1;
        rerange = block1-2:-1:block2;
    end
    if option == 2 || option == 3
        rerange1 = block1:block1+1;
        rerange = block1+2:block2;
    end
    for l = (rerange1)
        R_w = x(1,:)*(cos(2*pi*transpose(t)*(f(In1:In2)))));I_w =
x(1,:)*(sin(2*pi*transpose(t)*(f(In1:In2)))));
        C_w = (1/BS)*(R_w.^2+I_w.^2);
        C = (1-((2*C_w)/Nd2(1))).^(1-BS/20);
        A = C;
        I = find(A == max(A));
        if size(I,2) ~= 1
            I = I(1);
        end
        index(1) = I;
        f0 = f1; f1 = f(index(1));
    end
end

```

```

    end
    indexnew = index(1)-sidepoints:index(1)+sidepoints;
    if find(indexnew == In1), indexnew = In1:indexnew(end);
    elseif find(indexnew == In2), indexnew = indexnew(1):In2;
    end
    fnew = f(indexnew);
    w(rerangel) = f(index(rerangel))*60/order;
end
var = input('Give the variance\n');
if assignprior == 2
    maxindex = find(Prior(rerange(1),:) == max(Prior(rerange(1),:)));
    if size(maxindex,2) ~= 1, maxindex = maxindex(1);end
    indexnew = maxindex-sidepoints:maxindex+sidepoints;
    if find(indexnew == In1), indexnew = In1:indexnew(end);
    elseif find(indexnew == In2), indexnew = indexnew(1):In2;
    end
    fnew = f(indexnew);
end
for l = (rerange)
    R_w = x(1,:)*(cos(2*pi*transpose(t)*(fnew))); I_w =
x(1,:)*(sin(2*pi*transpose(t)*(fnew)));
    C_w = (1/BS)*(R_w.^2+I_w.^2);
    C = (1-((2*C_w)/Nd2(l))).^(1-Bs/20);
    Prior(l,:) = 1/sqrt(2*var*pi)*exp(-(f-2*f1+f0).^2/2*var);
    Priortemp = Prior(l,indexnew);
    A = Priortemp.*C/sum(Priortemp.*C);
    I = find(A == max(A));
    if size(I,2) ~= 1
        I = I(1);
    end
    index(1) = I+indexnew(1)-1;
    f0 = f1; f1 = f(index(1));
    indexnew = index(1)-sidepoints:index(1)+sidepoints;
    if find(indexnew == In1), indexnew = In1:indexnew(end);
    elseif find(indexnew == In2), indexnew = indexnew(1):In2;
    end
    fnew = f(indexnew);
end
w(rerange) = f(index(rerange))*60/order;
% Plots - Bayesian estimate
figure
mesh(freq,1:N,10*log10(abs(F)))
view(0,90), xlim([0 range/60+50])
colorbar
title(['Accel ',num2str(accn),' Time Spectral map over Bayesian with BS
= ',num2str(BS),' - ',num2str(x1*100),'% overlap'])
xlabel('Frequency (Hz)')
ylabel('Overlapping blocks')
zlabel('Amplitude (V)')
hold on
plot3(w/60*order,(1:N),zcord,'white')
hold off

figure
plot(wcentralTime,w)
grid
title(['Bayesian RPM estimate Accel ',num2str(accn),' BS =
',num2str(BS),' - ',num2str(x1*100),'% overlap'])
xlabel('Time (sec)')
ylabel('Instantaneous RPM (RPM)')

```

```

interpol = menu('Do you wish to spline interpolate?', 'Yes', 'No');
if interpol == 1
    chooseregion = menu('Choose a big zone around the region to
eliminate', 'Choose');
    [Xinterpol, Yinterpol] = ginput(2);
    [dump5, interpol1] = min(abs(min(Xinterpol)-wcentralTime));
    [dump6, interpol2] = min(abs(max(Xinterpol)-wcentralTime));
    eliminateregion = menu('Choose a region to eliminate', 'Choose');
    [Xelim, Yelim] = ginput(2);
    [dump7, elim1] = min(abs(min(Xelim)-wcentralTime));
    [dump8, elim2] = min(abs(max(Xelim)-wcentralTime));
    preelim = [interpol1:elim1 elim2:interpol2]; postelim =
interpol1:interpol2;
    w(interpol1:interpol2) = spline(preelim,w(preelim),postelim);
    % Plots - Bayesian estimate
    figure
    mesh(freq,1:N,10*log10(abs(F)))
    view(0,90), xlim([0 range/60+50])
    colorbar
    title(['Accel ', num2str(accn), ' Time Spectral map over Bayesian
with BS = ', num2str(BS), ' - ', num2str(x1*100), '% overlap'])
    xlabel('Frequency (Hz)')
    ylabel('Overlapping blocks')
    zlabel('Amplitude (V)')
    hold on
    plot3(w/60*order, (1:N), zcord, 'white')
    hold off

    figure
    plot(wcentralTime,w)
    grid
    title(['Bayesian RPM estimate Accel ', num2str(accn), ' BS =
', num2str(BS), ' - ', num2str(x1*100), '% overlap'])
    xlabel('Time (sec)')
    ylabel('Instantaneous RPM (RPM)')
end
restart3 = menu('Process further?', 'Yes', 'No');
end
end

store = menu('Do you want to store this estimate for', 'Averaging', 'Future
reference', 'Do not store');
if store == 1
    if Ave == 1
        wAve(Ave,:) = w;
        wAvecentral = wcentralTime;
        Ave = Ave+1;
    else wAve(Ave,:) = spline(wcentralTime, w,wAvecentral);
        Ave = Ave+1;
    end
elseif store == 2
    if Store == 1
        wStore(Store,:) = w;
        wStorecentral = wcentralTime;
        Store = Store+1;
    else wStore(Store,:) = spline(wcentralTime, w,wStorecentral);
        Store = Store+1;
    end
end
end

```

```

restart2 = menu('Do you want to restart process?','Yes','No');
if restart2 ==1
    close all, clear w;
end
end
if Ave>1
compareiterations = menu('Do you wish to compare iterations?','Yes','No');
if compareiterations == 1
    figure
    plot(wAvecentral,wAve)
    grid
    title('Comparison of Bayesian estimates from different sources')
    xlabel('Time (sec)')
    ylabel('Instantaneous RPM (RPM)')
end
ave = menu('Do you wish to average Bayesian estimates from different
sources?','Yes','No');
if ave == 1
    waverage = sum(wAve,1)/(Ave-1);
    figure
    plot(wAvecentral,waverage)
    grid
    title('Average of Bayesian estimates from different sources')
    xlabel('Time (sec)')
    ylabel('Instantaneous RPM (RPM)')
end
end

%% Comparison with Bayesian & Tach RPM assignment for RPM spectral map
if splinefit == 2, Time = Timeaveraged; irpm = irpmaveraged; end
[xTachaligns, xindex] = min(abs(deltaT*(1:size(accel,2))-Time(1)));
if xindex < BS
    Nfalls = 1;
else Nfalls = ceil(((xindex-BE)/((1-x1)*BS))+1);
end
[xTachends, xindex2] = min(abs(deltaT*(1:size(accel,2))-Time(end)));
if xindex2 < BS
    Nends = 1;
elseif xindex2 > BS+(N-1)*BS*(1-x1), Nends = N;
else Nends = ceil(((xindex2-BE)/((1-x1)*BS))+1);
end
irpm4temp = spline(Time,irpm,deltaT*(xindex:xindex2));
irpm4(1:xindex-(Nfalls-1)*BS*(1-x1)-1) = 0; irpm4(xindex-(Nfalls-1)*BS*(1-
x1):xindex-(Nfalls-1)*BS*(1-x1)-1+size(irpm4temp,2)) = irpm4temp;
M = zeros(1,Nends-Nfalls+1);
for k = 1:Nends-Nfalls
    if (Nfalls+(k-1)-1)*BS*(1-x1)+1 < xindex
        M(k) = sum(irpm4(BS*(1-x1)*(k-1)+1:BS+(k-1)*BS*(1-x1)))/(BS-
(xindex-(Nfalls+(k-1)-1)*BS*(1-x1)-1));
    else
        M(k) = mean(irpm4(BS*(1-x1)*(k-1)+1:BS+(k-1)*BS*(1-x1)));
    end
end
M(k+1) = mean(irpm4(BS*(1-x1)*(k)+1:end));
wTimediff = wcentralTime(Nfalls:Nends);
wcompare = w(Nfalls:Nends);
RPMdifference = wcompare - M;

%% Order analysis
% Order map
clear TVDFT IRPM F2 F3 I

```



```

Useirpm = menu('Use which RPM estimate for order
analysis?', 'Bayesian', 'Adaptive nth pulse');
if Useirpm == 1
    IRPM = w;
    Ndash = N; Nvec = 1:N; kdashed = 1;
else IRPM = M;
    Ndash = Nends-Nfalls+1; Nvec = Nfalls:Nends; kdashed = Nfalls;
end

IRPMdash = (1:floor(freq(end)/max(M/60))).'* IRPM;
figure
mesh(freq, IRPM, 10*log10(abs(F)))
view(0, 90)
colorbar
title(['RPM Spectral map with BS = ', num2str(BS), ' - ', num2str(x1*100), '%
overlap'])
xlabel('Frequency (Hz)')
ylabel('Instantaneous speed (RPM)')
zlabel('Amplitude (V)')
hold on
plot3(IRPMdash/60, IRPM, zcord, 'black')
hold off

orders = input('Which orders to track?(Give multiple orders in a row
vector)\n');
freq1 = zeros(Ndash, BS/2+1);
for k = 1:Ndash
    freq1(k, :) = freq/(IRPM(k)/60);
end
figure
mesh(freq1, IRPM, 10*log10(abs(F(Nvec, :))))
view(0, 90)
colorbar
title(['Accel ', num2str(accn), ' Order RPM Spectral map with BS =
', num2str(BS), ' - ', num2str(x1*100), '% overlap'])
xlabel('Orders'), xlim([0 floor(freq(end)/(max(IRPM)/60))+0.5])
ylabel('Instantaneous speed (RPM)')
zlabel('Amplitude (V)')
ordertracking = menu('Choose method for order tracking', 'Order cut (Freq
domain)', 'TVDFFT (Time domain)', 'Both');
if ordertracking == 1 || ordertracking == 3
    % Order cut
    I = zeros(1, Ndash); F3 = zeros(size(orders, 2), Ndash);
    for j = 1:size(orders, 2)
        kdash = kdashed;
        for k = 1:Ndash
            [discard, I] = min(abs(freq1(k, :) - orders(j)));
            F3(j, k) = F(kdash, I);
            kdash = kdash+1;
        end
    end
    F3sum = sum(abs(F3));
    figure
    semilogy(IRPM/60, (abs(F3)), IRPM, F3sum, 'black')
    grid
    legend(num2str(orders.'))
    title(['Accel ', num2str(accn), ' Order Track: Orders ', num2str(orders), '
with BS = ', num2str(BS), ' - ', num2str(x1*100), '% overlap'])
    xlabel('Frequency (Hz)')
    ylabel('Amplitude (V)')

```

```

end
if ordertracking == 2 || ordertracking == 3
    % TVDFT
    TVDFT = zeros(1,Ndash);
    for j = 1:size(orders,2)
        kdash = kdashed;
        for k = 1:Ndash
            An = (accel(BS*(1-x1)*(kdash-1)+1:BS+(kdash-1)*BS*(1-
x1))).*cos(2*pi*orders(j)*IRPM(k)/60*t);
            Bn = (accel(BS*(1-x1)*(kdash-1)+1:BS+(kdash-1)*BS*(1-
x1))).*sin(2*pi*orders(j)*IRPM(k)/60*t);
            An = sum(An)/BS; Bn = sum(Bn)/BS;
            TVDFT(j,k) = An + 1j*Bn;
            kdash = kdash+1;
        end
    end
    end
    Tvsum = sum(abs(TVDFT));
    figure
    semilogy(IRPM/60, (abs(TVDFT)), IRPM, Tvsum, 'black')
    grid
    legend(num2str(orders.'))
    title(['Accel ', num2str(accn), ' TVDFT Order Track: Orders
', num2str(orders), ' with BS = ', num2str(BS), ' - ', num2str(x1*100), '%
overlap'])
    xlabel('Frequency (Hz)')
    ylabel('Amplitude (V)')
end

%% Plots
close all
% 1 Time Spectral Map
figure
mesh(freq, 1:N, 10*log10(abs(F)))
view(0,90), xlim([0 range/60+50])
colorbar
title(['Accel ', num2str(accn), ' Bayesian on Time Spectral Map with BS =
', num2str(BS), ' - ', num2str(x1*100), '% Overlap'])
xlabel('Frequency (Hz)')
ylabel('Overlapping blocks')
zlabel('Amplitude (dB)')
hold on
plot3(w/60*order, (1:N), zcord, 'white')
hold off
% 2 Raw Tach RPM estimate
figure
plot(rawTime, rawirpm)
grid
title('Raw RPM estimate from Tach Signal')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 3 Comparison of Processed Tach and Raw Tach estimate
figure
plot(rawTime, rawirpm, 'green', Time, irpm, 'red')
grid, legend('Raw Tach', 'Processed Tach')
title('Comparison of Raw Tach estimate vs. Processed Tach')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 4 Comparison of Processed Tach and Assigned Tach
figure
plot(Time, irpm, 'blue', wTimediff, M, 'red')
grid, legend('Processed Tach', 'Assigned Tach')

```

```

title('Comparison of Processed Tach vs. Assigned Tach')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 5 Bayesian estimate
figure
plot(wcentralTime,w)
grid
title(['Bayesian RPM Estimate Accel ',num2str(accn),' BS = ',num2str(BS),'
- ',num2str(x1*100),'% Overlap'])
xlabel('Time (sec)')
ylabel('Instantaneous RPM (RPM)')
% 6 Comparison of Bayesian estimate and Raw Tach estimate
figure
plot(rawTime,rawirpm,'green',wcentralTime,w,'red')
grid, legend('Raw Tach','Bayesian')
title('Comparison of Raw Tach vs. Bayesian Estimate')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 7 Comparison of Bayesian estimate and processed Tach
figure
plot(Time,irpm,'blue',wcentralTime,w,'red')
grid, legend('Processed Tach','Bayesian')
title('Comparison of Processed Tach vs. Bayesian Estimate')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 8 Comparison of Bayesian estimate and assigned Tach
figure
plot(wTimediff,M,'blue',wcentralTime,w,'red')
grid, legend('Assigned Tach','Bayesian')
title('Comparison of Assigned Tach vs. Bayesian Estimate')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 9 Difference between Bayesian and Tach estimates
figure
plot(wTimediff,RPMdifference)
grid
title('Difference between Processed Tach Estimate and Bayesian Estimate')
xlabel('Time(sec)')
ylabel('Instantaneous speed estimate (RPM)')
% 10 Percentage difference between Bayesian and Tach RPM
figure
plot(wTimediff,RPMdifference./M*100)
grid
title('Percentage Difference between Processed Tach Estimate and Bayesian
Estimate')
xlabel('Time(sec)')
ylabel('Percentage difference in RPM (%)')
% 11 RPM Spectral Map with Bayesian estimate
figure
mesh(freq,w,10*log10(abs(F)))
view(0,90)
colorbar
title(['Accel ',num2str(accn),' RPM Spectral Map (Bayesian) with BS =
',num2str(BS),' - ',num2str(x1*100),'% Overlap'])
xlabel('Frequency (Hz)')
ylabel('Instantaneous speed (RPM)')
zlabel('Amplitude (V)')
% 12 RPM Spectral Map with Processed Tach estimate
figure
mesh(freq,M,10*log10(abs(F(Nfalls:Nends,:))))
view(0,90)

```

```

colorbar
title(['Accel ',num2str(accn),' RPM Spectral Map (Processed Tach) with BS = ',num2str(BS),' - ',num2str(xl*100),'% Overlap'])
xlabel('Frequency (Hz)')
ylabel('Instantaneous speed (RPM)')
zlabel('Amplitude (V)')
% 13 Comparison of Bayesian from different sources
if Ave > 1
if compareiterations == 1
    figure
    plot(wAvecentral,wAve)
    grid
    title('Comparison of Bayesian Estimates from Different Sources')
    xlabel('Time (sec)')
    ylabel('Instantaneous RPM (RPM)')
end
end

```