Arithmetic Compression of Images

S.L. Lahudkar*, R.K. Prasad**

Abstract

The aim of image compression is to reduce the amount of data needed to accurately represent an image such that this image can be economically transmitted and received. Currently, standards like JPEG, MPEG etc. are being employed for compression of images. One of the recent techniques being studied and developed is the arithmetic compression of images. In comparison to the well known Huffman coding algorithm, which is used by most of the existing standards, arithmetic compression overcomes the constraint that a symbol has to be encoded by a whole number of bits. Also, the compression ratio is better than that of Huffman. Huffman coding has another disadvantage that it requires the probabilities of the symbols in powers of 2.In this paper we have mentioned the various problems one can come across while implementing this algorithm as well as given solutions for the same. We have implemented the above algorithm to monochrome images. The results have been shown for a few standard monochrome images. This technique can be efficiently applied to gray images and it can be extended to color images also.

Keywords: Compression ratio, Lossless, Lossy, Arithmetic, Symbol, Encoder, Decoder, Entropy, Scaling.

* Asst. Prof, Deptt. of Electronics & Telecommunication, Imperial College of Engineering & Research, Pune. email: swapnil_lahudkar@rediffmail.com / swapnillahudkar@gmail.com

* *Principal, K.J. Somaiyya College of Engineering, Mumbai

1. Introduction

Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of reduction process is the removal of redundant data. Currently, image compression is recognized as an enabling technology. With image sensors of high spatial resolutions that are being used today, the image sizes can extend to several megabytes. Inspite of constraints on the bandwidth and channel capacity specified by Shannon's limit, compression permits transmission and archieving of such large image files.

Taking into consideration the relevance of compression and the phenomenal changes in technology in this direction, we have chosen to explore one such coming up technique called "arithmetic compression".

1.1 Lossless Compression

In certain applications like archieval of medical or business documents, satellite image processing, digital radiography etc. error free compression is of prime importance. This technique is composed of two independent operations:

a) Devising an alternative representation of the image.

b) Coding the representation.

The coding methods that are loss-less in nature include:

i) Variable length coding:

a) Huffman

b) Arithmetic

ii) LZW coding

1.2 Lossy Compression

Lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression. If the resulting distortion can be tolerated, the compression ratio can be significant. Such techniques can be applied to images on the internet, in video compression and multimedia applications. The compression methods that are lossy in nature include:

i) Lossy predictive coding

ii) Transform coding:

a) Discrete Cosine Transform (DCT)

b) Discrete Fourier Transform (DFT)

c) Walsh Hadamard Transform

iii) Wavelet Coding

2. Arithmetic Coding

Arithmetic coding overcomes the constraint that the symbol has to be coded by a whole number of bits. This leads to higher efficiency and a better compression ratio in general. Indeed Arithmetic Coding can be proven to almost reach the best compression ratio possible, which is bounded by the entropy of the data being encoded. During encoding the algorithm generates one code for the whole stream, this is done fully sequential manner, symbol after symbol.Arithmetic coding is a very efficient principle for Lossless data encoding, which satisfies all the requirements of what people understand of a modern compression algorithm. Finite precision integer arithmetic suffices for all calculations. These and other properties make it straightforward to derive hardware-based solutions. The decoder uses almost the same source code as the encoder, which also makes the implementation straightforward.

2.1 Coding of images using Arithmetic Compression

Our goal is to compress data, which might either be stored on a computer readable media or be sent over some form of stream. This data could represent anything, reaching from simple text upto graphics, binary executable programs etc. However, we do not distinguish here between all those data types. We simply see them all as binary input. A group of such input bits is what we will refer to as *symbol*. When looking at a message sequence, one can calculate a distinct probability of each symbol to occur in this sequence. We can directly conclude that the probability of every symbol is always contained in the interval [0,1) for any symbol, whereas the sum over all such probabilities is always 1. This interval is open-ended, because it would make no sense to encode a constant sequence holding only a symbol of probability 1, simply because in that case the full content of the sequence would have been known beforehand already.

2.2 Encoder and Decoder

An algorithm which encodes the sequence is called an ENCODER. The appropriate algorithm decoding the sequence again is called a DECODER. An encoder could be any algorithm transforming the input in such a way that there is a decoder to reproduce the raw input data. We are only going to consider codes that are able to reproduce the input data up to the last symbol.

2.2.1 Encoding to Real Numbers

All probabilities of the symbols fall into the range [0,1) while their sum equals 1 in every case. This interval contains infinite amount of real numbers, so it is possible to encode every possible sequence to a number in [0,1). One partitions the interval according to the probability of the symbols. By iterating this step for each symbol in the message, one refines the interval to a unique result that represents the message. Any number in this interval would be the valid code.

2.2.2 Upper and Lower bounds

Henceforth we call the upper and lower bounds of the entire current interval *high* and *low*. The

bounds of the sub intervals are calculated from the cumulative probabilities:

$$K(a_k) = \sum_{i=1}^{k} P_M(a_i) .$$

The values of *high* and *low* change during the encoding process whereas the cumulative probabilities remain constant.

2.2.3 Encoding

The first step in encoding is the initialization of the interval I = (*low*, *high*) by *low* = 0 and *high* = 1. When the first symbol, say $s1$ is read, the interval I can be resized to a new interval I' according to the symbol. The boundaries of I' are also called *low* and *high*. We choose I' to equal the boundaries of $s1$ in the model. However, how are these boundaries calculated? Let $s1 = a_k$ be the k^{th} symbol of the alphabet. Then the lower bound is

$$low := \sum_{i=1}^{k-1} P_M(a_i) = K(a_{k-1})$$

The new interval I' is set to (*low*, *high*). The most relevant aspect of this method is that the sub-interval I' becomes larger for more probable symbols $s1$. The larger the interval the lower the number of fractional places which results in shorter code words. All following numbers generated by the next iterations will be located in the interval I' since we use it as base interval as we did used [0,1) before. We proceed with the second symbol $s2$. However, now we have the problem that our model M describes a partition 7 of the interval [0,1), not of I' which was calculated in the previous step. We have to scale and shift the boundaries to match the new interval. Scaling is accomplished by a multiplication with *high-low*, the length of the interval. Shifting is performed by adding *low*. This results in the equation

$$low' := low + \sum_{i=1}^{j-1} P_M(a_i)\cdot(high - low) = low + K(a_{j-1})\cdot(high - low); \quad (3)$$

$$high' := low + \sum_{i=1}^{j} P_M(a_i)\cdot(high - low) = low + K(a_j)\cdot(high - low). \quad (4)$$

2.2.4 Decoding

To decode a sequence, one somewhat have to apply the encoder backwards. The value V = *Code* (S) is given and we have to restore the original sequence. We assume that the message length is known and equals l. In the first iteration we compare V with each interval $I = [K(a_k - 1), K(a_k))$ to find the one that contains V. It corresponds to the first symbol of the sequence, $s1$. To compute the next symbol, we have to modify the probability partition in the same way we did while encoding:

$$low' := low + K(a_{i-1})\cdot(high - low)$$
$$high' := low + K(a_i)\cdot(high - low),$$

where i has to comply

$$low \le V \le high$$

Here, a_i is the next symbol in the encoded sequence. This time, the start case is again a special case of the general formula. The iteration is very similar to the encoder, so from its implementation should arise no further problems.

2.2.5 Encoding As a Sequence of bits

To implement arithmetic coding effectively, we have to make certain restrictions. There are infinite real numbers, so the pure integer implementations are way faster on simple processors as found in fax machines (which actually use arithmetic coding in the G3 protocol). The output when encoded as a sequence of bits is a non-ambiguous sequence of bits that can be stored or transmitted.

2.2.5.1 Encoding

The encoder consists of a function and static variables that store the current state:

- **mLow** stores the current lower bound. It is initialized with 0.
- **mHigh** stores the current upper bound. It is initialized with 0x7FFFFFFF, the maximum value that fits in 31 bits.
- **mStep** stores a step size that is introduced later. It is not necessarily static in the encoder, but the decoder depends on this property.

2.2.5.2 Decoding

The task of the decoder is to follow the steps of the encoder one by one. Hence we have to determine the first symbol and update the bounds accordingly. This divides the decoder functionality into two functions:

First determines the interval that contains the symbol. This is accomplished by

calculating the code value of the symbol:

```
mStep = ( mHigh - mLow + 1 ) / total;

value = ( mBuffer - mLow ) / mStep;
```

- **mBuffer** variable that contains the encoded sequence.

The model can use the return value to determine the encoded symbol by comparing it to the cumulative count intervals. As soon as the proper interval is found, the boundaries can be updated like they were during encoding:

```
mHigh = mLow + mStep * high_count - 1;

mLow = mLow + mStep * low_count;
```

3. Scaling in Limited Ranges

When we use the presented methods to encode several symbols, a new problem arises: **mLow** and **mHigh** converge more and more and so further encoding will be impossible as soon as the two values coincide.

3.1 E1 and E2 Scaling

As soon as **mLow** and **mHigh** lie in the same half of the range of numbers , it is guaranteed that they will never leave this range again since the following symbols will shrink the interval. Therefore the information about the half is irrelevant for the following steps and we can already store it and remove it from consideration. Given the presented implementation, if hex values are used the most significant bits (MSB) of **mLow** and **mHigh** are equal in this case. 0 corresponds to the lower half while 1 represents the upper. As soon as the MSBs match, we can store them in the output sequence and shift them out. This is called *E*1 - respective *E*2-scaling.

3.2 E3 Scaling

Though E1 and E2 scaling are a step in the right direction, they are not sufficient on their own. They won't work when **mLow** and **mHigh** converge to the center of the interval: Both stay in their halves, but the interval soon becomes too small. They differ from each other, but further encoding is impossible. This is where E3 scaling comes into play: As soon as **mLow** leaves the lowest quarter (maximum value of the first quarter: **g_FirstQuarter**) and **mHigh** the highest (fourth) quarter (maximum value of the third quarter: **g_ThirdQuarter**), the total range is less than half of the original range and it is guaranteed that this won't change because of the ongoing shrinking. It is not immediately determinable which half will contain the result, but as soon as the next E1 or E2 scaling is possible, one knows the values that one could have stored earlier if one were able to foresee

this. This might sound strange, but it's the way E3 scaling works.

4. Results

Given below are some of the results obtained by applying arithmetic compression technique to certain monochrome images. Similarly this technique can also be implemented to color images.

FLOWERS.BMP

ORIGINAL IMAGE RECOVERED IMAGE

ENTROPY = 0.709 BPS
COMPRESSION RATIO = 0.709 BPP

BLOOD.BMP

ORIGINAL IMAGE RECOVERED IMAGE

ENTROPY = 0.887 BPS
COMPRESSION RATIO= 0.887 BPP

MONA.BMP

ORIGINAL IMAGE RECOVERED IMAGE

ENTROPY = 0.910 BPS
COMPRESSION RATIO = 0.910 BPP

5. Applications

- **Medical Images**: Normally medical images like X-Rays, CT scans etc. have a region of interest and a boundary region. The region of interest contains critical information and is therefore coded using a lossless technique like Arithmetic Coding, while the boundary region can be coded using a lossy technique

  Fax Protocols: Nowadays there are a lot of hidden applications of Arithmetic Coding, such as hardware-based codecs as for instance the fax protocols G3 and G4. This kind of application make Arithmetic coding maximally efficient by the use of a small alpha bet with an unevenly distributed probability.

- **Multimedia:** In order to compress a large number of pictures and other data, which require a lossless compression like this, is used.

- **Printing and Scanning:** A huge data is to be stored everyday for newspapers and hence requires an efficient compression algorithm.

6. Conclusion

With Arithmetic Coding, we have described a coding method, which is suitable for lossless data compression. We have seen, that Arithmetic Coding can work sequentially, encoding symbol per symbol and thus is able to send already encoded parts of a message before it is fully known.

Arithmetic Coding has a better efficiency than Huffman Coding. Also, Huffman coding is optimum only among the coding schemes which assign a fixed integer number of bits to each symbol. We know for sure that the Shannon theorem guarantees that compression below the entropy of the source is impossible. Indeed Arithmetic Coding can be proven to almost reach the best compression ratio possible, which is bounded by the entropy of the data being encoded. The technique can be applied to monochrome and gray images satisfactorily. Color images also can be coded.

References

1) Alexandre Krivolulhefts, "A Method for Progressive Near Lossless Compression", Proceedings of IEEE International Conference on Image Processing, pp 185-188, 2003.

2) K.Chen and T.Ramabadaran, "Near Lossless Compression of Medical Image Through Entropy Coded DPCM", IEEE Transactions on Medical Imaging,vol.13,no.9,pp 538-548, Sept.1994.

3) Gaurav Sharma, Utlen Celik and A. Murut Tekalp, "Gray Level Embedded lossless Image Compression".

4) I.Avcibas, N.Memon, B.Sankur and K. Sayood, *" A Progressive Lossless / Near Loss Less Image Compression Algorithm"*, IEEE signal processing letters 9(10), pp 312-314, 2002.

5) Kivijarvi J., "*A Comparison Of Loss-Less Compression Methods for Medical Images*", Computerized Medical Imaging and Graphics, pp 323-329.

6) R.C. Gonzalez, R.E. Woods and S.L. Eddins, "*Digital Image Processing*", Pearson Education, pp 282-284.

7) B.Chanda and D.D.Majumdar, *"Digital Image Processing and Analysis"*, PHI, pp 147-160.

Websites

- http://links.uwaterloo.ca/

Theoretical Investigation into Effect of Mesh Stiffness and Bearing Stiffness on Modal Frequencies of Geared Shaft

S. B. Wadkar*, Dr. S. R. Kajale**, T. R. Trivedi***

Abstract

The variation in stiffness of meshing teeth, as the number of gear teeth pair in contact changes, causes instabilities and vibrations in geared systems. It is a well established fact that bearing properties have a strong influence on shaft dynamics. However, a review of literature reveals the effect of bearing properties on modal frequencies is not sufficiently investigated. Most of the researchers presented gear dynamics models restricting to torsional deflections only, however they did not report the investigations considering lateral deflections. The above investigations may not be valid for a typical overhung type geared system rotating at a high speed. This paper presents a geared shaft system as an equivalent rotor system connected with spring. The equation of motion is written, and natural frequencies are computed by developing a computer program in MATLAB. The mesh force over a range of frequencies is estimated for free and forced vibrations considering torsional only and torsional plus lateral vibrations. Change in mesh force variation is observed if lateral vibration is included in the analysis. There is a variation of modal natural frequencies as the mesh stiffness and bearing stiffness changes. The results of investigation show that there is a significant change in mesh force owing to the introduction of lateral vibrations.

Key Words: Mesh stiffness, Natural frequencies, bearing stiffness, torsional vibrations.

* Prof. & Head Dept. of mechanical Engg., Bharati vidyapeeth university college of engineering, Pune

** Director SGGS, Nanded,

*** Post graduate student, Dept. of mechanical Engg., Bharati vidyapeeth university college of engineering, Pune

1. Introduction

Transmission system is a complex structure consisting of driver and many other components: shafts, gears, couplings and load. Commonly, gears are used in power transmission systems. The primary source of gear vibration and noise is the dynamic excitation owing to changing stiffness of the meshing teeth. The mesh stiffness associated with elastic tooth varies as the number of teeth in contact changes. Determination of these unstable operating conditions and identification of design parameters that minimise their occurrence are crucial to the design of a quieter gear systems. Many times in gear dynamics analysis, the effect of the lateral vibration is not taken in modeling the system assuming that the effect of the lateral deflection in a typical gear system is negligible as compared to the torsional effect. This assumption is not true in case of many configurations such as that of an overhung type gears rotating at high speed, in which the whirling resonance speed occurs within the operating frequency range.

H Vinayak and R Singh, used multi-body dynamics modelling strategy for rigid gears to include compliant gear bodies in multi-mesh transmissions [1]. The natural frequency and

vibration mode sensitivities to system parameters were studied by J Lin and R. G. Parker [2]. Non-Linear dynamic response of a spur gear pair (finite element modeling and experimental result comparison) was studied by R. G. Parker, S. M. Vijyakar, T. Imajo, [3]. Jian Lin and R. G. Parker worked on the parametric resonance in two-stage gears and concluded that the change in mesh stiffness causes parametric instabilities and severe vibrations in geared systems [4].Vibration and noise is a major consideration in the design of high performance geare transmissions, requiring smooth and quieter operation of machinery [5]. Shengxiang Jia, Ian Howard, and Jiande Wang, studied the dynamic modeling of multiple pairs of spur gears in mesh, including friction and geometrical errors [6].

It can be seen that many researchers have worked on changing mesh stiffness and bearing stiffness of geared system but interaction between torsional and lateral vibration is a neglected area. In this paper development of a computer program to calculate natural frequencies and mesh force is discussed and an attempt is made to establish interactions between the lateral and torsional responses in a simplified manner. Also variation of natural frequency with change in mesh stiffness and bearing stiffness is explained.

2.1 Mathematical Modelling of the System

Consider a linear dynamic system (Fig. 1) consisting of a pinion and gear driven by a motor and load. The self weight of shaft, gyroscopic effect of rotor is ignored in the model.

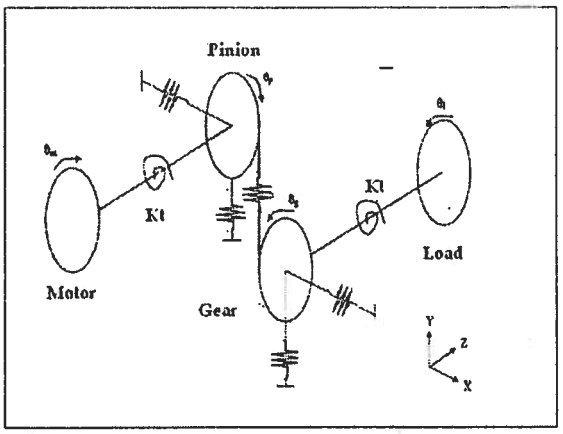


Fig. 1 Mathematical model of Geared shaft system.

The gears are modeled as rigid discs. Contact between pinion and gear is modeled as a spring having linear stiffness only in the tangential direction referred as mesh stiffness hereafter.

m_p, m_g = Mass of pinion and gear.

r_p, r_g = Pitch circle radius of pinion and gear.

U_p, U_g = Mass unbalance in pinion and gear.

N = Inverse of gear ratio.

Θ_p, Θ_g = Angular displacements of pinion and gear.

Θ_m, Θ_l = Angular displacements of motor and load.

K_m = Mesh stiffness.

φ = Pressure angle.

Ω = Rotational speed of pinion.

X_p, X_g = Displacement in Z direction (horizontal).

Y_p, Y_g = Displacement in Y direction (vertical).

T_m, T_l = Motor and load torques.

T_p, T_g = Pinion and gear torque.

The assumptions made in the equation of motion are:

1. Gears are modeled as rigid discs.
2. Axial motion and forces are negligible.
3. Lateral displacements are small.
4. Angular velocity is small.
5. Unbalance is small.
6. The bearing characteristics are linear.
7. Damping effect at bearings and mesh are ignored.

The equation of motion is written using the Lagrange method [7] :

$$
\begin{bmatrix}
m_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & m_g & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & m_p & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & m_g & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & I_p & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & I_g & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & J_m & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & J_l
\end{bmatrix}
\begin{Bmatrix}
\ddot{y}_p \\ \ddot{y}_g \\ \ddot{x}_p \\ \ddot{x}_g \\ \ddot{\theta}_p \\ \ddot{\theta}_g \\ \ddot{\theta}_m \\ \ddot{\theta}_l
\end{Bmatrix} +
$$

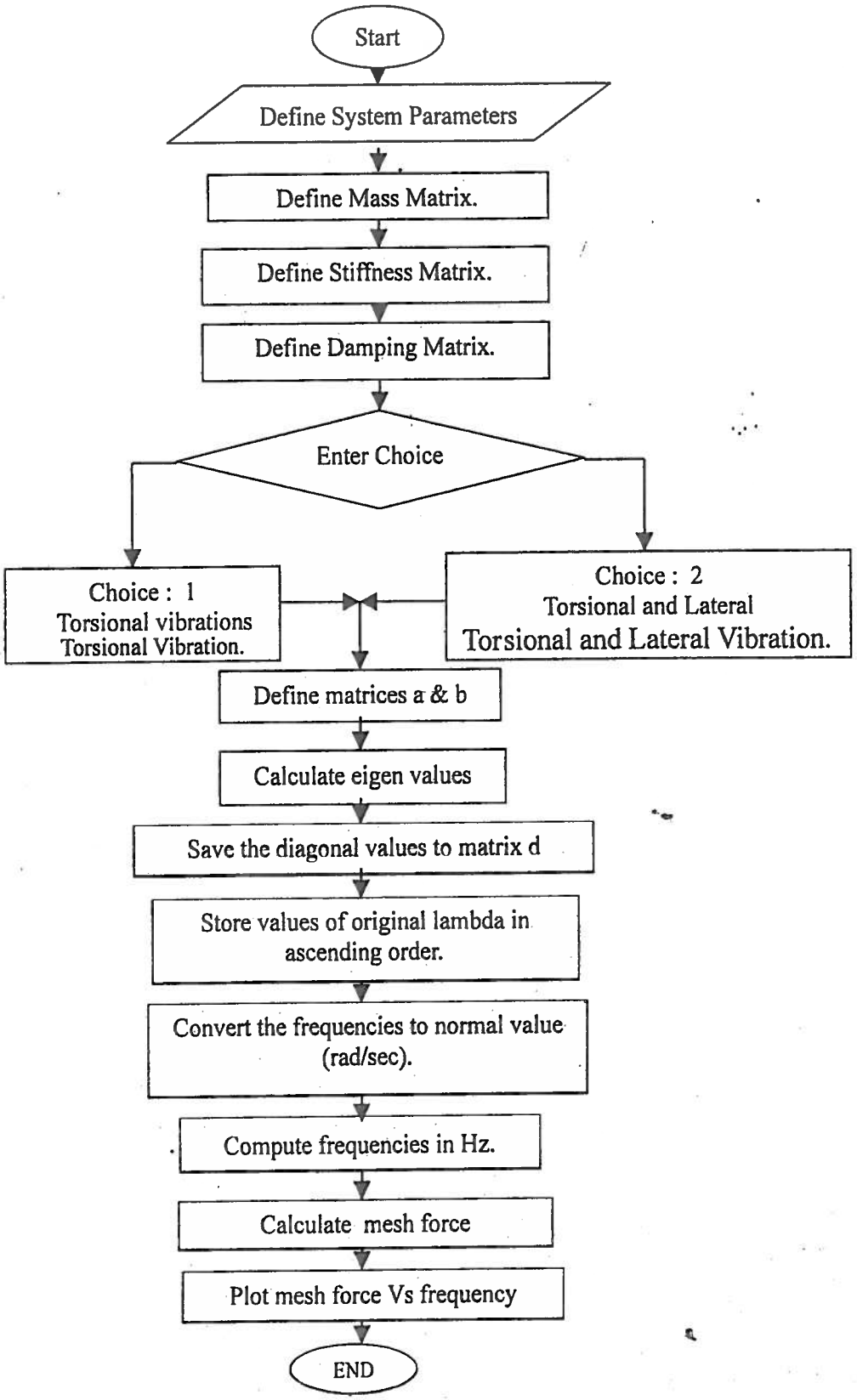
$$
\begin{bmatrix}
K_y + K_m & -K_m & 0 & 0 & r_p K_m & -r_p K_m & 0 & 0 \\
-K_m & K_y + K_m & 0 & 0 & -r_p K_m & r_g K_m & 0 & 0 \\
0 & 0 & K_x & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & K_x & 0 & 0 & 0 & 0 \\
r_p K_m & -r_p K_m & 0 & 0 & r_p^2 K_m + K_l & -r_p r_g K_m & -K_l & 0 \\
-r_p K_m & r_p K_m & 0 & 0 & -r_p r_g K_m & r_g^2 K_m + K_l & 0 & -K_l \\
0 & 0 & 0 & 0 & -K_l & 0 & K_l & 0 \\
0 & 0 & 0 & 0 & 0 & -K_l & 0 & K_l
\end{bmatrix}
\begin{Bmatrix}
y_p \\ y_g \\ x_p \\ x_g \\ \theta_p \\ \theta_g \\ \theta_m \\ \theta_l
\end{Bmatrix}
$$

$$
= \begin{Bmatrix}
U_p \Omega^2 \cos \Omega t \\
U_g N^2 \Omega^2 \cos(N\Omega t + \psi) \\
U_p \Omega^2 \sin \Omega t \\
U_g N^2 \Omega^2 \sin(N\Omega t + \psi) \\
T_p \\ T_g \\ T_m \\ T_l
\end{Bmatrix}
\quad \cdots\cdots (1)
$$

Natural frequencies of the system are obtained by solving the eigen values of the determinant of Eq. 1 considering both free and forced vibration

as well as torsional and torsional plus lateral vibrations. A MATLAB program Graphic User Interface (GUI) is developed. Fig. 2 shows the screen shot of the user interface of the program.

After entering all the data for input fields as shown in Fig. 2, one has to select the choice from the menu for torsional system or torsional and lateral system, the program will display graph of frequency Vs mesh force and all the values of the natural frequencies after pressing the calculate button on the menu screen.



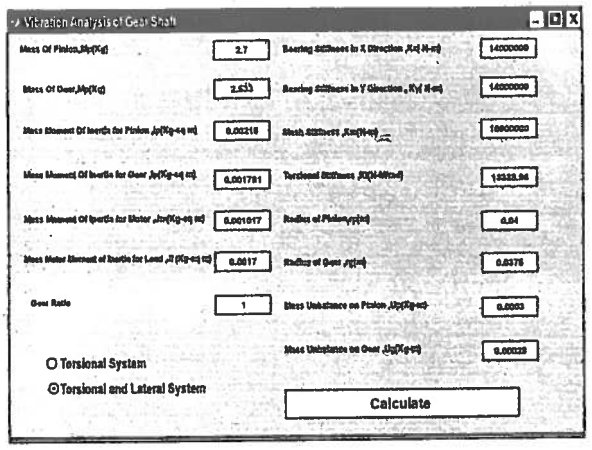


Fig. 2 Screen shot of user interface of program.

2.2 Torsional vibration response

The torsional vibration is obtained using Eq. 1. The value of bearing stiffness assigned is infinity to nullify the effect of the lateral vibrations. Thus the torsional vibrations of a system will be taken into account neglecting lateral vibrations. Therefore the present system has four degrees of freedom as given in Eq. 2.

$$\begin{bmatrix} I_p & 0 & 0 & 0 \\ 0 & I_g & 0 & 0 \\ 0 & 0 & J_m & 0 \\ 0 & 0 & 0 & J_l \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_p \\ \ddot{\theta}_g \\ \ddot{\theta}_m \\ \ddot{\theta}_l \end{Bmatrix} + \begin{bmatrix} r_p^2 K_m + K_t & -r_p r_g K_m & -K_t & 0 \\ -r_p r_g K_m & r_p^2 K_m + K_t & 0 & -K_t \\ -K_t & 0 & K_t & 0 \\ 0 & -K_t & 0 & K_t \end{bmatrix} \begin{Bmatrix} \theta_p \\ \theta_g \\ \theta_m \\ \theta_l \end{Bmatrix} = \begin{Bmatrix} T_p \\ T_g \\ T_m \\ T_l \end{Bmatrix} \quad (2)$$

$$\begin{bmatrix} r_p^2 K_m + K_t - \omega^2 I_p & -r_p r_g K_m & -K_t & 0 \\ -r_p r_g K_m & rg^2 Km + Kt - \omega^2 Ig & 0 & -K_t \\ -K_t & 0 & K_t - \omega^2 J_m & 0 \\ 0 & -K_t & 0 & K_t - \omega^2 J_l \end{bmatrix} \begin{Bmatrix} \Theta_p \\ \Theta_g \\ \Theta_m \\ \Theta_l \end{Bmatrix} = 0 \quad (3)$$

2.3 Combined torsional and lateral vibration response

The response of combined torsional and lateral vibration is computed by using Eq. 4. The combined system has eight degree of freedom, i.e. four in torsional and four in lateral direction.

The various natural frequencies are computed using Eq. 2, Eq. 3 and Eq. 4 using the developed GUI. Mesh force versus frequency response is thus predicted.

$$\begin{bmatrix} K_y + K_m - \omega^2 m_p & -K_m & 0 & 0 & r_p K_m & r_g K_m & 0 & 0 \\ -K_m & K_y + K_m - \omega^2 m_g & 0 & 0 & -r_p K_m & -r_g K_m & 0 & 0 \\ 0 & 0 & K_x - \omega^2 m_p & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_x - \omega^2 m_g & 0 & 0 & 0 & 0 \\ r_p K_m & -r_p K_m & 0 & 0 & r_p^2 K_m + K_t - \omega^2 I_p & -r_p r_g K_m & -K_t & 0 \\ -r_p K_m & r_p K_m & 0 & 0 & -r_p r_g K_m & r_g^2 K_m + K_t - \omega^2 I_g & 0 & -K_t \\ 0 & 0 & 0 & 0 & -K_t & 0 & K_t - \omega^2 J_m & 0 \\ 0 & 0 & 0 & 0 & 0 & -K_t & 0 & K_t - \omega^2 J_l \end{bmatrix} \begin{Bmatrix} Y_p \\ Y_g \\ X_p \\ X_g \\ \Theta_p \\ \Theta_g \\ \Theta_m \\ \Theta_l \end{Bmatrix} = 0 \quad (4)$$

2.4 System parameters

The system parameters used for analysis are given in Table 2.1

Table2.1

Parameter	Value
Mass of pinion, m_p	2.7 kg
Mass of gear, m_g	2.533 kg
Mass Moment of Inertia for pinion, I_p	0.00216 kg-m^2
Mass Moment of Inertia for gear, I_g	0.001781 kg-m^2
Mass Moment of Inertia for motor, J_m	0.001017 kg-m^2
Mass Moment of Inertia for load, J_l	0.017 kg-m^2
Bearing stiffness in x-direction, K_x	1.4 x 10^7 N/m
Bearing stiffness in y-direction, K_y	1.4 x 10^7 N/m
Mesh stiffness, K_m	1 x 10^7 N/m
Torsional stiffness, K_t	13338.96 N-m/rad
Radius of pinion, r_p	0.04 m
Radius of gear, r_g	0.0375 m
Mass unbalance on pinion, U_p	0.0003 kg-m
Mass unbalance on gear, U_g	0.00028 kg-m
Module	2.58 mm
No. of teeth on pinion	29
No. of teeth on gear	27
Pressure Angle (ö)	20 ^{0} (0.349 radians)

3. Discussion and results

Using the GUI program, computed modal frequencies for system parameters are given in Table 2.1. The comparison of response of mesh force Vs frequency for torsional and torsional plus lateral vibration is also presented here.

3.1 Torsional Vibration Analysis

The modal frequencies for the Torsional mode of vibration are estimated by solving the eigen values of the determinant of Eq. 3. The computed modal frequencies for the system parameters from Table2.1 are:

±0, ±381.7476, ±665.5171, ±882.2625 (Hz).

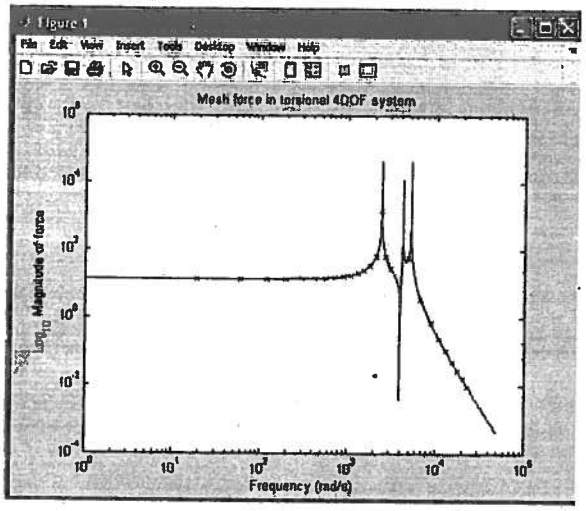


Fig. 3 Mesh force Vs frequency (torsional vibrations only)

3.2 Torsional plus lateral vibration analysis

The modal frequencies for the torsional plus lateral vibration are estimated by solving the eigen values of the determinant of Eq. 4. The modal frequencies for system parameters of Table2.1 are:

±0, ±241.4450, ±362.4117, ± 368.1675, ±374.1679, ±528.2062, ±672.5216, ±897.1014 (Hz).

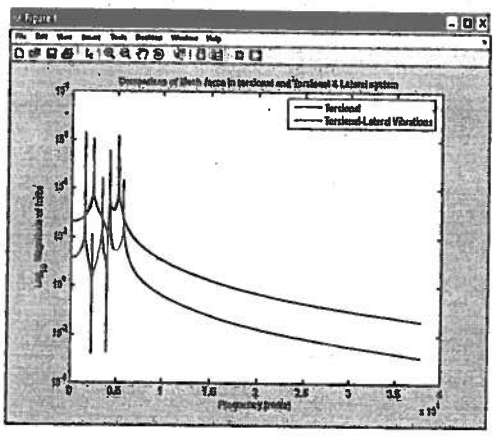


Fig. 4 Comparison of mesh force Vs frequency (Torsional only and torsional plus lateral vibrations)

Comparison of mesh force (Fig. 4) indicates more number of peaks (05) in case of combined torsional and lateral vibrations as compared with torsional vibration (02). These peaks of mesh force will predominantly affect the performance of gear system and may cause fatigue. However, the magnitude of mesh force throughout the frequency range attains a lower value.

3.3 Effect of mesh stiffness on natural frequency

Effect of gear mesh stiffness on the modal frequencies of geared shaft system is predicted by varying mesh stiffness (10^6 to 10^7 N/m). Fig. 5 shows that as the mesh stiffness increases natural frequency increases baring mode 3 & 5.

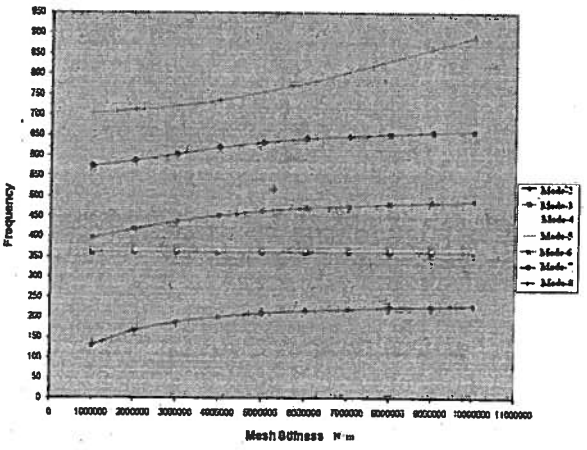


Fig. 5 Mesh stiffness Vs frequency

Table: 3

Mode Number	Frequency Hz		Percentage increase in frequency
	Minimum	Maximum	
1	0	0	0
2	130.4608	229.4231	75.8562
3	362.4117	362.4117	0
4	366.9895	368.108	0.3047
5	374.1679	374.1679	0
6	396.0175	486.9673	22.9661
7	573.2741	659.7224	15.0797
8	704.1497	892.0767	26.6850

Fig. 5 shows that the modal frequency varies with mesh stiffness. It has maximum variation (75.8562%) for second mode and (26.6850%) in case of 8 th mode (Table. 3).

3.4 Effect of bearing stiffness variation on natural frequency

The range of the selected bearing stiffness is $1x10^6$ to $1x10^7$ N/m with an increment of 10, 00,000. Fig. 6 shows the effect of bearing stiffness on natural frequency.

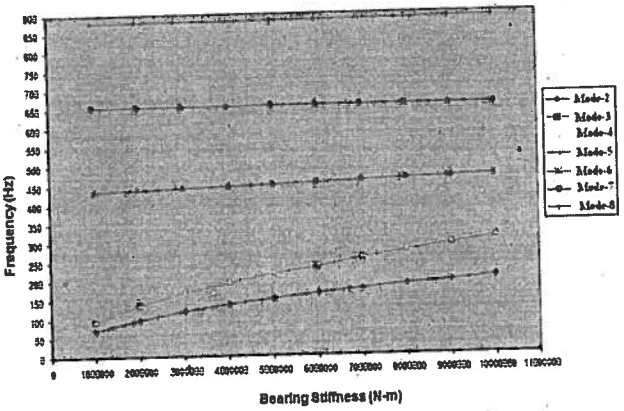


Fig. 6 Bearing stiffness Vs frequency.

Table: 4

Mode Number	Frequency Hz		Percentage increase in frequency
	Minimum	Maximum	
1	0	0	0
2	70.0846	202.5486	189.0058
3	96.8586	306.2938	216.2277
4	98.4418	311.1771	216.1026
5	100.0006	316.2296	216.2277
6	435.8012	469.8555	7.8141
7	654.8155	657.9297	0.4755
8	878.0877	887.2946	1.0485

Fig. 6 shows that the modal frequency varies with bearing stiffness. It has maximum variation (189.0058%) for second mode and (1.0485%) in case of 8 th mode (Table. 4).

It can be concluded that the increase in mesh stiffness causes considerable frequency change for higher modes than lower modes, the increase in bearing stiffness causes considerable frequency change for lower modes than higher modes.

References

(1) H Vinayak & R sinh, "Multi-body Dynamics and Modal Analysis of Compliant Gear Bodies", Journal of Sound and Vibration vol. 120 issue 2 pp. 171 – 214, 1998.

(2) J. Lin and R. G. Parker, "Sensitivity of Planetary Gear Natural Frequencies and Vibration modes to Modal Parameters", Journal of Sound and vibration vol. 228 issue 1 pp. 109 – 128 1999.

(3) R. G. Parker, S. M. Vijyakar and T. Imajo, "Non-Linear Dynamic Response of a Spur Gear Pair: Modelling and Experimental Comparisons", Journal of Sound and vibration vol. 237 issue 3 pp. 435 – 455 2000.

(4) Jian Lin and R. G. Parker, "Parametric Resonances in Two-Stage Gears from Fluctuating Mesh Stiffness", International Journal of gearing and Transmissions Issue. 3 pp. 127 – 134 2001

(5) David P. Fleming, "Effect of Bearing Dynamic Stiffness on Gear Vibration" 9th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery, 2002.

(6) Shengxiang Jia, Ian Hward and Jiande wang," The Dynamic Modeling of Multiple Pairs of Spur Gears in Mesh, Including Friction and Geometrical Errors" International Journal of Rotating Machinery vol. 9 pp. 437 – 442 2003.

(7) S. Graham Kelly, "*Fundamentals of mechanical vibrations*". Second edition, Mc Graw Hill Publications.

8) Trivedi T. R., "Vibration Analysis of Geared Shaft." M.E. mechanical (CAD/CAM) thesis, Bharati vidyapeeth university, college of engineering, Pune 2008.