

# The development of solvers for Symbolic Computational Dynamics



A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

By

Niloufar Motazed

The University of Sheffield

Department of Mechanical Engineering

June 2017







# Abstract

The research in this thesis deals primarily with the solution of the differential equation based models that are encountered in the study of engineering dynamics, and concentrates exclusively on enhancing processes of approximate analytical solution. This enhancement centres initially on the premise that asymptotic methods offer a powerful and adaptable group of formalisms that can be used, with care, to solve a very wide range of nonlinear dynamics problems, represented either as sets of nonlinear ordinary differential equations or as sets of nonlinear partial differential equations and boundary conditions. A further premise is that asymptotic methods can be structured in such a way that the user can apply them algorithmically, and an excellent example of this, which is used in this thesis, is the perturbation method of multiple scales. The final premise is that the algorithmic structure of the solution method lends itself to symbolic computation, and that this then offers the user an extremely powerful base-line tool for investigation, on the assumption that the tool is tested and reasonably validated.

It has been shown in this thesis that the perturbation method of multiple scales can be fully automated and that it is then capable of analysing very large-scale systems with many degrees of freedom. The usual expectation would be that the symbolic computations required to solve the problem would then give way to a numerical phase during which suitable data is substituted into the equations and then some form of high impact visualisation tool would be implemented for data output. However, numerical methods are highly dependent on the boundary conditions and the data used and they cannot necessarily generate a truly generalised solution to the problem in hand. They act like a 'black box' and as such can sometimes be seen to fail to present clearly the general physics of the systems being examined.

The alternative approach in applied dynamics is to use numerical integration directly to solve either the reduced-order, yet physically representative nonlinear model of the system, or even a fuller higher dimensional differential equation model. But it is still the case that numerical

output does not necessarily give a general picture, nor does it offer a full understanding of the relative significance of terms within the governing equations

In order to enhance the generality offered by approximate analytical modelling a new generation of solver has been proposed in the form of a wider computational study which has been termed Symbolic Computational Dynamics. The application process of Symbolic Computational Dynamics, as it is considered in this thesis, comprises two strands of symbolic computation. The first strand leads to the analytical solution to the differential equation model using a method such as multiple scales, noting that the code for this is termed the core solver. The second strand interrogates that solution continually, as it evolves in a structured manner, so that the underlying mathematical-physical links are established between terms, quantities, and operators. This process starts from the initial statement of the governing equations and continues right through to the final solution, and the strand of computation that does this is termed the term-tracker.

Therefore, the requirement of the solver part is to generate a general solution which gives an insight into the physics of the system, and the highly nonlinear interactions that frequently occur. Currently, practical application of these methods tends to be limited to models comprising a few degrees of freedom (or generalised coordinates), as they are mostly applied manually. This restriction is lifted when using a computer to do the analytical calculations and one can envisage moving from a few degrees of freedom to hundreds or even thousands.

Therefore the core solver comprises a symbolic approximate analytical method, based here on the perturbation method of multiple scales, and running as a process that is computationally transparent, rather like that which emerges when using a pen and a paper. The term-tracker generates all the extra information during the solution procedure. The fully symbolic solution procedure provides detailed information about each step of the analysis, while the term-tracker can highlight the connection between the assumptions and decisions that underpin the physical model, the resulting equations of motions, the solution procedure and then the final result in specific equation form.

In this study, the concept of a Symbolic Computational Dynamics solver has been advanced in several ways; firstly, an implementable version of early proposals for the Source and Evolution Encoding Method (SEEM) for comprehensive term-tracking is introduced, and a computerised basis for this is established. A generalised algorithm has been developed to apply the SEEM potentially to any symbolic solution procedure, so it is not at all limited to use with the

perturbation method of multiple scales. The SEEM makes it possible for the analyst to track each individual physical parameter and all physical-mathematical assumptions and simplifications through an approximate analytical solution process.

Secondly, the *Blueprint* visualisation method has been proposed, as a first attempt at visualisation of the results of a Symbolic Computational Dynamic solver. This method can interactively illustrate the connections between equations and the generated SEEM encodings information in a 3D graphical structure. This visualisation provides the analyst with new information that was previously hidden within the structure of the adopted solution procedure.

Finally, a combination of the number of quantities in each term and their encoding information is used to define a new normalised parameter, called the Strength Factor (SF). The SF value is implemented into the *Blueprint* visualisation method. The SF value can assist the user to estimate the strength of each term in an equation. Under a series of pre-defined conditions some terms can be considered generically negligible and then removed from the analysis; this being a somewhat new result in the application of asymptotic procedures such as multiple scales to problems of nonlinear dynamics. As a further development the possibility of manufacturing a solid three-dimensional printed structure for each solution method has been suggested. Moreover, the outputs of the new developments of Symbolic Computational Dynamic solver are discussed for two well known, and somewhat challenging, nonlinear dynamics case studies; those of the parametrically excited pendulum and the autoparametrically coupled beam problem, respectively.



# Acknowledgments

Everything started about seven years ago, where I put a freshly printed University of Sheffield poster on my bedroom wall and dreamed about my future. I am not sure why and how I was so sure that my journey begins from this university, but I was definitely right. Like everything else, my PhD journey was quite nonlinear, and it was almost impossible without supports and encouragements of my family, supervisor and friends.

At first, I want to thank my PhD supervisor, Prof Matthew Cartmell for his kind supports and guidance during this period. It was a very unique opportunity to have him as a supervisor. He taught me how to work hard, be innovative and passionate about my research.

I would like to thank my lovely husband Sam, for his kindness, encouragements and patience. I have always tried to adopt his work ethics and passion for engineering. I want to thank my parents, Abdolali and Fariba, for unconditionally supporting me to follow my own dreams, and keep believing in me. I want to acknowledge my sister, Yasaman, for her encouragements and sharing her own experiences of PhD life.

I want to thank Dr Jem Rongong for taking over the official supervision duty after departure of Matthew from Sheffield University. His contribution brought new perspective to the project. I would like to thank Dr Lizzy Cross for being my second supervisor, she was always supportive in difficult times. I would like to thank Prof Stephen Beck, my MSc supervisor, for keeping me involved in one of his interesting projects. Dr Olga Ganilova for her encouragements and kind friendship.

I want to acknowledge the Mechanical Department of University of Sheffield for funding this project. I should also thank my friends and colleagues who were always understanding and encouraging, Sayeh, Banafsheh, Daniela, Kartik, and Alex.



# Table of Contents

<b>Abstract</b>	<b>I</b>
<b>Acknowledgments</b>	<b>V</b>
<b>Table of Contents</b>	<b>VII</b>
<b>List of Figures</b>	<b>IX</b>
<b>List of Tables</b>	<b>XI</b>
<b>Nomenclature</b>	<b>XIII</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Motivation .....	1
1.2 Background .....	2
1.3 Aims and objectives .....	3
1.4 Thesis outline .....	4
<b>Chapter 2 Literature Review</b>	<b>5</b>
2.1 Numerical methods .....	6
2.2 Approximate analytical methods.....	9
2.3 Symbolic computational calculations .....	10
2.4 Term-tracking methods .....	12
2.5 Summary.....	17
<b>Chapter 3 Symbolic Computational Dynamics Solvers</b>	<b>19</b>
3.1 Introduction .....	19
3.2 Core solver .....	20
3.3 The Source Evolution and Encoding Method .....	22
3.4 Parametric excitation of a pendulum .....	24
3.5 Summary.....	45
<b>Chapter 4 Computerisation of the Source Evolution and Encoding Method</b>	<b>47</b>
4.1 The core solver.....	47
4.2 The general procedure for encoding application .....	47

4.3	The encoding algorithms .....	51
4.4	Displaying the encoding information .....	66
4.5	Symbolic Computational Dynamics solver .....	71
4.6	Summary .....	75
<b>Chapter 5 Visualisation Methods</b>		<b>77</b>
5.1	Introduction.....	77
5.2	Successful visualisation criteria.....	78
5.3	The ColourMap visualisation method .....	79
5.4	The Blueprint visualisation method .....	80
5.5	Computerisation of the visualisation method.....	82
5.6	A case study.....	89
5.7	Summary .....	100
<b>Chapter 6 An Evaluation Method for Symbolic Terms</b>		<b>103</b>
6.1	Introduction.....	103
6.2	The Strength factor .....	105
6.3	Implementation of the Significance Factor .....	106
6.4	An autoperametric case study .....	107
6.5	Post-processing the Strength Factor information.....	127
6.6	A three-dimensional structural analogy for the solution procedure .....	149
6.7	Summary .....	155
<b>Chapter 7 Conclusions and Future Work</b>		<b>157</b>
7.1	The Source and Evolution Encoding Method .....	157
7.2	The Blueprint visualisation method .....	158
7.3	A symbolic term evaluation method.....	158
7.4	Alternative visualisation for the solution procedure .....	158
7.5	Closing note.....	159
<b>List of Publications</b>		<b>161</b>
<b>References</b>		<b>163</b>
<b>Appendices</b>		<b>167</b>

# List of Figures

Figure 1-1 The six-year-old boy's first drawing [14]. .....	2
Figure 1-2 The second drawing [14]. .....	2
Figure 1-3 Is this just a differential equation? (For the answer check Figure 7-1). .....	2
Figure 2-1 A hinged beam loaded by a pressure pulse.....	7
Figure 2-2 The results of ten different finite element codes applied to a beam, loaded with a pressure pulse [29]. .....	7
Figure 2-3 Some examples of inherent Finite element codes errors [31]. .....	8
Figure 2-4 Explicit physical-mathematical links existing within a dynamical modelling procedure and approximate analytical solution .....	14
Figure 2-5 The Source Encoding Method applied to the first six equation in the analysis of a parametrically excited pendulum [69]. .....	15
Figure 2-6 An example that shows how encodings information can be displayed using the Tooltip function. ....	15
Figure 2-7 An example of showing history of a term using a tree structure in the Source Encoding Method [7]. .....	16
Figure 2-8 The application of unencodeEquation function to unencode and encode equations. ....	16
Figure 2-9 (a) The encoding vectors are mixed with the term during analysis, (b) Unique encoding vectors are removed (note this a partial presentations of an equation). .....	17
Figure 3-1 Overview of a generic Symbolic Computational Dynamic solver.....	20
Figure 3-2 Overview of possible Symbolic Computational Dynamic solver in the future. ....	20
Figure 3-3 Appearance of the solution procedure in a Symbolic Computational Dynamics solver.....	21
Figure 3-4 Configuration and symbols used to describe the problem of a parametrically excited pendulum [15]. .....	25

Figure 4-1 A graphical representation of the decomposition process of an equation into the fundamental quantities. ....	48
Figure 4-2 A graphical representation of the combination process; encoded quantities into the encoded equation. ....	48
Figure 4-3 A graphical representation of ‘the SEEM encoding cycle’. ....	50
Figure 4-4 The encoded information is always one step behind the solution procedure. ....	59
Figure 4-5 Graphical representation of the relationship between the SEEMencoder and the SEEMcore functions.....	66
Figure 4-6 The output of the SEEMencoder function format, for the equation of motion of in-plane response of the problem of the autoparametrically excited beam system. ....	67
Figure 4-7 A graphical demonstration of the application of the Tooltip function for an arbitrary encoded expression; the highlighted area shows the mouse pointer position. ....	68
Figure 4-8 A graphical demonstration of the application of the Labeled function for an arbitrarily encoded expression.....	68
Figure 4-9 A graphical representation of the Symbolic Computational Dynamic notebook. ....	72
Figure 4-10 The encoding and post-processing functions defined at the beginning of a given Symbolic Computational Dynamic notebook.....	72
Figure 4-11 A screen-shot from a defined input equation in a Symbolic Computational Dynamics solver.....	73
Figure 4-12 A screen-shot from the SCD solver; how equations are defined, encoded and displayed.....	74
Figure 4-13 An example showing the transparent structure of the SCD solver.....	74
Figure 5-1 A graphical review of the development of Symbolic Computational Dynamics in chapter 3 and 4.....	78
Figure 5-2 Initial overview of the ColourMap visualisation method.....	79
Figure 5-3 A graphical representation of the ColourMap visualisation method.....	80
Figure 5-4 A graphical representation of the initial idea for the Blueprint visualisation method.....	81
Figure 5-5 User panel for setting up the visualisation method.....	82
Figure 5-6 Control panel of the visualisation interface.....	83
Figure 5-7 The output of the two-dimensional Blueprint visualisation method.....	84
Figure 5-8 Mathematica bug in visualising different link tags. ....	84
Figure 5-9 Solving the Mathematica bug in tagging links. ....	85
Figure 5-10 The three-dimensional Blueprint visualisation panel. ....	86
Figure 5-11 An example of the output for the three-dimensional visualisation.....	87

Figure 5-12 A graphical example of the interactive plots for the SCD solvers. .... 87

Figure 5-13 The numerical response of the case study is plotted based on the symbolic general solution. .... 88

Figure 5-14 An interactive animation of the response based on the symbolic solution. .... 88

Figure 5-15 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem. .... 89

Figure 5-16 The automatically generated Source and Evolution Encoding Method information for equation (2) of the parametrically excited pendulum problem. .... 89

Figure 5-17 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem. .... 90

Figure 5-18 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem. .... 90

Figure 5-19 The automatically generated Source and Evolution Encoding Method information for equation (5) of the parametrically excited pendulum problem. .... 90

Figure 5-20 The Blueprint visualisation for the fundamental equations in the analysis of the parametrically excited pendulum problem. .... 90

Figure 5-21 The visualisation graph up to as far as equation (5) for the analysis of the parametrically excited pendulum problem. .... 91

Figure 5-22 The overall Blueprint visualisation graph for equation (6). The first level of the encodings for  $\theta_0$  is highlighted in orange, and the second level in yellow. .... 92

Figure 5-23 Detailed discussion for selected terms in equation (6), where; (a) is nonlinear stiffness term, (b) is the external excitation term, (c) is a perturbational correction term, (d) damping term, (e) is a perturbational correction term, and (f) is the inertia term. .... 94

Figure 5-24 Encoding information for equation (7). .... 94

Figure 5-25 Overall visualisation graph for equation (8). .... 95

Figure 5-26 The detailed encoding information for selected terms in equation (8); where, (a) is the inertia term, (b) is the first order perturbation term, (c) is the external excitation term, (d) is the damping term, (e) is the nonlinear stiffness term, and (f) is a perturbational correction term. .... 96

Figure 5-27 Encoding information for the solution of the zeroth-order perturbation equation. .... 97

Figure 5-28 Overall visualisation graph for equation (10). .... 98

Figure 5-29 A selected detailed discussion of the SEEM information for equation (10); where, (a) is the nonlinear restoring force, (b) and (c) are the definition of the restoring force based on the zeroth-order perturbation solution.....	99
Figure 5-30 Encoding information for equation (11). .....	100
Figure 6-1 A graphical example of two terms with different Blueprint encoding structures. ...	104
Figure 6-2 The development procedure of the Symbolic Computational Dynamics solver in this thesis. ....	105
Figure 6-3 A graphical example of implementation of the Strength Factor number in the Blueprint visualisation graph.....	107
Figure 6-4 Coupled beam system based on Cartmell [70]. AB is the primary beam, and BC is the secondary beam. $W(t)$ represents the external excitation. ....	108
Figure 6-5 The visualisation graph for the primary beam's equation of motion. ....	109
Figure 6-6 A screenshot of the automatically generated encodings for eqn (1). ....	110
Figure 6-7 Overall visualisation graph for eqn (2). ....	111
Figure 6-8 Overall visualisation graph for eqn (3). ....	112
Figure 6-9 Overall visualisation graph for eqn (4). ....	113
Figure 6-10 Overall visualisation graph of eqn (5). ....	114
Figure 6-11 Overall visualisation graph of eqn (6). ....	114
Figure 6-12 Fundamental equations for the problem of autoparametrically excitation of a coupled beam system.....	115
Figure 6-13 Overall visualisation graph for eqn (7). ....	115
Figure 6-14 A detailed investigation of the encoding for eqn (7); (a) is the external excitation term, (b) is the linear stiffness, (c) is a stiffness term, (d) is the damping term, (e) and f are coupling terms, (g) is a perturbational correction term, and finally (h) and (i) are the inertia terms. ....	118
Figure 6-15 Overall visualisation graph for eqn (8). ....	119
Figure 6-16 A detailed investigation of the encoding information for eqn (8); where, (a) is a stiffness term, (b) is a perturbational correction term, (c) is the damping term, (d) is a perturbational correction term, and (e) is the coupling term.....	120
Figure 6-17 A detailed investigation of the encoding information for eqn (9). ....	121
Figure 6-18 A detailed investigation of the encoding information for eqn (10). ....	122
Figure 6-19 Overall visualisation graph for eqn (11). ....	123

Figure 6-20 A detailed investigation of eqn (11); (a) is a linear stiffness, (b) is the inertia, (c) is the external excitation, (d) is a damping term, (e) is a coupling term, and finally (f) is a perturbational correction term. ....	125
Figure 6-21 Overall visualisation graph for eqn (12). ....	125
Figure 6-22 A detailed investigation of the encoding information for eqn (12); where, (a) is the stiffness term, (b) is the inertia, (c) is the damping term, and (d) is the perturbational correction term. ....	126
Figure 6-23 A typical Strength Factor graph for the eqn (2) of the autoparametrically excited coupled beam system. ....	127
Figure 6-24 The original and modified amplitude responses for the primary (a) and secondary (b) beams (based on the numerical values given in Table 6-3. ....	129
Figure 6-25 The original and modified amplitude responses for the primary and secondary beams (based on varied numerical input data values). ....	130
Figure 6-26 The Strength Factor graph of eqn (1). ....	131
Figure 6-27 The original and modified responses after removing $X\tau * X\tau\gamma\epsilon nsf$ . ....	132
Figure 6-28 The Strength Factor plot of eqn (7). ....	133
As the results, $2dr\epsilon drD0D1X0$ , and $X\tau * X\tau\gamma\epsilon nsf$ , $D02X1\epsilon p$ , and $\epsilon p\omega 12X1$ are considered less effective terms in this equation. As discussed in Figure 6-29 removing $X\tau * X\tau\gamma\epsilon nsf$ has no particular effect on the response of the system. ....	133
Figure 6-30 The original and modified responses after removing $2dr\epsilon drD0D1X0$ . ....	134
Figure 6-31 The original and modified responses after removing $D02X1\epsilon p$ . ....	135
Figure 6-32 The original and modified responses after removing $\epsilon p\omega 12X1$ . ....	135
Figure 6-33 The Strength Factor configuration for eqn (8). ....	136
Figure 6-34 The original and modified responses after removing $\epsilon p\omega 22Y1$ . ....	137
Figure 6-35 The original and modified responses after removing $D02Y1\epsilon p$ . ....	137
Figure 6-36 The original and modified responses after removing $2drD0D1Y0\epsilon dr$ . ....	138
Figure 6-37 The Strength Factor graph for eqn (11). ....	139
Figure 6-38 The original and modified responses after removing $2drD0D1X0\epsilon dr\epsilon p$ . ....	140
Figure 6-39 The Strength Factor configuration for eqn (12). ....	141
Figure 6-40 The original and modified responses after removing $2drD0D1Y0\epsilon dr\epsilon p$ . ....	142
Figure 6-41 The Strength Factor configuration of eqn (20). ....	142
Figure 6-42 The original and modified responses after removing $iA2dmpf\epsilon dmpf\zeta 1\omega 12\epsilon p$ . ....	143
Figure 6-43 The original and modified responses after removing $iD1A2dr\epsilon dr\omega 1\epsilon p$ . ....	144

Figure 6-44 The Strength Factor configuration for eqn (22). ..... 145

Figure 6-45 The original and modified responses after removing  $iB2dm\psi\epsilon d\psi\zeta2\omega22\epsilon p$ .  
 ..... 146

Figure 6-46 The original and modified responses after removing  $iD1B2dr\epsilon dr\omega2\epsilon p$ . ..... 146

Figure 6-47 The Strength Factor configuration for eqn (24). ..... 147

Figure 6-48 The original and modified responses after removing  $2AA\_y\epsilon nsf\omega12\epsilon p$  ..... 148

Figure 6-49 The original Blueprint visualisation graph for the first nine equations in the  
 problem of the auto-parametrically resonant coupled beam system. .... 149

Figure 6-50 The 3D graph positioning algorithms in Mathematica..... 150

Figure 6-51 The modified Blueprint visualisation graph for the first nine equations in the  
 autoparametrically resonant coupled beam system..... 150

Figure 6-52 An alternative presentation of the Blueprint visualisation method for the solution  
 of the parametrically excited pendulum problem. .... 151

Figure 6-53 An alternative presentation of the Blueprint visualisation method for  
 autoparametrically resonant coupled beam system..... 152

Figure 6-54 A conjectured 3D printed sample of a solution procedure for a parametrically  
 excited pendulum problem. .... 153

Figure 6-55 Different angle views of a conjectured 3D printed sample of a solution procedure  
 for a parametrically excited pendulum problem. .... 153

Figure 6-56 Plan view of the Blueprint visualisation for the autoparametrically resonant  
 coupled beam system, showing the physical sub-systems of the problem in the form of  
 cylindrical regions A and B. .... 154

Figure 7-1 This is not just a differential equation. .... 160

# List of Tables

Table 2-1 The percentage of applying nine types of assumptions in ten case studies [38].	13
Table 3-1 The Source Evolution and Encoding method logic summary.	22
Table 3-2 The percentage of contribution of each equation for the final solution.	45
Table 4-1 An example of the index term and term for an arbitrary term.	54
Table 4-2 The Source and Evolution Encoding vector format is compared to the computerised encoding vector.	57
Table 4-3 Criteria that are checked for finding the position of the argument of an exponential function.	60
Table 4-4 An example showing that the SEEMcore function is independent of the input function operator.	62
Table 4-5 Criteria that are checked for finding the position of a quantity in the base of a power function.	63
Table 4-6 Display format option structures.	69
Table 6-1 A points system in calculation of the Strength Factor.	106
Table 6-2 Calculating the Strength Factor value for a term.	106
Table 6-3 Experimental data for the coupled beam system parameters [70].	128
Table 6-4 The Strength Factor guideline applied to eqn (1).	131
Table 6-5 The Strength Factor guideline applied to eqn (7).	133
Table 6-6 Summarising the Strength Factor term selection for a case study.	148



# Nomenclature

$A$	Arbitrary complex function of the slow time scale
$a$	The amplitude of the response
$B$	Arbitrary complex function of the slow time scale
$b$	Amplitude of response for the secondary beam
$c$	Damping coefficient for the pendulum case study
$D$	D-Operator notation is used to define the partial derivatives
$dt[i]$	$i$ th time derivative operator
$g$	Acceleration due to gravity
$j$	Actual equation number, input of the encoding function
$k$	The actual source of the equation
$l$	Pendulum rod length
$m$	Pendulum mass
$n_i$	The $i$ th digit of the Source and Evolution Encoding number
$q$	The support displacement as fraction of the pendulum length
$s$	Relevant side of the equation, input of the encoding function
$t$	Real time
$T$	Kinetic energy

$T_0$	Fast time scale
$T_1$	Slow time scale
$u$	External excitation force for the pendulum case study
$V$	Potential energy
$W$	The external excitation
$X$	In-plane response of the couple beam system
$x$	Arbitrary equation number in chapter 3
$X_0$	Zeroth-order perturbation term for the in-plane response of the coupled beam
$X_1$	First order perturbation term for the in-plane response of the coupled beam
$Y$	Out-of-plane response of the coupled beam system
$y$	Arbitrary equation number used in chapter 3
$Y_0$	Zeroth-order perturbation term for the out-of-plane response of the coupled beam system
$Y_1$	First order perturbation term for the out-of-plane response of the coupled beam system
$z$	Arbitrary equation number
$\alpha$	The phase angle
$\beta$	Damping ratio in chapters 3, 4, and 5. Also the phase angle for the secondary beam in chapter 5
$\gamma$	Nonlinear stiffness coefficient
$\varepsilon$	Small parameter
$\zeta_1$	The damping coefficient for the primary beam of the coupled beam system

$\zeta_2$	the damping coefficient for the secondary beam of the coupled beam system
$\theta$	Instantaneous angle of rotation for the pendulum case study
$\theta_0$	Zeroth-order perturbation term for the pendulum case study
$\theta_1$	First order perturbation term for the pendulum case study
$\mu$	The coefficient of centripetal acceleration of the coupled beam
$\sigma$	Detuning parameter
$\tau$	Nondimensional time based on the natural frequency, expect chapter 6 where it is equal to real time
$\varphi$	Phase angle of the pendulum case study
$\phi$	The damping torque for the pendulum case study
$\Omega$	External excitation frequency
$\omega$	Nondimensional frequency of excitation ( $\Omega/\omega_0$ )
$\omega_0$	The first linear natural frequency of the pendulum
$\omega_1$	The first linear natural frequency for the primary beam
$\omega_2$	The first linear natural frequency for the secondary beam

### Subscripts

$cplf$	Primary beam coupling term
$cpls$	Secondary beam coupling term
$dmpf$	Primary beam damping term
$dmpr$	Secondary beam damping term
$dp$	Pendulum damping term
$dr$	Second time derivative

<i>ex</i>	External excitation term
<i>ns</i>	Nonlinear stiffness
<i>p</i>	Perturbation parameter
<i>PPR</i>	Principle parametric response
<i>res</i>	Resonance conditions

### Abbreviations

<i>envec</i>	Encoding vector
CEV	Computerised Encoding Vector
<i>eqn</i>	Equation
HOT	Higher Order Terms
<i>indextrm</i>	All the encoding vectors for a particular term
LHS	Left-hand side
nan	Not a number
RHS	Right-hand side
SCD	Symbolic Computational Dynamics solver
SEEM	Source and Evolution Encoding Method
<i>trm</i>	All the quantities without the encoding information in an equation

### Syntaxes

#	Simplification of exponential functions for the second time
\$	Simplification of exponential functions for the third time
*	Simplification of exponential functions for the first time
**	Simplification of exponential functions for the fifth time
£	Simplification of exponential functions for the fourth time

# Chapter 1 Introduction

Symbolic Computational Dynamics solvers are capable of applying analytical solution methods to reduced order problems expressed in differential equation form. A general solution for the problem of interest, to some degree of accuracy, can generally be determined using the mathematics underpinning these methods. The intention behind developing the subject area of Symbolic Computational Dynamics is to provide a new way of gaining insight into the physics of wide-ranging dynamical systems and to uncover the deepest possible information about the highly nonlinear interactions that frequently occur.

The research topic of Symbolic Computational Dynamics was initiated around twenty years ago and the research reported in this thesis is a major part of an ongoing development of software techniques necessary to create usable Symbolic Computational Dynamics solvers. This work builds on ideas originally proposed by Cartmell, Khanin, and Forehand [1-13], and which have been substantially improved through novel implementations in software by the author of this thesis. Many new ideas have been contributed through the research reported in this thesis and as a result the topic is now about to enter a new phase of development.

## 1.1 Motivation

The story of this research is in some ways similar to the introduction of the Little Prince Book, by Antoine de Saint-Exupéry [14]. In this story, a six-year-old boy was looking at a book called “true story from nature”, and he saw a picture of a boa swallowing an animal. The boy considered the picture and made his own drawing, as shown in Figure 1-1. The boy then showed his drawing to some adults and wondered if they were going to be scared by it. Interestingly, they looked at the drawing and wondered why should anyone be frightened by a hat. Then the boy begins to show extra information about his drawing, by making a new one, see Figure 1-2.

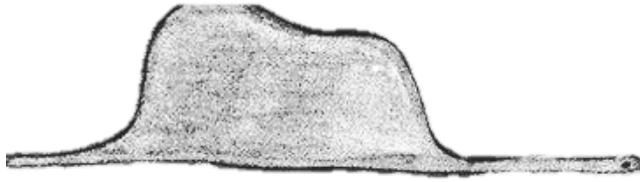


Figure 1-1 The six-year-old boy's first drawing [14].

Without having certain key information about a topic it is possible to be entirely misguided about it. In this example, without the second drawing, it is almost impossible to say that the real object is an elephant in a boa's stomach, and most people would be happy to think of it as a hat, without striving to know what else it could be.



Figure 1-2 The second drawing [14].

Taking the discussion further, into an engineering context, in particular such as the study of nonlinear dynamics, it is possible to look at Figure 1-3 and ask: "*is this just a differential equation?*", or to wonder what lies within that equation, and whether an understanding of that inner part of the definition might present a larger, and possibly far less obvious context. This research endeavours to offer a fresh perspective for asymptotic symbolic solutions in the field of nonlinear dynamics.

$$D_0^2 \theta_0 + \varepsilon D_0^2 \theta_1 + 2\varepsilon D_0 D_1 \theta_0 + 2\varepsilon \beta D_0 \theta_0 + \theta_0 + \varepsilon \theta_1 - \varepsilon q \omega^2 \theta_0 \cos(\omega T_0) + \varepsilon \gamma \theta_0^3 = 0$$

Figure 1-3 Is this just a differential equation? (For the answer check Figure 7-1).

## 1.2 Background

There is no absolute linear system in the universe, and as Jon Juel Thomsen said: "*A linear system represents a mathematical abstraction a useful and productive invention of the human mind*" [15]. In some cases, it is possible to assume that the system is constrained to operating within a limited range, and so the response can be calculated using a linear theory. The theories

developed for studying linear systems are quite advanced and have numerous robust application methodologies, as long as the numerical limitations and constraints are well known in advance. However, a great deal has been uncovered and understood about nonlinear dynamics, particularly in the last 40 years, the general concept of nonlinearity still contains extensive areas yet to be discovered.

In the world of mechanical systems and structural dynamics it is possible to categorise nonlinearity sources into four main groups: geometrical nonlinearities, material nonlinearities, nonlinear body forces, and physical configuration nonlinearities. Regardless of the source, nonlinearities are always mathematically shown in a unique form in the governing equations of motion. Due to the inherent complexity of nonlinear systems, both the physical model and mathematical solution procedure representing these systems always include some form of simplifications or assumptions.

The ultimate goal of this research is to offer a new perspective to analyse nonlinearity within engineering problems. The creation of a new set of computational tools aims to establish a connection between the physics of the actual system, the mathematical solution procedure, and the final response. In other words trying to define a mathematical-physical meaning for each term involved in the mathematics of the adopted solution procedure. Furthermore, the ability to track the effects of each, and every assumptions or simplification in the analysis is seen to be an essential constituent part of this process, in order to ensure no loss of information.

### **1.3 Aims and objectives**

The aim of this research is to advance the functionality of the computational tools that are so vital to the development of the full implementation of Symbolic Computational Dynamics, and in so doing to create a fresh perspective for symbolically analysing the sorts of nonlinear differential equations encountered in mechanical engineering dynamics. Ultimately, if properly developed, such a tool should be able to create a complete information link between the physical concept of an engineering problem, the procedure used to solve it, and the final response. To achieve this target the followings objectives may be defined:

- To develop and demonstrate the concept of the Source and Evolution Encoding Method as a basis for implementing the historical tracking features central to the concept of symbolic computational dynamics.
- Then to create a practical computational basis for implementing the Source and Evolution Encoding Method into Symbolic Computational Dynamics.

- After this, to develop and then introduce a precursor method for interactive visualisation of the mathematical features inherent to Symbolic Computational Dynamics.
- Finally, to propose a symbolic term evaluation method that can be used to distinguish between different levels of importance of terms within a differential equation model, and then to use that as a basis for identifying negligible terms in the equations.

## **1.4 Thesis outline**

This thesis consists of seven chapters. After the introduction, the second chapter provides a concise summary of the previous research in the topic of symbolic calculations. Chapter three is the heart of this thesis and discusses the methodology of the Source and Evolution Encoding Method. This is followed by the fourth chapter, which includes the algorithms that have been developed to implement the Source and Evolution Encoding Method within a demonstrator Symbolic Computational Dynamics solver. Chapter five includes the process of developing and implementing the first visualisation method for the demonstrator Symbolic Computational Dynamics solver. Then, leading on from this, chapter six proposes the very first symbolic term evaluation concept, based on the Source and Evolution Encoding Method. Finally, chapter seven provides conclusions and suggestions for future work for this research.

# Chapter 2 Literature Review

Traditionally some of the more difficult engineering dynamics problems have been solved manually with a pen and paper, using complex modelling strategies, a mixture of physically and mathematically based assumptions, and highly bespoke mathematics. This has inevitably required a combination of a highly expert and knowledgeable analyst, possessing a solid background in engineering, physics and mathematics, and the identification of a meaningful yet tractable solution procedure. Deriving an accurate solution for even a single degree of freedom nonlinear problem can be very challenging indeed, and this difficulty can then be dramatically upscaled when complicated, coupled nonlinear systems are taken into account.

Numerical analysis methods (for ordinary differential equations) of engineering dynamics can be based on numerical integration techniques and in a more generalisable sense on finite element analysis, and a very large number of commercial codes are now available for both approaches. Generally, it is the case for commercial numerical solvers that the complete solution process is automated; from the initial stages in which the equations of motion are constructed in numerical form, through the solution generation process, with built-in sophisticated convergence and error self-analysis techniques provided, to the final post-processing stages required for graphical output and visualisation, possibly in the form of animations running against a time scale.

There is no doubt that validated and benchmarked commercialised platforms based on numerical analysis are of immense utility and practical importance, however it can also be argued that as such computational systems become more automated and less transparent, the analyst potentially loses certain forms of control so that the processes of physics-based derivation and physics-influenced solution modification and specialisation are not as apparent as they necessarily have to be when approximate analytical modelling and solution is attempted, usually manually, by an expert engineer or engineering mathematician.

As a consequence of this insight, a new approach to modelling and solution based on specialised computation known generically as Symbolic Computational Dynamics, has been introduced. Key

features of this approach are highlighted, discussed, and developed in detail within this thesis. This chapter aims to provide a brief review of the context, the history where relevant, and the previous research that has been carried out in the development of symbolic solvers for certain types of nonlinear dynamics problems.

## 2.1 Numerical methods

This section only considers the finite element methods that can be based on processes of numerical integration, on the premise that some forms of computable differential equations of motion can be derived and structured appropriately within the solver, or in the formal process of finite element analysis. The latter is particularly powerful as the computational structures necessary for the formulation of finite element algorithms lend themselves naturally and efficiently to many elegant forms of scaling up. This leads to finite element analysis, which is now routinely capable of intricate and potentially accurate calculations of dynamic responses in the largest and most complicated structures. However, despite this, there is the over-riding fact that generic and numerically independent informatic linkages between the physics of the problem and the emergent solution still remain elusive and are very difficult to define within any finite element formulation, no matter how sophisticated it may be.

In the finite element methods, the structure of interest is generally discretised into several small elements, and these elements are then connected at key points known as nodes, according to precise rules which accord appropriately with the underlying physics of the problem. These processes result in a series of algebraic equations, which can be expressed numerically within the procedure used and then an approximate solution for the unknowns can be determined at each node [16].

Finite elements are used in various application areas of engineering such as statics, dynamics, thermo-fluids, electrodynamics, and more recently across multi-physics domains. For example, *NASTRAN* (NASA STRucture Analysis) [17] is a finite element based programme, which was developed in 1964, originally for the design of space vehicles. The commercial version of this software can now be used for analysis across many different industries, for which definable physics can be exploited for modelling and solution. Another powerful and popular numerical package is *MSC ADAMS* [18], noting that this is capable of analysing complicated nonlinear dynamics problems. The *Vortex* package, based on a Lagrangian formulation [19], transforms static CAD models to interactive systems in a 3D environment and is capable of modelling real-time dynamic reactions, deformations, collisions, and rigid body dynamics [20-22].

There are several advantages offered by finite element analysis, not least that they are capable of producing approximate solutions for problems that are very complicated and where analytical solutions cannot be found, or where methods for doing such analysis cannot readily be defined. Also, it is possible to model highly complex geometries and loadings within finite element analysis [23-27].

Despite many powerful features inherent to finite element analysis, there are certain sources of error that must be always considered. Regardless of the modelling quality, the output of commercial finite element packages is invariably portrayed in a colourful eye-catching form and this can sometimes obscure fundamental inaccuracies in the solution.

For example, considering the investigation that had been carried out by Yu [28]. As Figure 2-1 shows, a pressure pulse was applied to a straight beam, the beam was hinged at both ends. The response of this system was defined as a lateral displacement of the beam at the middle point, against time. To determine the response, ten validated finite element codes have been used by ten expert users. As Figure 2-2 shows, it is not possible to identify which one of these codes is showing a valid result [29, 30].

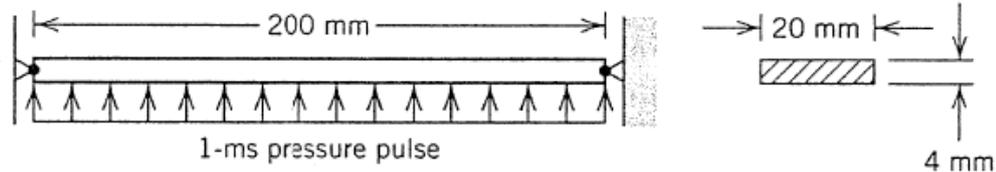


Figure 2-1 A hinged beam loaded by a pressure pulse.

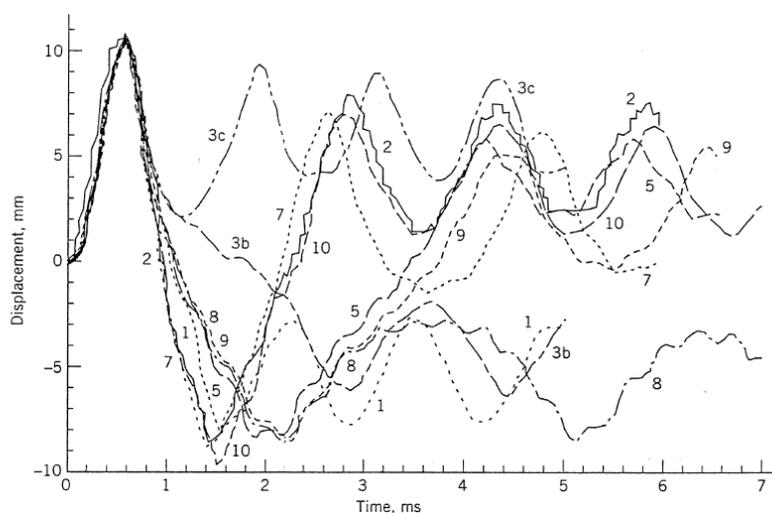


Figure 2-2 The results of ten different finite element codes applied to a beam, loaded with a pressure pulse [29].

Computational errors can be inherent to finite element formulations. For instance, the computational domain is always a simplified form of the real geometry. As Figure 2-3-a shows, an actual structure is not equivalent to its discretised model [31].

The other source of error is the fact that finite element methods are based on piecewise polynomial interpolation. In a typical finite element analysis, the structure of interest is discretised into small elements to evaluate a field quantity; the field quantity is the quantity of interest in the analysis, for example: it can be displacement, stress, and so on. The field quantity for each element is then determined by interpolation of the field values at nodes of that element. Afterwards, the elements are connected together and the overall field quantity is determined by piecewise polynomial interpolation. Therefore, the number of polynomial expression is equal to the number of elements. As interpolation is an approximation of a continuous function, therefore, it can lead to inaccuracy in the result of a finite element analysis, see Figure 2-3-b.

Furthermore, the result of numerical integrations within the finite element formulation is an approximation [32]. Gauss quadrature integration approximates an integral into a weighted summation of a finite number of terms, the formulation shown in Figure 2-3-c.

Finally, there is the ubiquitous problem of specifying a necessary and sufficient number of digits for numerical representation in the computer code itself, see Figure 2-3-d.

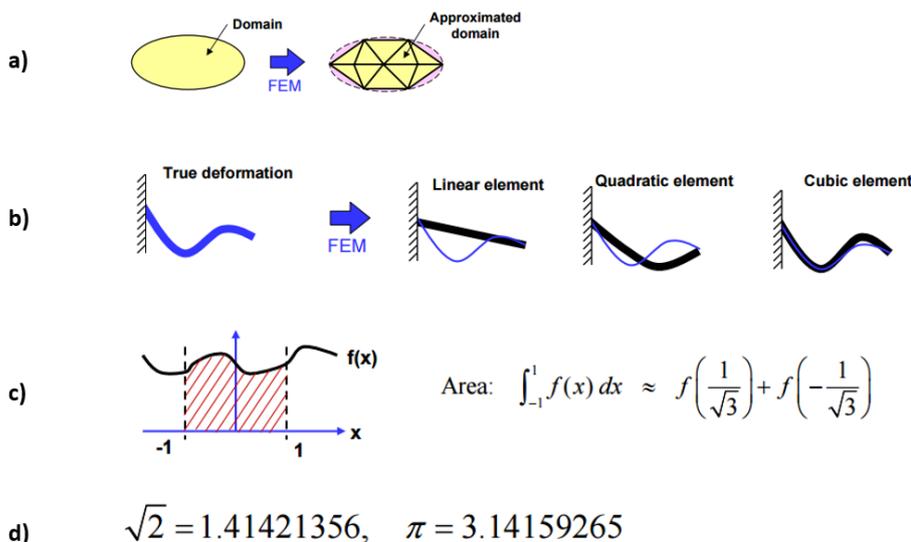


Figure 2-3 Some examples of inherent Finite element codes errors [31].

## 2.2 Approximate analytical methods

In these methods the nonlinear solution is derived by perturbing the response of the corresponding linear system [33]. The approximate analytical methods are generally valid for weakly nonlinear systems, whereby the nonlinear terms are considerably smaller than the linear ones. In other words, the motion must be finite with a moderate amplitude to have an acceptable accuracy [15]. However, it can be noted that the multiple scales method has been applied to strongly nonlinear systems in several studies such as Lakrad and Belhaq [34].

There are several approaches which can be used to apply mathematical perturbation to a system, such as: The *Straight Forward Expansion*, the method of *Lindstedt–Poincaré*, *Harmonic Balance*, and the method of *Multiple Scales*. The choice of the perturbation method is mostly a matter of personal preference, in general none of these methods has a significant superiority over others [15]. Therefore, in this research for the convenience of the author, the method of multiple scales is practiced. Note that it is possible to reach the same conclusion by applying the other methods.

The method of multiple scales is originally developed by Nayfeh [35]; where, the dependent variables in the equation of motion are represented by using two or more independent variables (or scales). The underlining feature of this method is to take time (equation (2.1)) as a series of independent time scales. The expansion is assumed to be uniformly convergent. The variables or coordinates must be expressed by a uniformly valid expression to create a set of perturbation equations [35-37].

A fast time scale ( $T_0$ ) is used for describing dynamics for which the operating frequencies are near the linear natural frequencies of the system, fast time scale is equal to the real time ( $t$ ). However, the slower time scales are employed to define slow variations of amplitude and phase. The final solution is built-up gradually by formulating the results of each perturbation. The first and second time derivatives are determined based on the perturbation expansion and shown in equations (2.3) and (2.4), respectively. The number of independent time scales is limited by the order at which the expression is truncated. In these equations  $\varepsilon$  is a small perturbation parameter and  $T_j$  shows the slower time scales.

$$T_j = \varepsilon^j t \quad (2.1)$$

$$x(t, \varepsilon) = \sum_{j=0}^{m-1} \varepsilon^j x_j(T_0, T_1, \dots, T_m) + O(\varepsilon T_m) \quad (2.2)$$

$$j = 0, 1, 2, \dots$$

$$\varepsilon \ll 1$$

$$D_j = \frac{\partial}{\partial T_j}$$

(2.3)

$$\frac{d}{dt} = D_0 + \varepsilon D_1 + O(\varepsilon^2) + \dots$$

$$\frac{d^2}{dt^2} = D_0^2 + 2\varepsilon D_0 D_1 + O(\varepsilon^2) + \dots$$

(2.4)

To investigate the functionality of this method, the solution procedure for a problem of the parametrically excited beam is investigated in part B of the appendix. Furthermore, an experimental investigation for this problem has been conducted in this research. The results of the experimental and theoretical solution are compared.

## 2.3 Symbolic computational calculations

Symbolic computational calculations can considerably increase insight into the dynamics of a system [3]. The ideal case is to develop a powerful symbolic solver that has the ability to apply analytical methods into complex dynamical problems [3, 38]. The hierarchical structure and conceptual simplicity of the perturbation methods, especially the multiple scales method, have made them more conventional to be implemented to a computerised algorithm. The development of the symbolic computational codes has started with formulations that were only capable of deriving equations of motion, later the topic of symbolic solvers was introduced.

### 2.3.1 Symbolic generation of equation of motion

Multibody codes are mainly based on two formulations: The Eulerian and the Lagrangian. In the Eulerian codes, the analysis starts with the Newton and Euler equations, while, Lagrangian codes are based on Lagrange's and Hamilton equations [39].

In 1983, the SYMBOD was developed as a computerised symbolic programme. This code was capable of deriving equations of motion automatically [40]. In 1985, NEWEUL [41, 42] was programmed to generate equations of motion symbolically, based on the Newton-Euler principle. Lieh and Haque [43] developed a closed-form equation formulation, based on the principle of virtual work. It was stated that the algorithm had the ability to generate automatically both the linear and nonlinear equations of motion symbolically. The output of this algorithm was in the form of a FORTRAN code [44] and it was possible to display equations like

a handwritten document. More efforts on generating formulations based on the both Lagrange [45-47] and Newtonian [48, 49] have been made. As a result there are a few fully symbolic codes that can be used to generate and solve formulations for the dynamics of flexible bodies [50-54].

Forehand, Khanin [9] have developed the MultiFlex.m function in *Mathematica* code. This Lagrangian formulation can derive equations of motion for a multibody system symbolically. The initial code was based on the Fissette, Johnson [52] method, this package was capable of modelling both rigid and flexible bodies.

### **2.3.2 Symbolic generation of the solution procedure**

In 1987, Rand and Armbruster [55] developed the MACSYMA, which was capable of applying some of the perturbation methods to oscillating systems. The perturbation equations were generated symbolically, however, the solution for these equations were only available for limited cases. Furthermore, MACSYMA was not designed to evaluate solutions for the modulation equations (secular equations). The user intervention in the analysis process was compulsory in some cases [56].

Large numbers of the symbolic solvers have been developed within the Maple interface [57]. For example; Sanchez [58] symbolically applied the method of multiple scales to a single degree of freedom problem. Later, Corless, Jeffrey [59] studied the application of a perturbation method for two fluid mechanics related problems. Furthermore, Vakhidov and Vasiliev [60] developed a code based on the Hamiltonian formulation. Yu [61-63] developed a symbolic solver based on the multiple scales, this code used the number of real and complex eigenvalues for the linearized system, and the order of normal forms as inputs. Wang, Steyn-Ross [64] developed a semi-automated symbolic solver based on the multiple scales method. Other examples can be found that were involved in developing a symbolic code, based on a perturbation method, such as Franciosi and Tomasiello [65], Pismen *et al.* [66, 67].

Khanin and Cartmell [12] introduced an advanced symbolic computational solver, based on two perturbation methods. In this study, both the straight forward expansion and multiple scales method were applied to the equation of motion of a pendulum. This solver was programmed within the *Mathematica* code interface. Later, Khanin, Cartmell [2] developed a generalised symbolic solver, based on the multiple scales method. This solver was implemented into a *Mathematica* function, named "MultipleScales.m". The generality and wider range of applications are the standing points of this solver, comparing to the other existing symbolic codes. Furthermore, this code is able to perform the perturbation analysis in both forms of

automated and semi-automated. In order to increase the application of this code, a parallelised version [1, 11, 13] was introduced.

The Khanin, Cartmell [2] symbolic solver has been used in several studies, for instance; the nonlinear cutting model for a primary chatter was evaluated using this package [68]. All possible resonance conditions were determined using this powerful function. Moreover, the results of this study were compared with other numerical methods (neural network), the package performance was described as quick and sophisticated.

A thorough review on the multiple scales methods and special cases has been done by Cartmell, Ziegler [5]. In this paper, special case studies and different versions of the multiple scales were discussed. Furthermore, it was suggested to introduce a semi-analytical solver to the MultipleScales code, in order to find solutions for problems that the closed-form solutions are hard to find.

## **2.4 Term-tracking methods**

The essentiality for evaluation of the term-tracking method was originally demonstrated by Cartmell and Forehand [38]. In this study the principal effects of assumptions and simplifications made during a solution procedure for reduced order models were investigated. The information loss within an approximate analytical method is shown to be noticeable, and it is evident that it can be attributed to the solution evolution process by the analysis of inherent issues. In this paper, the assumptions and simplifications have been made within ten sample problems in engineering dynamics were investigated. The results of this study are summarised in Table 2-1. For example, the assumption that there is a mass-less element in the system is used in three cases out of ten (30%).

Table 2-1 The percentage of applying nine types of assumptions in ten case studies [38].

Assumption types	Percentage of using the assumption in ten case studies
Mass-less element	30%
Maclaurin expansion	30%
Modal space	40%
Single degree of freedom	40%
Classical linear viscous damping	90%
Harmonic excitation	90%
Necessary discretisation	10%
Frequency normalised	20%
Nondimensionalised	100%

It is universally acceptable to nondimensionalise, however, frequency normalisation is not always acceptable. Generally, damping and excitation are considered linearly viscous and harmonic, but this is not necessarily always going to be a physically accurate assumption. For simplification, some elements are frequently considered to be massless, and this is generally applied to beam and mass problems where each property is visually separable, but this is actually a gross assumption that can often affect the accuracy of the solution.

As well as assumptions and simplification, the structure of the derived response is important. For example, the symbolic form of the equation of motion, part of the solution procedure and the amplitude ( $a$ ) and phase ( $\varphi$ ) response for the problem of a parametrically excited pendulum are given in Figure 2-4. It is possible to notice the highlighted symbols within the equation of motion has appeared in another form within the response. As well as this explicit connection, Cartmell and Forehand [69] showed the theoretical possibility of the existence of implicit physical-mathematical links within a dynamical modelling procedure and approximate analytical solution.

As a result, the *term-tracking* methods for the purpose of identifying, and somehow extracting, these connections have been developed. Currently two types of term-tracking method have been developed; the *Source Encoding Method (SEM)* and the *Source and Evolution Encoding Method (SEEM)*. It should be noted that the SEEM method is fully introduced in Chapter 3 of this thesis.

$$\begin{array}{l}
 \left. \begin{array}{l}
 \text{Equation of Motion} \\
 \ddot{\theta} + \frac{c}{m}\dot{\theta} + \frac{1}{l}(g + \ddot{u})\sin(\theta) = 0 \\
 \omega_0^2 = \frac{g}{l} \quad \beta = \frac{c}{2m\omega_0} \quad u(t) = ql\cos(\Omega t) \quad \sin(\theta) = \theta - \gamma\theta^3 \\
 \ddot{\theta} + 2\beta\omega_0\dot{\theta} + (\omega_0^2 - q\Omega^2\cos(\Omega t))(\theta - \gamma\theta^3) = 0 \quad \gamma = \frac{1}{6}
 \end{array} \right\} \\
 \\
 \left. \begin{array}{l}
 \text{Multiple Scales} \\
 T_j = \varepsilon^j t \quad \theta = \theta_0 + \varepsilon\theta_1 \\
 D_j = \frac{\partial}{\partial T_j} \quad \frac{d}{dt} = D_0 + \varepsilon D_1 + O(\varepsilon^2) + \dots \\
 \frac{d^2}{dt^2} = D_0^2 + 2\varepsilon D_0 D_1 + O(\varepsilon^2) + \dots
 \end{array} \right\} \\
 \\
 \left. \begin{array}{l}
 \text{Final response} \\
 a = a_0 e^{-\beta\tau_1} \\
 \varphi = \varphi_0 - \frac{3}{8}a_0^2 \frac{\gamma}{2\beta} e^{-2\beta\tau_1}
 \end{array} \right\}
 \end{array}$$

Figure 2-4 Explicit physical-mathematical links existing within a dynamical modelling procedure and approximate analytical solution

### 2.4.1 Source Encoding Method

The Source Encoding Method (SEM) has a straightforward logic which can identify and track the position of each term during the analysis, without providing any extra information about the quantities within the term [7]. The word *term* here defines a group of symbolic quantities which, when taken as a group, have physical meaning, such as inertia, stiffness, damping, or excitation. As the solution analysis proceeds the physical definition of the terms that arise tends in some cases to recede, but the concept of a mathematically grouped set of symbolised quantities tends to continue to hold true. On that basis, the SEM method uses a  $(n, sc)$  format to encode each term.  $n$  indicates the equation number,  $s$  corresponds to the position of the term within the equation right-hand side (RHS) and is shown by  $R$ , while the left-hand side (LHS) is symbolised by  $L$ , finally  $c$  demonstrates the position of the term within the corresponding side of the equation. For example,  $(10, L4)$  means the fourth term on the left-hand side of equation 10. Also, in this approach, the position (above or below the equation) means that the encoding is shown to have a specific meaning. The encoding would be displayed above a term for referencing/identification and below for presenting the history of the term. The SEM is conveniently demonstrated in [7] through an application to the equation of motion of a parametrically excited pendulum. The first few steps of this analysis with the SEM encoding information are shown in Figure 2-5.

$$\begin{aligned}
& \overset{(1,L1)}{\ddot{\theta}} + \overset{(1,L2)}{2\varepsilon\beta\dot{\theta}} + \overset{(1,L3)}{\theta} - \overset{(1,L4)}{\varepsilon q \omega^2 \theta \cos(\omega T_0)} + \overset{(1,L5)}{\varepsilon \gamma \theta^3} = 0 \quad (1) \\
& \theta(\tau, \varepsilon) = \overset{(2,R1)}{\tilde{\theta}_0} + \overset{(2,R2)}{\varepsilon \tilde{\theta}_1} + O(\varepsilon^2) \quad (2) \\
& \frac{d}{dt} = \overset{(3,R1)}{\tilde{D}_0} + \overset{(3,R2)}{\varepsilon \tilde{D}_1} + O(\varepsilon^2) \quad (3) \\
& \frac{d^2}{dt^2} = \overset{(4,R1)}{\tilde{D}_0^2} + \overset{(4,R1)}{2\varepsilon \tilde{D}_0 \tilde{D}_1} + O(\varepsilon^2) \quad (4) \\
& \overset{(5,L1)}{\underbrace{D_0^2 \theta_0}_{(1,L1)(2,R1)(4,R1)}} + \overset{(5,L2)}{\underbrace{\varepsilon D_0^2 \theta_1}_{(1,L1)(2,R1)(4,R1)}} + \overset{(5,L3)}{\underbrace{2\varepsilon D_0 D_1 \theta_0}_{(1,L1)(2,R1)(4,R2)}} + \overset{(5,L4)}{\underbrace{2\varepsilon \beta D_0 \theta_0}_{(1,L2)(2,R1)(3,R1)}} + \overset{(5,L5)}{\underbrace{\theta_0}_{(1,L3)(2,R1)}} \\
& + \overset{(5,L6)}{\underbrace{\varepsilon \theta_1}_{(1,L3)(2,R2)}} - \overset{(5,L6)}{\underbrace{\varepsilon q \omega^2 \theta_0 \cos(\omega T_0)}_{(1,L4)(2,R1)}} + \overset{(5,L7)}{\underbrace{\varepsilon \gamma \theta_0^3}_{(1,L5)(2,R2)}} = 0 \quad (5)
\end{aligned}$$

Figure 2-5 The Source Encoding Method applied to the first six equation in the analysis of a parametrically excited pendulum [69].

#### 2.4.2 Computerisation of the Source Encoding Method

Due to the transparent nature of the SEM, this method can be readily automated within a computer code. Forehand and Cartmell [7] used five main functions to apply the SEM encoding to a modified version of the multiple scales solver written by Khanin, Cartmell [2].

In summary, this method defines a multiplier (an encoding vector) at the beginning of the analysis for each term. As the equation number and the position of each term changes, the encoding vector is updated. The encoding information is saved within the solution procedure structure, and it is possible for it to be displayed using the *Tooltip* function. The *Tooltip* is a built-in function in the *Mathematica* code interface; it displays encoding information as the mouse pointer moves over the term. Figure 2-6 provides an example of the encoded function whereby the mouse pointer is placed over the highlighted term and the encoding indicates that it is the third term on the left-hand side of that equation. It is also possible to present the encoding results in the form of a tree-like structure in *Mathematica* [7].

$$\theta[t] + -q \varepsilon \omega^2 \text{Cos}[t \omega] \theta[t] + \underbrace{\gamma \varepsilon \theta[t]^3}_{(4,L3)} + 2 \beta \varepsilon \theta'[t] + \theta''[t] = 0$$

Figure 2-6 An example that shows how encodings information can be displayed using the Tooltip function.



Figure 2-7 An example of showing history of a term using a tree structure in the Source Encoding Method [7].

As the analysis progresses, the information gathered within the encoding vectors becomes more detailed, and can quickly become confusing for the user, and at the same time this increase in information complexity places more demands on computation processing time. To overcome this issue a new function defined as, *unencodeEquation* is introduced to erase the encoding history after several stages. Therefore, it is possible for the user to unencode and encode equations after each step, thus preserving memory and enhancing computational efficiency.

Figure 2-8-a shows a long encoding vector that is displayed for the analysis of the parametrically excited pendulum. It is possible to use this information to identify the source of the highlighted term in the equation of motion (fifth term on the left-hand side). In Figure 2-8-b, the user then decides to unencode and encode the same equation, so the specific encoding is changed to the seventh term on the left-hand side of equation 7.

$$\begin{aligned}
 \text{(a)} \quad & \theta_0 [T_0, T_1] + -q \in \omega^2 \text{Cos} [T_0 \omega] \theta_0 [T_0, T_1] + \\
 & \gamma \in \theta_0 [T_0, T_1]^3 + \in \theta_1 [T_0, T_1] + 2 \beta \in \theta_0^{(1,0)} [T_0, T_1] + \\
 & 2 \in \theta_0^{(1,1)} [T_0, T_1] + \theta_0^{(2,0)} \left[ \underset{\text{(1,L5)(2,R1)(4,R1)}}{T_0, T_1} \right] + \in \theta_1^{(2,0)} [T_0, T_1] = \theta \\
 \\
 \text{(b)} \quad & \theta_0 [T_0, T_1] + -q \in \omega^2 \text{Cos} [T_0 \omega] \theta_0 [T_0, T_1] + \\
 & \gamma \in \theta_0 [T_0, T_1]^3 + \in \theta_1 [T_0, T_1] + 2 \beta \in \theta_0^{(1,0)} [T_0, T_1] + \\
 & 2 \in \theta_0^{(1,1)} [T_0, T_1] + \theta_0^{(2,0)} \left[ \underset{\text{(5,L7)}}{T_0, T_1} \right] + \in \theta_1^{(2,0)} [T_0, T_1] = \theta
 \end{aligned}$$

Figure 2-8 The application of *unencodeEquation* function to unencode and encode equations.

The other challenge to emerge within the computerisation of the SEM was found to be in saving the encoding information during the mathematical procedure. This is caused due to the fact that the encoding vectors are stored in the inherent structure of the equations, and so when the differential equations are subsequently solved the encodings are integrated within the terms, and this leads to a complex structure when visualised, as explained in Figure 2-9-a. To overcome

this obstacle a function is defined to remove all the unique encoding vectors that are repeated or placed in inappropriate positions (such as the denominators or indices of a power function), as shown in Figure 2-9-b.

$$(a) \quad \Theta_1[T_0, T_1] = \frac{e^{-3i T_0 - i T_1} \sigma \omega^2 \text{Conjugate}[A[1][T_1]] \text{term}[25, 2, 1]}{2 (\text{term}[25, 1, 1] - 9 \text{term}[25, 1, 2])}$$

$$(b) \quad \Theta_1[T_0, T_1] = -\frac{1}{16} e^{-3i T_0 - i T_1} \sigma \omega^2 \text{Conjugate}[A[1][T_1]] \text{term}[25, 1, 1] \text{term}[25, 1, 2] \text{term}[25, 2, 1] +$$

Figure 2-9 (a) The encoding vectors are mixed with the term during analysis, (b) Unique encoding vectors are removed (note this a partial presentations of an equation).

Therefore, it is essential to develop a new term-tracking method which is able do track each quantity as well as the terms through the solution procedure.

## 2.5 Summary

In this chapter, after a short discussion of the numerical methods, the existing gap for developing a transparent symbolic analytical solver is highlighted. Afterwards, the general concept of the perturbation method of multiple scales is discussed. This is followed by a comparison between the theoretical response and experimental investigation of a parametrically excited beam. The detail of the solution procedure and conducted experiment are provided in the appendix.

The history of the symbolic computational solvers was briefly discussed, starting from the automatic generation of the equation of motion to fully automated symbolic solution procedure. Finally, the history and reasoning for the concept of the term-tracking methods are discussed.



# Chapter 3 Symbolic Computational Dynamics Solvers

## 3.1 Introduction

A majority of currently available computational tools for modelling engineering systems are restricted in terms of user interaction where the core processor is hidden away from the analyst. The ethos behind the Symbolic Computational Dynamics (SCD) solver is to deliberately design a high degree of flexibility and control for the analyst to exploit and to apply. The usual processes of modelling nonlinear systems introduce a considerable number of *ad hoc* simplifications and assumptions, and these have to be introduced by the user on the premise of physical and mathematical acceptability and consistency. It is therefore reasonable to assume that by taking control away from the user unknown opportunities for missing information or the accumulation of unexpected errors in the calculations and final response of the system could emerge.

Figure 3-1 shows a general overview of the principle behind the SCD solver; showing how it connects the mechanical system, the solution procedure and the final response in a unique structure. SCD solvers are potentially capable of processing distinct types of mechanical systems, the requirement for the input to the system can vary and depends on the selected core solver. In this study, perturbation methods are of interest. Therefore, the application of these solvers is currently limited to weakly nonlinear systems.

At present the general input to an SCD solver is the equation(s) of motion. The structure of such a solver is based on a *term-tracker* that operates in parallel with the core solver. The term-tracker depends on its logic, adds extra information into the structure of the equations that emerge from the solution process. This extra information is saved in a special format which guarantees no interaction with the actual solution procedure itself. The complete solution procedure plus the term-tracking information (or, encodings) is presented during and after the analysis.

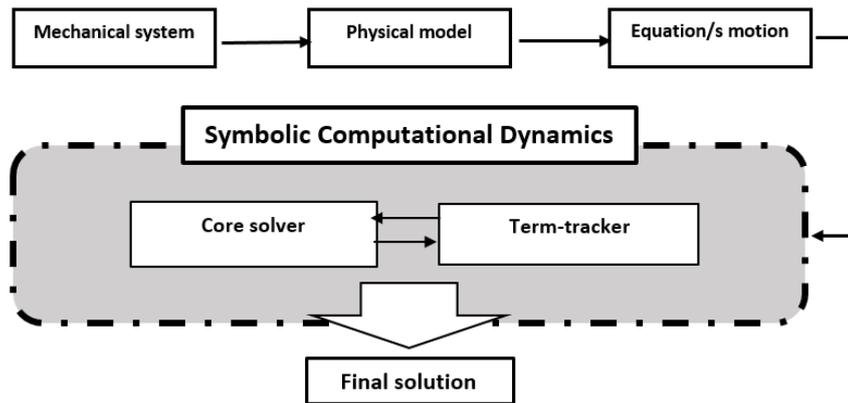


Figure 3-1 Overview of a generic Symbolic Computational Dynamic solver.

As Figure 3-1 shows a generic SCD solver currently consists of a core solver and a term-tracker, whereas the rest of the process must be done manually. It is expected that automated generation of the physical model and the equation(s) of motion will be included in SCDs in the future, Figure 3-2. Considering the solver is currently based on the general multiple scales method, dimension reduction must be done in the process of physical modelling.

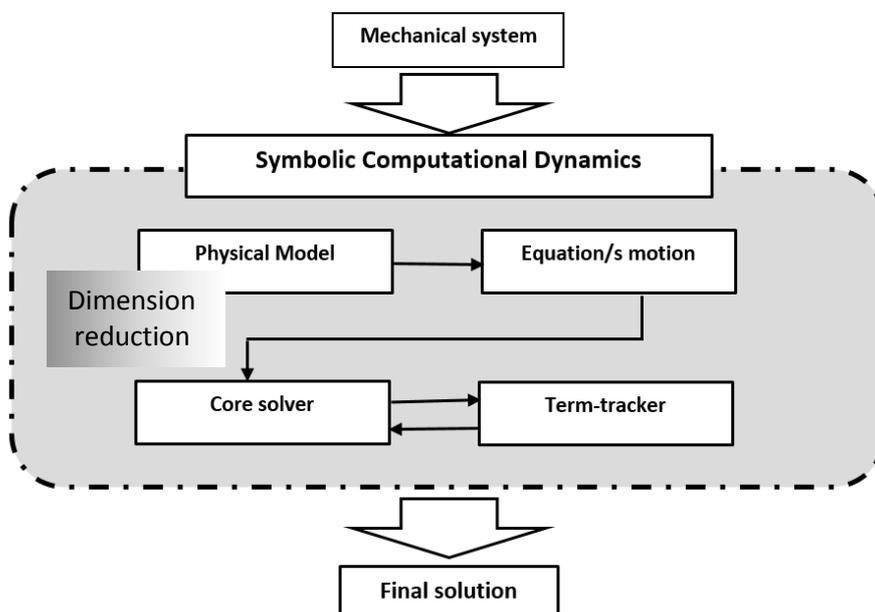


Figure 3-2 Overview of possible Symbolic Computational Dynamic solver in the future.

### 3.2 Core solver

The flexibility of an SCD solver is assured by the potential ability of the user to change the core solver. The core solver can be based on any approximate analytical method; in this study, the multiple scales method is selected. The solver functions like a white box, providing as much

information as possible. A general overview of the first few equations of the analysis of the dynamics of a parametrically excited pendulum is shown in Figure 3-3. The solution procedure which develops within an SCD solver is completely symbolic and looks essentially like a corresponding hand-written document.

$$\begin{aligned}
 \text{Eqn [1]} \quad & \gamma_{ns} \epsilon_{ns} \theta[\tau]^3 + \theta[\tau] - \omega^2 \text{Cos}[\tau \omega] q_{ex} \epsilon_{ex} \theta[\tau] + \\
 & 2_{dp} \beta_{dp} \epsilon_{dp} (\theta')[\tau] + (\theta'')[\tau] = \theta \\
 \text{Eqn [2]} \quad & \theta = \theta_0[T_0, T_1] + \epsilon_p \theta_1[T_0, T_1] \\
 \text{Eqn [3]} \quad & dt[1] = D_0 + D_1 \epsilon_p \\
 \text{Eqn [4]} \quad & dt[2] = D_0^2 + D_0 D_1 2_{dr} \epsilon_{dr} \\
 \text{Eqn [5]} \quad & \theta[\tau]^3 = \theta_0[T_0, T_1]^3 \\
 \text{Eqn [6]} \quad & D_0^2 \theta_0[T_0, T_1] + D_0 2_{dp} \beta_{dp} \epsilon_{dp} \theta_0[T_0, T_1] + \\
 & D_0 D_1 2_{dr} \epsilon_{dr} \theta_0[T_0, T_1] + \gamma_{ns} \epsilon_{ns} \theta_0[T_0, T_1]^3 + \\
 & \theta_0[T_0, T_1] - \omega^2 \text{Cos}[T_0 \omega] q_{ex} \epsilon_{ex} \theta_0[T_0, T_1] + \\
 & D_0^2 \epsilon_p \theta_1[T_0, T_1] + \epsilon_p \theta_1[T_0, T_1] = \theta \\
 \text{Eqn [7]} \quad & D_0^2 \theta_0[T_0, T_1] + \theta_0[T_0, T_1] = \theta \\
 \text{Eqn [8]} \quad & D_0^2 \theta_1[T_0, T_1] + \theta_1[T_0, T_1] = \\
 & - \frac{D_0 2_{dp} \beta_{dp} \epsilon_{dp} \theta_0[T_0, T_1]}{\epsilon_p} - \frac{D_0 D_1 2_{dr} \epsilon_{dr} \theta_0[T_0, T_1]}{\epsilon_p} - \\
 & \frac{\gamma_{ns} \epsilon_{ns} \theta_0[T_0, T_1]^3}{\epsilon_p} + \frac{\omega^2 \text{Cos}[T_0 \omega] q_{ex} \epsilon_{ex} \theta_0[T_0, T_1]}{\epsilon_p} \\
 \text{Eqn [9]} \quad & \theta_0[T_0, T_1] = A[T_1] e^{i T_0} + \bar{A}[T_1] e^{-i T_0}
 \end{aligned}$$

Figure 3-3 Appearance of the solution procedure in a Symbolic Computational Dynamics solver.

A modified version of the symbolic multiple scales solver as first introduced by Forehand and Cartmell [7] has been used in this research. The solver is semi-automated, and this provides the developer and user with the freedom to select the necessary steps of the analysis. Depending on the personal preference of the user, and also the specific nature of the mathematical problem itself, the number of output equations formally within a standard perturbation analysis can differ from problem to problem. It can be very detailed, thereby defining a single equation for each minor step of the analysis, or concise; skipping some unnecessary steps in the output (noting that they will still be generated internally and processed accordingly). This means that the information created by the term-tracker (the encodings) is unique to the solution procedure, and the encodings must always be presented with the details of the solution procedure. For example, when considering Figure 3-3, it is possible to avoid stating equation (5) and instead substitute the result directly in equation (6).

### 3.3 The Source Evolution and Encoding Method

It has been shown that the term-tracker has to run in parallel with the core solver in order to create the extra information that is sought. This information assists in identifying the important stages of analysis for each quantity. In this research the Source Evolution and Encoding Method (SEEM) [69] term-tracker is developed and discussed in detail. The purpose of the SEEM is to highlight the contribution of each specific symbolic quantity to the analysis by extracting relevant information. The information that is gathered for each quantity in the analysis is called its *encoding*.

The encodings are defined when the quantity is firstly introduced to the analysis and they evolve during the solution procedure. Therefore, the details of the encodings for each quantity are adjustable and mainly depend on the encoding level. The encodings are displayed inside a bracket and each encoding element is separated with a comma, and the syntactical structure of this is called the *encoding vector*. Currently, the SEEM is categorised into the four encoding levels and the encoding logic is summarised in Table 3-1.

Table 3-1 The Source Evolution and Encoding method logic summary.

Encoding level	Description	Display
<b>First level</b>	Origin of the equation and the small parameter order ( $\varepsilon$ )	$(n_1, n_2)$
<b>Second level</b>	Equation number in which the quantity becomes explicit	$(n_1, n_2, n_3)$
<b>Third level</b>	Equation number in which the explicit form is modified	$(n_1, n_2, n_3)$
<b>Compound</b>	Simplification of exponential functions for the first time	$(* , n_{latest})$
	Simplification of exponential functions for the second time	$(# , n_{latest})$
	Simplification of exponential functions for the third time	$(\$ , n_{latest})$
	Simplification of exponential functions for the fourth time	$(\pounds , n_{latest})$
	Simplification of exponential functions for the fifth time	$(** , n_{latest})$

The **first level** of encoding is applied when a quantity appears for the first time in the analysis and no encoding vector is allocated to it. The encoding vector is defined at this step and it always contains of two digits  $(n_1, n_2)$ ;  $n_1$  is the equation number, at which the quantity is introduced for the first time, and  $n_2$  is the order of the small parameter ( $\varepsilon$ ). Despite other encoding levels, all the quantities in the analysis should have been assigned the first level of encoding at some point of the analysis. Quantities have to pass the first level in order to be considered for the other levels of encoding. The first level of encoding works as an identifier which helps to quickly classify both the origin and significance of the quantity.

The **second level** of encoding is only considered for the quantities that have been introduced before (those that have already passed through the first level of encoding) and which have reappeared in an explicit-form equation for the first time. The explicit equation number ( $n_3$ ) is added into the encoding vectors for all the quantities, ( $n_1, n_2, n_3$ ). The explicit equations that emerge within the early stages of the approximate analytical solution method play a major role in the final solution. For example considering the symbolic solution for the zeroth-order perturbation equation, it is essential for the user to be able to identify the history of each quantity as it directly connects to the physical system.

In order to collect more information about the quantities that are participating in an explicit-form equation, a **third level** of encoding can be defined. The third level of encoding is added, where a quantity with a second level of encoding is modified from its original form. It is necessary to follow a quantity with a second level of encoding, and track the changes in the solution procedure. The way to do this is to add a fourth encoding element to the encoding vector ( $n_1, n_2, n_3, n_4$ ), indicating exactly where the particular quantity is significantly modified.

The main policy of SEEM is to avoid any possible numerical or algebraic cancellations as a way of preventing information loss. Having said this it is understood that keeping all the quantities unmodified in their original form can make equations appear to be unnecessarily complicated. Therefore, the **compound level** of encoding is defined, and in practice the use of this has mainly focused on handling the exponential functions that routinely arise in the algebra of the multiple scales method, and where  $e$  is the base of the natural logarithm and  $i$  the imaginary number within the exponent.

Expression (3.1) is an arbitrary demonstration for the application of the compound level encoding. The syntax \* is used to show that this simplification is done for the first time. It is assumed that two exponential quantities, with different mathematical-physical origins, are subsumed together in an arbitrary equation, with number  $z$ . Considering the right-hand side (RHS) of this expression, one can easily see that this term is created by subsuming two exponential terms created from two arbitrary equations  $x$  and  $y$  of the preceding analysis. If the exponential term is subsequently subsumed within other exponential terms for a second, third, fourth and fifth time then the #, \$, £, and \*\* syntactical symbols are used, respectively, to denote that this further compounding has happened.

$$\underbrace{e}_{(x,0,21)} \underbrace{(\tilde{i} T_0)}^{(x,0,21)} * \underbrace{e}_{(y,1,11)} \underbrace{(\tilde{i}\omega T_0)}^{(y,1,11)} \rightarrow \underbrace{e}_{(*,Z)} \underbrace{\tilde{i} \left( \underbrace{1}_{(x,0,21)} + \underbrace{\omega}_{(y,0,11)} \right) T_0}_{(*,Z)} \tag{3.1}$$

**3.3.1 Special considerations**

It is important to note that from a consistent operational and logistical point of view encoding information should be applied to all the quantities in a perturbation analysis, but with the exception of time and the dependent variables (before introducing the perturbation expansion).

The reason for this is that the dependent variables are necessarily restructured within the required form of the perturbation equations, and so each correction term has to be addressed to its perturbational source rather than to the main dependent variable(s) in the equation of motion. The encoding of time doesn't have a particular meaning in context because its definition remains unchanged from equation to equation, despite the use of time scales within the analysis. The effects of time-scaling are accommodated automatically within the structured mathematics of the perturbation method itself.

A vehicle for discussion of the SEEM methodology in practice is provided next in the form of the detailed analysis of vibration of a parametrically excited pendulum, close to principal parametric resonance. This is a single degree of freedom problem, therefore with one generalised coordinate in the Lagrangian sense, and characterised by the presence of a time variant stiffness term, in conjunction with the usual physically based terms representing inertia, damping, and stiffness.

**3.4 Parametric excitation of a pendulum**

A simplified physical model of an engineering system, a parametrically excited pendulum is shown in Figure 3-4. To define the equation of motion of this system, each component of the system is defined by a unique symbol:

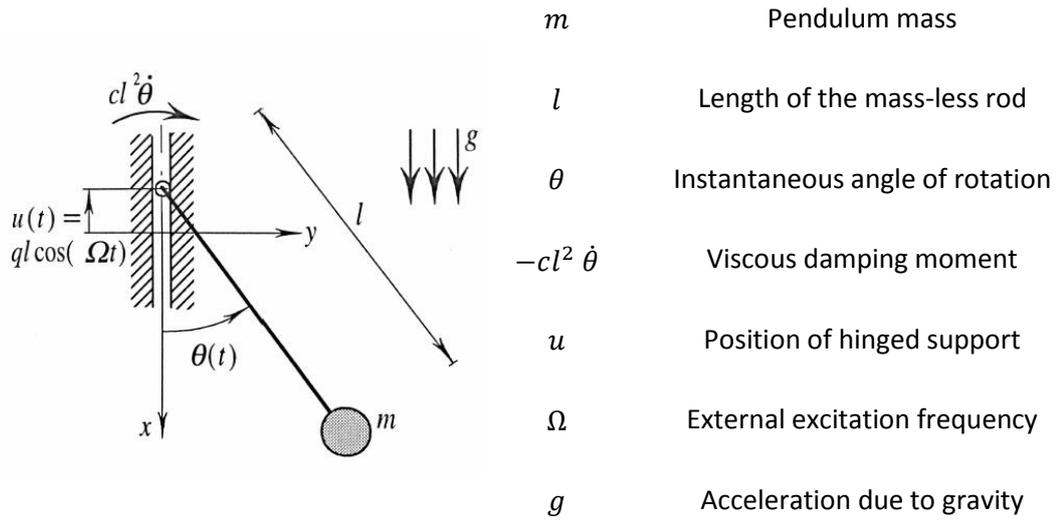


Figure 3-4 Configuration and symbols used to describe the problem of a parametrically excited pendulum [15].

### 3.4.1 Physical modelling

The kinetic and potential energy of the system is given in equations (3.2) and (3.3), respectively.

$$T = \frac{1}{2}m(\dot{u}^2 + 2l\dot{\theta}\sin(\theta) + l^2\dot{\theta}^2) \quad (3.2)$$

$$V = -mg(l\cos(\theta) - u) \quad (3.3)$$

The equation of motion (3.4) is then derived using Lagrange's equation for a single-DOF non-conservative system [39].

$$\ddot{\theta} + \frac{c}{m}\dot{\theta} + \frac{1}{l}(g + \ddot{u})\sin(\theta) = 0 \quad (3.4)$$

The linear natural frequency ( $\omega_0$ ) of the system is equal to  $\sqrt{g/l}$ , damping ratio ( $\beta$ ) is defined as  $c/(2m\omega_0)$ , and the external excitation ( $u$ ) is expressed as  $ql\cos(\Omega t)$ . The standard form of the equation of motion is given below:

$$\ddot{\theta} + 2\beta\omega_0\dot{\theta} + (\omega_0^2 - q\Omega^2\cos(\Omega t))\sin(\theta) = 0 \quad (3.5)$$

The pendulum is excited parametrically, as the external excitation appeared as a parameter in the system (stiffness). The equation of motion is nonlinear because of the  $\sin(\theta)$  term. To solve this equation, it is assumed that the rotation of the system is finite and limited, so a Taylor

expansion up to the second term (3.6) can represent  $\sin(\theta)$  term in this equation. Then the nondimensionalised form of the real time ( $t$ ) and the excitation frequency ( $\Omega$ ) are given (3.7). Substituting equations (3.6) and into equation (3.5), results equation (3.8).

$$\sin(\theta) \cong \theta - \frac{1}{6}\theta^3 \quad (3.6)$$

$$t = \omega_0\tau, \omega = \Omega/\omega_0 \quad (3.7)$$

$$\ddot{\theta} + 2\beta\dot{\theta} + (1 - q\omega^2\text{Cos}(\omega\tau))(\theta - \frac{1}{6}\theta^3) = 0 \quad (3.8)$$

In perturbation methods, the relative importance of each term must be considered mathematically. There is no established method for the term ordering, and the analyst must apply it based on his/her experience [70]. The linear inertia term ( $\ddot{\theta}$ ), which is fundamental to the motion of system is strong, as well as the stiffness term. Finally, the damping term ( $2\beta$ ), the external excitation amplitude term ( $q$ ) and the nonlinear term ( $1/6 \theta^3$ ) are considered to be weak and  $\varepsilon$  as a small parameter is multiplied to these terms.  $\varepsilon$  is a multiplier that used to highlight the significance of each term in an equation and it does not have any physical meaning [15]. After applying the mentioned assumptions to the equation of motion, terms with the order of  $\varepsilon^2$  and higher are removed. The final form of the equation of motion is shown in equation (3.9),  $\gamma$  is the coefficient of the nonlinear stiffness.

$$\ddot{\theta} + 2\varepsilon\beta\dot{\theta} + (1 - \varepsilon q\omega^2\text{Cos}(\omega\tau))\theta - \varepsilon\gamma\theta^3 = 0 \quad (3.9)$$

By applying such assumptions systematically up to the point at which an ordered differential equation of motion, or set of equations of motion, is/are defined means that up to this stage of analysis, a considerable number of assumptions has been made and the term-tracking method could assist the user to track each assumption through the solution procedure. These assumptions are considered as *modelling related assumptions* and they are distinguished from the *solution process related assumptions*.

### 3.4.2 Solution procedure

At this stage of the analysis the method of multiple scales can start to be applied to the equation of motion (3.9). For consistency, the equation number tag has to be changed from the thesis format (i.e. (3.9)) to (1) to reflect the logical structure of the SEEM as would be applied to such a problem. The first level of the SEEM encoding is applied to the equation of motion (1), the

equation number and the order of each term are both shown in a bracket below each term. It can be seen that no encoding vector is defined for  $T_0$  and  $\theta$ , for the reasons discussed above.

Then it is assumed that the approximate solution containing the zeroth and first order perturbation solutions can satisfactorily predict the response of the pendulum accurately. A perturbation expansion, which is conventionally assumed to be uniformly valid, is introduced and encoded in equation (2). The fast and slow independent time scales are defined as  $T_0 = \tau$  and  $T_1 = \varepsilon \tau$ , respectively.

$$\ddot{\theta} + \underbrace{2\varepsilon\beta}_{(1,1)} \dot{\theta} + \left( \underbrace{1}_{(1,0)} - \underbrace{\varepsilon q \omega^2}_{(1,1)} \cos(\underbrace{\omega}_{(1,0)} T_0) \right) \theta + \underbrace{\varepsilon \gamma}_{(1,1)} \theta^3 = 0 \quad (1)$$

$$\theta(\tau, \varepsilon) = \underbrace{\theta_0}_{(2,0)} + \underbrace{\varepsilon \theta_1}_{(2,1)} + O(\varepsilon^2) \quad (2)$$

Then the first and second derivatives up to first-order perturbation are given and encoded in (3) and (4), respectively (noting that the D-Operator notation is used to define the partial derivatives:  $D_i = \partial/\partial T_i$ ).

$$\frac{d}{dt} = \underbrace{D_0}_{(3,0)} + \underbrace{\varepsilon D_1}_{(3,1)} + O(\varepsilon^2) \quad (3)$$

$$\frac{d^2}{dt^2} = \underbrace{D_0^2}_{(4,0)} + \underbrace{2\varepsilon D_0 D_1}_{(4,1)} + O(\varepsilon^2) \quad (4)$$

Equation (5) defines the cubic form of the generalised coordinate  $\theta$  based on the perturbation expansion up to the first order of  $\varepsilon$ . Since the cubic form now appears in explicit-form the equation number is added as the third digit of the encoding. The encoding vector for this term is shown as (2,0,5). This information can help the user to identify the source of each quantity, and the stage that the quantity is explicitly defined as:

$$\theta^3 = \underbrace{\theta_0^3}_{(2,0,5)} + O(\varepsilon). \quad (5)$$

The next step is to substitute equations (2), (3), (4), and (5) into equation (1), and then the higher order terms can be removed, resulting in:

$$\begin{aligned} \underbrace{D_0^2}_{(4,0)} \underbrace{\theta_0}_{(2,0)} + \underbrace{\varepsilon}_{(2,1)} \underbrace{D_0^2}_{(4,0)} \underbrace{\theta_1}_{(2,1)} + \underbrace{2\varepsilon D_0 D_1}_{(4,1)} \underbrace{\theta_0}_{(2,0)} + \underbrace{2\varepsilon\beta}_{(1,1)} \underbrace{D_0}_{(3,0)} \underbrace{\theta_0}_{(2,0)} + \underbrace{\theta_0}_{(2,0)} + \underbrace{\varepsilon}_{(2,1)} \underbrace{\theta_1}_{(2,1)} \\ - \underbrace{\varepsilon q \omega^2}_{(1,1)} \underbrace{\theta_0}_{(2,0)} \cos(\underbrace{\omega}_{(1,0)} T_0) + \underbrace{\varepsilon \gamma}_{(1,1)} \underbrace{\theta_0^3}_{(2,0)} = 0 \end{aligned} \quad (6)$$

The SEEM created a considerable amount of information in this early stage. For example, it is possible to distinguish between  $\underbrace{\varepsilon}_{(1,1)}$ , that is introduced in the modelling stage and  $\underbrace{\varepsilon}_{(2,1)}$  which is introduced in the solution procedure, both  $\varepsilon$ s are small but not necessarily equal.

### Perturbation equations

The zeroth-order perturbation equation (7) is obtained by taking the terms of order  $\varepsilon^0$  out from equation (6) and then setting them to zero. The same action is taken for all terms to  $\varepsilon^1$  in order to structure the first-order perturbation equation (8). From this conventional step, the two necessary perturbation equations can be constructed. When taking the right-hand side (RHS) of equation (8), it can be seen that this is the first time in the analysis that the SEEM encoding method prevents the cancellation of divisor terms; this is because the origins of terms are different and so in the SEEM context they are definitionally different. Also, there is no sign of equation number (8) within the encoding vectors, as no particular modification has yet been carried out at that point.

$$\underbrace{D_0^2}_{(4,0)} \underbrace{\theta_0}_{(2,0)} + \underbrace{\theta_0}_{(2,0)} = 0 \quad (7)$$

$$\begin{aligned} \underbrace{D_0^2}_{(4,0)} \underbrace{\theta_1}_{(2,1)} + \underbrace{\theta_1}_{(2,1)} = - \frac{\underbrace{\varepsilon}_{(4,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{2D_0 D_1}_{(4,1)} \underbrace{\theta_0}_{(2,0)} - \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{2\beta}_{(1,1)} \underbrace{D_0}_{(3,0)} \underbrace{\theta_0}_{(2,0)} + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\gamma}_{(1,1)} \underbrace{\theta_0^3}_{(2,0,5)} \\ + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{q \omega^2}_{(1,1)} \underbrace{\theta_0}_{(2,0)} \cos(\underbrace{\omega}_{(1,0)} T_0) \end{aligned} \quad (8)$$

The general harmonic solution for the zeroth-order perturbation equation is routinely obtained and is shown in equation (9) in exponential form, in keeping with the usually accepted notational formalities of the method. In this equation  $A$  is an as-yet arbitrary and complex function of the slow time scale  $T_1$ , and  $\bar{A}$  represents its complex conjugate. The third encoding digit is added to the encoding vectors of all the quantities on the RHS, as  $\theta_0$  is defined explicitly in this equation. The origin of complex amplitude  $A$  origin is referred back to equation (2) because it physically relates to the  $\underbrace{\theta_0}_{(2,0)}$  in equation (7).

$$\theta_0 = \underbrace{A}_{(2,0)} e^{\underbrace{i}_{(9,0)} \underbrace{T_0}_{(9,0)}} + \underbrace{\bar{A}}_{(2,0,9)} e^{\underbrace{-i}_{(9,0)} \underbrace{T_0}_{(9,0)}} \quad (9)$$

Then the solution of the zeroth-order perturbation equation is substituted into the cubic form of  $\theta_0$ , resulting:

$$\begin{aligned} \theta_0^3 = & \underbrace{A^3}_{(2,0,9,10)} e^{\underbrace{i}_{(9,0)} \underbrace{3}_{(1,0,10)} \underbrace{T_0}_{(9,0)}} + \underbrace{\bar{A}^3}_{(2,0,9,10)} e^{\underbrace{-i}_{(9,0)} \underbrace{3}_{(1,0,10)} \underbrace{T_0}_{(9,0)}} \\ & + \underbrace{3}_{(10,0)} \underbrace{A^2}_{(2,0,9,10)} \underbrace{\bar{A}}_{(2,0,9,10)} e^{\underbrace{i}_{(9,0)} \underbrace{T_0}_{(9,0)}} \\ & + \underbrace{3}_{(10,0)} \underbrace{A}_{(2,0,9,10)} \underbrace{\bar{A}^2}_{(2,0,9,10)} e^{\underbrace{-i}_{(9,0)} \underbrace{T_0}_{(9,0)}} \end{aligned} \quad (10)$$

Equation (10) is an example of how the logic behind the SEEM starts to become complicated. If one looks closely it can be seen that there are two types of number 3, with the same numerical value but dissimilar encoding sources. The first one is the number 3 within the index of the exponential function, and the encoding refers it back to equation (1). The second one is the number 3 in the  $3A^2\bar{A}e^{iT_0}$  and  $3A\bar{A}^2e^{-iT_0}$  terms, which are defined because of the algebraic procedure in this equation structure,  $(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$ . Applying the SEEM to this equation has clearly addressed the origin of each quantity to its physical source.

The third level of the SEEM is used for the first time in this analysis. The encodings of  $A$  and  $\bar{A}$  are defined as (2,0,9,10), which clearly shows that the origin of this zeroth-order term is equation (2), that it has appeared in an explicit equation structure in equation (9) and finally that its structure has been modified in equation (10).

Then the exponential form of the harmonic function within the external excitation term in equation (1) is given as:

$$\cos(\omega T_0) = \frac{1}{2} \left( e^{\underbrace{i}_{(1,0,11)} \underbrace{\omega}_{(1,0,11)} \underbrace{T_0}_{(1,0,11)}} + e^{\underbrace{-i}_{(1,0,11)} \underbrace{\omega}_{(1,0,11)} \underbrace{T_0}_{(1,0,11)}} \right) \quad (11)$$

The solution of the zeroth-order perturbation equation (9) and the cubic form of  $\theta_0$  (5) are substituted into the first order perturbation equation (8), and the result is shown in equation (12).

$$\begin{aligned}
\underline{D}_0^2 \underline{\theta}_1 + \underline{\theta}_1 &= - \underbrace{2}_{(4,0)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{D}_1}_{(4,1)} \underbrace{\underline{A}}_{(2,0,9)} \underbrace{i}_{(9,0)} \underbrace{e}_{(9,0)}^{i T_0} \\
&\quad - \underbrace{2}_{(4,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{D}_1}_{(4,1)} \underbrace{\bar{\underline{A}}}_{(2,0,9)} \underbrace{-i}_{(9,0)} \underbrace{e}_{(9,0)}^{-i T_0} \\
&\quad - \underbrace{2}_{(1,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\beta}}_{(1,1)} \underbrace{\underline{A}}_{(2,0,9)} \underbrace{i}_{(9,0)} \underbrace{e}_{(9,0)}^{i T_0} \\
&\quad - \underbrace{2}_{(1,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\beta}}_{(1,1)} \underbrace{\bar{\underline{A}}}_{(2,0,9)} \underbrace{-i}_{(9,0)} \underbrace{e}_{(9,0)}^{-i T_0} \\
&\quad - \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\gamma}}_{(1,1)} \underbrace{\underline{A}^3}_{(2,0,9,10)} \underbrace{e}_{(9,0)}^{i \underbrace{3}_{(1,0,10)} T_0} \\
&\quad - \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\gamma}}_{(1,1)} \underbrace{\bar{\underline{A}}^3}_{(2,0,9,10)} \underbrace{e}_{(9,0)}^{-i \underbrace{3}_{(1,0,10)} T_0} \\
&\quad - \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\gamma}}_{(1,1)} \underbrace{\underbrace{3}_{(10,0)}} \underbrace{\underline{A}^2}_{(2,0,9,10)} \underbrace{\bar{\underline{A}}}_{(2,0,9,10)} \underbrace{e}_{(9,0)}^{i T_0} \\
&\quad - \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{\underline{\gamma}}_{(1,1)} \underbrace{\underbrace{3}_{(10,0)}} \underbrace{\underline{A}}_{(2,0,9,10)} \underbrace{\bar{\underline{A}}^2}_{(2,0,9,10)} \underbrace{e}_{(9,0)}^{-i T_0} \\
&\quad + \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{2}} \underbrace{\underline{A}}_{(1,0,11)} \underbrace{e}_{(2,0,9) (*,12)}^{i \left( \underbrace{\omega}_{(1,0,11)} + \frac{1}{\underline{9,0}} \right) T_0} \\
&\quad + \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{2}} \underbrace{\underline{A}}_{(1,0,11)} \underbrace{e}_{(2,0,9) (*,12)}^{-i \left( \underbrace{\omega}_{(1,0,11)} + \frac{-1}{\underline{9,0}} \right) T_0} \\
&\quad + \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{2}} \underbrace{\bar{\underline{A}}}_{(1,0,11)} \underbrace{e}_{(2,0,9) (*,12)}^{-i \left( \underbrace{\omega}_{(1,0,11)} + \frac{1}{\underline{9,0}} \right) T_0} \\
&\quad + \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{2}} \underbrace{\bar{\underline{A}}}_{(1,0,11)} \underbrace{e}_{(2,0,9) (*,12)}^{i \left( \underbrace{\omega}_{(1,0,11)} + \frac{-1}{\underline{9,0}} \right) T_0}
\end{aligned} \tag{12}$$

The information added by implementation of the SEEM has created a large amount of new structure to each term. From this process, it is now possible to track each quantity to its origin and precisely identify stages at which the quantities have been modified.

The cancellation of small parameter ‘epsilon divisor terms’ in this equation is avoided because of the differences emergent within the encoding information. In a standard multiple scales analysis, there is no difference between the small parameter  $\underline{\underline{\epsilon}}_{(1,1)}$  which is introduced in the equation of the motion and the small parameter  $\underline{\underline{\epsilon}}_{(2,1)}$  introduced within the perturbation expansion, as so as the results they would usually be cancelled out. The small parameter ‘epsilon’ is originally defined as arbitrarily small, but it doesn’t necessarily mean that it is always equal to a unique small numerical value. Therefore,  $\frac{\underline{\underline{\epsilon}}_{(1,1)}}{\underline{\underline{\epsilon}}_{(2,1)}}$  in this equation is not definitionally equal to 1. However, there can be exceptions to this, and one case in point is where it is clear that the numerical value of the  $\underline{\underline{\epsilon}}_{(2,1)}$  which is introduced within the perturbation expansion  $\underline{\underline{\epsilon}}_{(4,1)}$  equal to the  $\underline{\underline{\epsilon}}_{(4,1)}$  because they both are introduced into the analysis based on the perturbation expansion, but in different equations. However, cancellation is still avoided so as to retain as much information as possible.

The SEEM signifies the distinctions between the values of  $\underline{\underline{\omega}}_{(1,1)}$ ,  $\underline{\underline{\omega}}_{(4,1)}$ , and  $\underline{\underline{\omega}}_{(1,0,11)}$  which are, respectively, related directly to the damping term, the second derivative definition after introducing the perturbation expansion, and the exponential form of the external excitation term.

Moreover, the application of the compound level of the SEEM is used in this equation for the first time. For example,  $\underline{\underline{e}}_{(*,12)}^{i \left( \underline{\underline{\omega}}_{(1,0,11)} + \frac{1}{\underline{\underline{\omega}}_{(9,0)}} \right) T_0}$  means  $\underline{\underline{e}}_{(9,0)}^{i T_0}$  is subsumed into  $\underline{\underline{e}}_{(1,0,11)}^{i \underline{\underline{\omega}}_{(1,0,11)} T_0}$  at equation (12) for the first time (therefore showing up as an operation defined by \*). At a result, applying the compound level of the SEEM helps to simplify the structure of the equation, without losing any fundamental encoding information.

### Solvability conditions

The perturbation expansion in the multiple scales method must be uniformly valid, therefore,  $\theta_0$  must always be bigger than  $\theta_1$ . Terms that cause  $\theta_1$  to grow too large too quickly, and therefore which could threaten this uniformity, are called *secular terms*. Terms containing the natural frequency of the homogenous system are resonant and so when they appear in the exponents of the complex frequencies their sum must be zero in order to ensure that a valid perturbation expansion is maintained. In this case, the natural frequency of the homogeneous

system is equal to one, therefore, terms containing  $e^{\frac{i}{\omega} T_0}$  are removed and set to zero. We

note that at each level of perturbation the method requires that *secular* terms are routinely processed off-line in order to obtain information about amplitude and phase at that level of perturbation.

There are some terms in equation (12) that can be resonant depending on the value of  $\omega$ , which is used here to represent the external excitation frequency. If  $\omega$  is set to the numerical value of

$2, \frac{\varepsilon}{\omega} q \omega^2$  and its complex conjugate are then considered

to be *secular*.

As a results the *principal parametric resonance* condition is shown in equation (13). The *near-resonance* condition is accommodated when a system is excited near to a resonance condition such as this. To formulate the near-resonance condition of (14), the detuning parameter (which offers a small latitude around the resonant point) is add to equation (13) as shown.

$$\omega = \frac{2}{(13,0)} \quad (13)$$

$$\omega = \frac{2}{(13,0)} + \frac{\varepsilon}{(14,1)} \frac{\sigma}{(14,1)} \quad (14)$$

### Near-resonance case

For determining the near-resonance solution, equation (14) must be substituted into first order perturbation equation (12). In a standard multiple scale analysis, all the ‘epsilons’ are equal and therefore, whenever it is required, the ‘epsilons’ can be replaced with the definition of the perturbation parameter, equation (15). However, according to the SEEM concept, the ‘epsilons’ in an analysis are not necessarily equal and they must not cancel out. These two statements lead to an interesting observation, which is summarised below:

By considering  $e_{(\ast,12)}^{i_{(1,0,11)}(\omega + \frac{-1}{9,0})T_0}$  and then applying the near-resonance condition clearly

leads to  $e_{(\ast,12)}^{i_{(13,0)}(\frac{2}{14,1} + \frac{\sigma}{9,0} + \frac{-1}{9,0})T_0}$ . In order to guarantee a valid solution,  $\varepsilon_{(14,1)}$  must be

equal to  $T_1/T_0$ , resulting in  $e_{(\ast,12)}^{i_{(13,0)}(\frac{2}{14,1} T_0 + \frac{\sigma}{14,1} T_1 + \frac{-1}{9,0} T_0)}$ . This provides evidence to show that

the small parameter in the detuning parameter ( $\varepsilon_{(14,1)}$ ) must be equal to the perturbation

expansion ( $\varepsilon_{(2,1)}$ ), so it can remove the explicit presence of  $T_0$  in the index of the exponential

function.

This leads to a noteworthy observation that even in the standard multiple scales method the ‘epsilons’ should not really be considered to be identical, and it is shown that by using the SEEM carefully in this way distinctions between ‘epsilons’ can be made, and then all ‘epsilons’ should be retained when it is useful to do so, but not at other times.

$$\varepsilon_{(2,1)} = T_1/T_0 \quad (15)$$

The first order perturbation equation for the near-resonant case is provided in equation (16). As the analysis progresses the encoding information in each term becomes more diverse. In the initial stages of the analysis most of the terms had almost the same origin and order. In a standard multiple scales method, without applying the SEEM, it is virtually impossible for the user to identify the source of each quantity, for example those quantities within  $Ae^{(3T_0 + \sigma T_1)}$  which appears in equation (16). Applying the SEEM method has restructured that term to

$A_{(2,0,9)} e_{(\#16)}^{i_{(13,0)}(\frac{2}{14,1} T_0 + \frac{\sigma}{14,1} T_1 + \frac{-1}{9,0} T_0)}$ .  $A$  is the complex amplitude of the zeroth-order

perturbation part of the response and it is shown in this form in equation (9). The # within the encodings of  $e$  and  $i$ , are showing that the index is simplified for the second time. The summation of  $\frac{2}{(13,0)} T_0 + \frac{1}{(9,0)} T_0$  in the index shows how the external excitation relates to the

homogeneous natural frequency. Without the SEEM method this summation was simply equal to 3, a nondimensionalised frequency in the form of a numerical number without any particular physical definition.

$$\begin{aligned}
\frac{D_0^2}{(4,0)} \theta_1 + \theta_1 = & - \frac{2}{(4,1)} \frac{\underline{\varepsilon}}{(2,1)} \underline{D_1} \underline{A} \underline{i} \underline{e} \begin{matrix} i T_0 \\ (9,0) \end{matrix} - \frac{2}{(4,1)} \frac{\underline{\varepsilon}}{(2,1)} \underline{D_1} \underline{\bar{A}} \underline{-i} \underline{e} \begin{matrix} -i T_0 \\ (9,0) \end{matrix} \\
& - \frac{2}{(1,1)} \frac{\underline{\varepsilon}}{(2,1)} \underline{\beta} \underline{A} \underline{i} \underline{e} \begin{matrix} i T_0 \\ (9,0) \end{matrix} - \frac{2}{(1,1)} \frac{\underline{\varepsilon}}{(2,1)} \underline{\beta} \underline{\bar{A}} \underline{-i} \underline{e} \begin{matrix} -i T_0 \\ (9,0) \end{matrix} \\
& - \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{\gamma} \underline{A^3} \underline{e} \begin{matrix} i \\ (9,0) \end{matrix} \begin{matrix} 3 \\ (1,0,10) \end{matrix} T_0 - \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{\gamma} \underline{\bar{A}^3} \underline{e} \begin{matrix} -i \\ (9,0) \end{matrix} \begin{matrix} 3 \\ (1,0,10) \end{matrix} T_0 \\
& - \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{\gamma} \begin{matrix} 3 \\ (1,1) \end{matrix} \underline{A^2} \underline{\bar{A}} \underline{e} \begin{matrix} i \\ (9,0) \end{matrix} T_0 \\
& - \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{\gamma} \begin{matrix} 3 \\ (1,1) \end{matrix} \underline{A} \underline{\bar{A}^2} \underline{e} \begin{matrix} -i \\ (9,0) \end{matrix} T_0 \\
& + \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{q\omega^2} \frac{1}{2} \underline{A} \underline{e} \begin{matrix} i \\ (\#16) \end{matrix} \begin{pmatrix} 2 T_0 + \sigma T_1 + \frac{1}{2} T_0 \\ (13,0) \quad (14,1) \quad (9,0) \end{pmatrix} \\
& + \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{q\omega^2} \frac{1}{2} \underline{A} \underline{e} \begin{matrix} -i \\ (\#16) \end{matrix} \begin{pmatrix} 2 T_0 + \sigma T_1 + \frac{-1}{2} T_0 \\ (13,0) \quad (14,1) \quad (9,0) \end{pmatrix} \\
& + \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{q\omega^2} \frac{1}{2} \underline{\bar{A}} \underline{e} \begin{matrix} -i \\ (\#16) \end{matrix} \begin{pmatrix} 2 T_0 + \sigma T_1 + \frac{1}{2} T_0 \\ (13,0) \quad (14,1) \quad (9,0) \end{pmatrix} \\
& + \frac{(1,1)}{(2,1)} \frac{\underline{\varepsilon}}{\underline{\varepsilon}} \underline{q\omega^2} \frac{1}{2} \underline{\bar{A}} \underline{e} \begin{matrix} i \\ (\#16) \end{matrix} \begin{pmatrix} 2 T_0 + \sigma T_1 + \frac{-1}{2} T_0 \\ (13,0) \quad (14,1) \quad (9,0) \end{pmatrix}
\end{aligned} \tag{16}$$

The next step is to select the *secular* terms from equation (16) and define the solvability conditions. This is done by taking out the terms containing  $e^{iT_0}$  out and setting them to zero, leading to the *solvability condition*, as given in equation (17). Having secular terms in the equation makes the first order perturbation grow too quickly and make the perturbation expansion invalid as  $\theta_1 \gg \theta_0$ .

$$\begin{aligned}
& - \underbrace{2}_{(4,1)} \frac{\underbrace{\varepsilon}_{(4,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{D_1}_{(4,1)} \underbrace{A}_{(2,0,9)} \underbrace{i}_{(9,0)} - \underbrace{2}_{(1,1)} \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\beta}_{(1,1)} \underbrace{A}_{(2,0,9)} \underbrace{i}_{(9,0)} \\
& - \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\gamma}_{(1,1)} \underbrace{3}_{(10,0)} \underbrace{A^2}_{(2,0,9,10)} \underbrace{\bar{A}}_{(2,0,9,10)} \\
& + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underbrace{2}_{(1,0,11)}} \underbrace{\bar{A}}_{(2,0,9)} \underbrace{e}_{(\$,16)} \underbrace{i}_{(\$,16)} \left( \underbrace{2}_{(13,0)} T_0 + \underbrace{\sigma}_{(14,1)} T_1 + \underbrace{-2}_{(9,0)} T_0 \right) = 0
\end{aligned} \tag{17}$$

The SEEM has made it simple for the user to detect the changes from the last stage quickly. The \$ character in the encoding index of the last term shows that this term has been simplified for the third time, by dividing the whole term by  $e^{iT_0}$ . Also, there are three different number 2s in this equation;  $\underbrace{2}_{(1,0,11)}$  relates to the external excitation frequency,  $\underbrace{2}_{(13,0)}$  is the nondimensionalised principal parametric resonance frequency, and  $\underbrace{2}_{(9,0)}$  relates to the nondimensionalised system natural frequency.

In this initial proposal for the SEEM most cancellations and simplifications must be avoided. In order to follow this requirement dividing the LHS of equation (17) by  $e^{iT_0}$  obviously results in a very complex equation form (3.10). However, in cases where the encodings are exactly the same in divisor terms it is then advantageous to perform the cancellation and simplify the expression, as nothing is necessarily lost.

$$\begin{aligned}
& - \frac{\underline{\underline{\varepsilon}}_{(4,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \frac{\underline{\underline{D}}_1}{(4,1)} \frac{\underline{\underline{A}}}{(2,0,9)} \frac{i}{(9,0)} \frac{e_{(9,0)}^{i T_0}}{e_{(17,0)}^{i T_0}} \\
& - \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \beta \frac{\underline{\underline{A}}}{(1,1)} \frac{i}{(2,0,9)} \frac{e_{(9,0)}^{i T_0}}{e_{(17,0)}^{i T_0}} \\
& - \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \gamma \frac{\underline{\underline{\beta}}}{(1,1)} \frac{\underline{\underline{A}}^2}{(10,0)} \frac{\underline{\underline{A}}}{(2,0,9,10)} \frac{e_{(9,0)}^{i T_0}}{e_{(17,0)}^{i T_0}} \\
& + \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{\underline{\omega}}_{(1,0,11)}} \frac{\underline{\underline{A}}}{(2,0,9)} e_{(\$,16)}^{i} \left( \frac{\underline{\underline{\omega}}_{(13,0)} T_0 + \underline{\underline{\sigma}}_{(14,1)} T_1 + \underline{\underline{\omega}}_{(9,0)} T_0 - \underline{\underline{\omega}}_{(17,0)} T_0 \right) \\
& = 0
\end{aligned} \tag{3.10}$$

Removing the secular terms and their complex conjugates, yields equation (18):

$$\begin{aligned}
\frac{\underline{\underline{D}}_0^2}{(4,0)} \frac{\underline{\underline{\theta}}_1}{(2,1)} + \underline{\underline{\theta}}_1 &= - \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \gamma \frac{\underline{\underline{A}}^3}{(1,1)} \frac{e_{(9,0)}^{i} \underline{\underline{\beta}}^3 T_0}{(9,0)} \\
& - \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \gamma \frac{\underline{\underline{A}}^3}{(1,1)} \frac{e_{(9,0)}^{-i} \underline{\underline{\beta}}^3 T_0}{(9,0)} \\
& + \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{\underline{\omega}}_{(1,0,11)}} \frac{\underline{\underline{A}}}{(2,0,9)} e_{(\$,16)}^{i} \left( \frac{\underline{\underline{\omega}}_{(13,0)} T_0 + \underline{\underline{\sigma}}_{(14,1)} T_1 + \underline{\underline{\omega}}_{(9,0)} T_0 \right) \\
& + \frac{\underline{\underline{\varepsilon}}_{(1,1)}}{\underline{\underline{\varepsilon}}_{(2,1)}} \underbrace{q\omega^2}_{(1,1)} \frac{1}{\underline{\underline{\omega}}_{(1,0,11)}} \frac{\underline{\underline{A}}}{(2,0,9)} e_{(\$,16)}^{-i} \left( \frac{\underline{\underline{\omega}}_{(13,0)} T_0 + \underline{\underline{\sigma}}_{(14,1)} T_1 + \underline{\underline{\omega}}_{(9,0)} T_0 \right)
\end{aligned} \tag{18}$$

The particular solution for this first order perturbation equation is given in equation (19). As  $\theta_1$  is expressed explicitly in this equation, the third encoding element is added into the encoding vector for each quantity.

$$\begin{aligned}
& \theta_1 \\
& \underline{(2,1)} \\
& = - \frac{\underline{(1,1,19)} \underline{\varepsilon}}{\underline{\varepsilon} \underline{(2,1,19)}} \frac{1}{\underline{\underline{g}} \underline{(19,0)}} \underline{\gamma} \underline{(1,1,19)} \underline{A^3} \underline{(2,0,9,10)} \underline{e} \underline{(9,0,10,19)} \underline{(9,0,10,19)} \underline{T_0} \\
& - \frac{\underline{(1,1,19)} \underline{\varepsilon}}{\underline{\varepsilon} \underline{(2,1,19)}} \frac{1}{\underline{\underline{g}} \underline{(19,0)}} \underline{\gamma} \underline{(1,1,19)} \underline{\bar{A}^3} \underline{(2,0,9,10)} \underline{e} \underline{(9,0,10,19)} \underline{(9,0,10,19)} \underline{T_0} \\
& + \frac{\underline{(1,1,19)} \underline{\varepsilon}}{\underline{\varepsilon} \underline{(2,1,19)}} \underline{q\omega^2} \underline{(1,1,19)} \frac{1}{\underline{\underline{z}} \underline{(1,0,11,19)}} \frac{1}{\underline{\underline{g}} \underline{(19,0)}} \underline{A} \underline{(2,0,9)} \underline{e} \underline{(\$,16)} \underline{(9,0,10,19)} \underline{T_0} + \underline{\sigma} \underline{(14,1,19)} \underline{T_1} + \underline{1} \underline{(9,0,19)} \underline{T_0} \\
& + \frac{\underline{(1,1,19)} \underline{\varepsilon}}{\underline{\varepsilon} \underline{(2,1,19)}} \underline{q\omega^2} \underline{(1,1,19)} \frac{1}{\underline{\underline{z}} \underline{(1,0,11,19)}} \frac{1}{\underline{\underline{g}} \underline{(19,0)}} \underline{\bar{A}} \underline{(2,0,9,19)} \underline{e} \underline{(\$,16)} \underline{(9,0,10,19)} \underline{T_0} + \underline{\sigma} \underline{(14,1,19)} \underline{T_1} + \underline{1} \underline{(9,0,19)} \underline{T_0}
\end{aligned} \tag{19}$$

$A[T_1]$  is expressed in the polar form in equation (20),  $a[T_1]$  and  $\varphi[T_1]$  are the amplitude and the phase, respectively. As all the quantities are introduced in this equation, and the first level of the SEEM is applied. By that means the equation number and the order of the small parameter 'epsilon' is added into the encoding vector.

$$A[T_1] = \frac{1}{\underline{\underline{z}} \underline{(20,0)}} \underline{a[T_1]} \underline{(20,0)} e^{i\varphi[T_1]} \underline{(20,0)} \tag{20}$$

Then equation (20) is substituted into the solvability condition (17) and multiplied by  $\underline{e^{-i\varphi[T_1]}} \underline{(20,0)}$

resulting in:

$$\begin{aligned}
& - \underline{i} \underline{(9,0)} \frac{\underline{\underline{z}} \underline{(4,1)}}{\underline{\underline{z}} \underline{(20,0)}} \frac{\underline{\varepsilon} \underline{(4,1)}}{\underline{\varepsilon} \underline{(2,1)}} \underline{D_1} \underline{a[T_1]} \underline{(20,0)} - \underline{i} \underline{(9,0)} \frac{\underline{\underline{z}} \underline{(1,1)}}{\underline{\underline{z}} \underline{(20,0)}} \frac{\underline{\varepsilon} \underline{(1,1)}}{\underline{\varepsilon} \underline{(2,1)}} \underline{\beta} \underline{a[T_1]} \underline{(20,0)} - \frac{\underline{\underline{z}} \underline{(10,0)}}{\underline{\underline{z}} \underline{(20,0)}} \frac{\underline{\varepsilon} \underline{(1,1)}}{\underline{\varepsilon} \underline{(2,1)}} \underline{\gamma} \underline{a[T_1]^3} \underline{(20,0)} \\
& + \frac{\underline{\varepsilon} \underline{(1,1)}}{\underline{\varepsilon} \underline{(2,1)}} \frac{1}{\underline{\underline{z}} \underline{(1,0,11)}} \frac{1}{\underline{\underline{z}} \underline{(20,0)}} \underline{q\omega^2} \underline{a[T_1]} \underline{(20,0)} \underline{e} \underline{(\$,21)} \underline{(9,0,10,19)} \underline{T_0} + \underline{\sigma} \underline{(14,1)} \underline{T_1} + \underline{-2} \underline{(9,0)} \underline{T_0} - \underline{\underline{z}} \underline{(21,0)} \underline{\varphi[T_1]} \underline{(20,0)} \\
& - \underline{i} \underline{(9,0)} \times \underline{i} \underline{(21,0)} \frac{\underline{\underline{z}} \underline{(4,1)}}{\underline{\underline{z}} \underline{(20,0)}} \frac{\underline{\varepsilon} \underline{(4,1)}}{\underline{\varepsilon} \underline{(2,1)}} \underline{a[T_1]} \underline{(20,0)} \underline{D_1} \underline{\varphi[T_1]} \underline{(20,0)} = 0
\end{aligned} \tag{21}$$

By considering the large amount of encoding information that is generated in the exponential

term  $e^{\underset{(\mathcal{E},21)}{\overset{i}{\underbrace{\left( \underset{(13,0)}{\underbrace{2}{\underbrace{T_0}{\mathcal{E}}}} + \underset{(14,1)}{\underbrace{\sigma}{\mathcal{E}}} T_1 + \underset{(9,0)}{\underbrace{-2}{\mathcal{E}}} T_0 - \underset{(21,0)(20,0)}{\underbrace{2}{\mathcal{E}}} \varphi[T_1] \right)}}}$ , it can now be seen that this term is created

from equations (9), (13), (14), (20), and (21). Also, the  $\mathcal{E}$  symbol within the encodings of the exponential term shows that this term has been simplified for the fourth time.

### Modulation equations

To obtain the modulation equations, the imaginary and real parts of equation (21) are separated out and set to zero, resulting in equations (22) and (23). There is no evidence of equations (22) and (23) within the encodings, as no particular modification has been made at this stage.

$$\begin{aligned} & \frac{\underset{(4,1)}{\underbrace{2}{\mathcal{E}}}}{\underset{(20,0)}{\underbrace{2}{\mathcal{E}}}} \frac{\underset{(4,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \underset{(4,1)}{D_1} \underset{(20,0)}{a[T_1]} + \frac{\underset{(1,1)}{\underbrace{2}{\mathcal{E}}}}{\underset{(20,0)}{\underbrace{2}{\mathcal{E}}}} \frac{\underset{(1,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \underset{(1,1)}{\beta} \underset{(20,0)}{a[T_1]} \\ & + \frac{\underset{(1,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \frac{1}{\underset{(1,0,11)}{\underbrace{2}{\mathcal{E}}}} \frac{1}{\underset{(20,0)}{\underbrace{2}{\mathcal{E}}}} \underset{(1,1)}{q\omega^2} \underset{(20,0)}{a[T_1]} \sin\left( \underset{(13,0)}{\underbrace{2}{\mathcal{E}}} T_0 + \underset{(14,1)}{\underbrace{\sigma}{\mathcal{E}}} T_1 + \underset{(9,0)}{\underbrace{-2}{\mathcal{E}}} T_0 \right. \\ & \left. - \underset{(21,0)}{\underbrace{2}{\mathcal{E}}} \varphi[T_1] \right) = 0 \end{aligned} \quad (22)$$

$$\begin{aligned} & - \frac{\underset{(10,0)}{\underbrace{3}{\mathcal{E}}}}{\underset{(20,0)}{\underbrace{2^3}{\mathcal{E}}}} \frac{\underset{(1,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \underset{(1,1)}{\gamma} \underset{(20,0)}{\alpha[T_1]^3} + \frac{\underset{(1,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \frac{1}{\underset{(1,0,11)}{\underbrace{2}{\mathcal{E}}}} \frac{1}{\underset{(20,0)}{\underbrace{2}{\mathcal{E}}}} \underset{(1,1)}{q\omega^2} \underset{(20,0)}{a[T_1]} \cos\left( \underset{(13,0)}{\underbrace{2}{\mathcal{E}}} T_0 + \underset{(14,1)}{\underbrace{\sigma}{\mathcal{E}}} T_1 \right. \\ & \left. + \underset{(9,0)}{\underbrace{-2}{\mathcal{E}}} T_0 - \underset{(21,0)}{\underbrace{2}{\mathcal{E}}} \varphi[T_1] \right) - \underset{(9,0)}{i} \times \underset{(21,0)}{i} \frac{\underset{(4,1)}{\underbrace{2}{\mathcal{E}}}}{\underset{(20,0)}{\underbrace{2}{\mathcal{E}}}} \frac{\underset{(4,1)}{\underbrace{\mathcal{E}}}}{\underset{(2,1)}{\underbrace{\mathcal{E}}}} \underset{(20,0)}{a[T_1]} \underset{(4,1)}{D_1} \underset{(20,0)}{\varphi[T_1]} \\ & = 0 \end{aligned} \quad (23)$$

So that it is possible to identify the stationary response of the system it is essential to be able to find conditions for which the amplitudes and the phases are constant with time. The notation of singular points is related to autonomous systems, therefore,  $\psi$  as a new variable is introduced to eliminate  $T_1$ . The encodings clearly address each quantity to its source. The third digit of encoding is added at this point to the encoding vector of quantities with just a second level of encoding.

$$\underbrace{\varphi[T_1]}_{(20,0,24)} = \frac{\underbrace{2}_{(13,0,24)}}{\underbrace{2}_{(21,0,24)}} T_0 + \frac{\underbrace{\sigma}_{(14,1,24)}}{\underbrace{2}_{(21,0,24)}} T_1 + \frac{\underbrace{-2}_{(9,0,24)}}{\underbrace{2}_{(21,0,24)}} T_0 - \frac{\underbrace{\psi}_{(24,0)}}{\underbrace{2}_{(21,0,24)}} \quad (24)$$

The autonomous pair of modulation equations is obtained by substituting equation (24) into equations (22) and (23), resulting in:

$$\begin{aligned} & \frac{\underbrace{2}_{(4,1)}}{\underbrace{2}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(4,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{D_1}_{(4,1)} \underbrace{a[T_1]}_{(20,0)} + \frac{\underbrace{2}_{(1,1)}}{\underbrace{2}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\beta}_{(1,1)} \underbrace{a[T_1]}_{(20,0)} \\ & + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \frac{1}{\underbrace{2}_{(1,0,11)}} \frac{1}{\underbrace{2}_{(20,0)}} \underbrace{q\omega^2}_{(1,1)} \underbrace{a[T_1]}_{(20,0)} \sin(\underbrace{\psi}_{(24,0)}) = 0 \end{aligned} \quad (25)$$

$$\begin{aligned} & - \frac{\underbrace{3}_{(10,0)}}{\underbrace{2^3}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\gamma}_{(1,1)} \underbrace{a[T_1]^3}_{(20,0)} + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \frac{1}{\underbrace{2}_{(1,0,11)}} \frac{1}{\underbrace{2}_{(20,0)}} \underbrace{q\omega^2}_{(1,1)} \underbrace{a[T_1]}_{(20,0)} \cos(\underbrace{\psi}_{(24,0)}) \\ & - \underbrace{i}_{(9,0)} \times \underbrace{i}_{(21,0)} \frac{\underbrace{2}_{(4,1)}}{\underbrace{2}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(4,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{a[T_1]}_{(20,0)} \left( - \frac{\underbrace{D_1}_{(4,1)}}{\underbrace{2}_{(21,0,24)}} \underbrace{\psi}_{(24,0)} + \frac{\underbrace{\sigma}_{(14,1)}}{\underbrace{2}_{(21,0,24)}} \right) = 0 \end{aligned} \quad (26)$$

Furthermore, the multiplication in  $-\underbrace{i}_{(9,0)} \times \underbrace{i}_{(21,0)}$  is avoided as the encodings information are not similar.

The singular points can be calculated by stipulating the conditions that  $D_1\psi[T_1] = D_1\alpha[T_1] = 0$ , physically meaning that  $\psi$  and  $\alpha$  are not dependent on  $T_1$ , and therefore virtually static. Applying these conditions to the modulation equations and dividing the LHS of the equation by  $\alpha[T_1]$ , yields:

$$\frac{\underbrace{2}_{(1,1)}}{\underbrace{2}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\beta}_{(1,1)} + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \frac{1}{\underbrace{2}_{(1,0,11)}} \frac{1}{\underbrace{2}_{(20,0)}} \underbrace{q\omega^2}_{(1,1)} \sin(\underbrace{\psi}_{(24,0)}) = 0 \quad (27)$$

$$\begin{aligned} & - \frac{\underbrace{3}_{(10,0)}}{\underbrace{2^3}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \underbrace{\gamma}_{(1,1)} \underbrace{a[T_1]^2}_{(20,0)} + \frac{\underbrace{\varepsilon}_{(1,1)}}{\underbrace{\varepsilon}_{(2,1)}} \frac{1}{\underbrace{2}_{(1,0,11)}} \frac{1}{\underbrace{2}_{(20,0)}} \underbrace{q\omega^2}_{(1,1)} \cos(\underbrace{\psi}_{(24,0)}) \\ & - \underbrace{i}_{(9,0)} \times \underbrace{i}_{(21,0)} \frac{\underbrace{2}_{(4,1)}}{\underbrace{2}_{(20,0)}} \frac{\underbrace{\varepsilon}_{(4,1)}}{\underbrace{\varepsilon}_{(2,1)}} \frac{\underbrace{\sigma}_{(14,1)}}{\underbrace{2}_{(21,0)}} = 0 \end{aligned} \quad (28)$$

By considering equations (28) and (29) it is possible to note that most of the quantities would have been cancelled or multiplied out in a standard multiple scales analysis. Also, there is no evidence of equations (28) and (29) appearing here as no new quantity has been added at this point.

There are two equations and two unknowns in using equations (28) and (29). Therefore, it is possible to calculate the explicit value of the amplitude ( $a[T_1]$ ) and autonomous form of phase ( $\psi[T_1]$ ). The first possibility is  $a[T_1] = 0$ , which means the pendulum is at rest. The second possibility is determined by solving equations (27) and (28), resulting in:

$$a^2 = \frac{\frac{\sigma}{(14,1,29)} \frac{2^2}{(21,0,29)} \frac{2}{(4,1,29)} \frac{\varepsilon}{(4,1,29)}}{\frac{3}{(10,0,29)} \frac{2}{(21,0,29)} \frac{\gamma}{(1,1,29)} \frac{\varepsilon}{(1,1,29)}} + \frac{2^2}{(21,0,29)} \sqrt{\frac{-2^2}{(1,1,29)} \frac{2^2}{(1,0,11)} \frac{\beta^2}{(1,1,29)} \frac{\gamma^2}{(1,1,29)} \frac{\varepsilon^2}{(2,1,29)} \frac{\varepsilon^2}{(1,1,29)} + \frac{\omega^4}{(1,1,29)} \frac{q^2}{(1,1,29)} \frac{\gamma^2}{(1,1,29)} \frac{\varepsilon^2}{(1,1,29)} \frac{\varepsilon^2}{(1,1,29)}}{\frac{3}{(10,0,29)} \frac{2}{(1,0,11)} \frac{\gamma^2}{(1,1,29)} \frac{\varepsilon^2}{(1,1,29)}} \quad (29)$$

$$\psi_{(24,0)} = \quad (30)$$

$$\text{Arctan}\left[ \frac{-\frac{2}{(20,0)} \frac{\varepsilon}{(2,1)} \frac{\beta}{(1,1)}}{\frac{3}{(10,0,30)} \frac{\varepsilon}{(1,1,30)} \frac{\gamma}{(1,1,30)} \frac{a[T_1]^2}{(20,0,30)} + \frac{i}{(9,0,30)} \times \frac{i}{(21,0,30)} \frac{2}{(20,0,30)} \frac{\varepsilon}{(2,1,30)} \frac{\sigma}{(14,1,30)}} \right]$$

The encoding information generated by SEEM is critical in these equations because it shows how the amplitude and phase of the response is connected to the physical source.

The next step in the analysis is to construct the near-resonance solution. The zeroth-order perturbation solution based on  $a[T_1]$  and  $\varphi[T_1]$  are given as follow:

$$\theta_0 = \frac{1}{2} \frac{a[T_1]}{(20,0,31)} e_{(*) , 31}^{i \left( \frac{1}{(9,0,31)} T_0 + \frac{\varphi[T_1]}{(20,0,31)} \right)} + \frac{1}{2} \frac{a[T_1]}{(20,0,31)} e_{(*) , 31}^{-i \left( \frac{1}{(9,0,31)} T_0 + \frac{\varphi[T_1]}{(20,0,31)} \right)} \quad (31)$$

This equation is structured from equations (9) and (20), and the third encoding digit is added at this stage. The compound level is used in this equation as well. The trigonometrical form of equation (31) is defined as:

$$\theta_0 = \frac{a[T_1]}{(20,0,31)} \cos\left(\frac{1}{(9,0,31)} T_0 + \frac{\varphi[T_1]}{(20,0,31)}\right) \quad (32)$$

It is possible to notice that the encodings of the terms are not affected by this action. Then equation (24) is substituted into (32), resulting in:

$$\theta_0 = \frac{a[T_1]}{(20,0,31)} \cos\left(\frac{1}{(9,0,31)} T_0 + \frac{\frac{2}{(21,0)} \frac{\omega}{(14,0,34)}}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}} T_0 + \frac{\frac{\sigma}{(21,0)} \frac{\omega}{(14,0,34)}}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}} T_1 + \frac{\frac{-2}{(21,0)} \frac{\psi}{(24,0)}}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}} T_0 - \frac{\frac{\psi}{(21,0)} \frac{\omega}{(24,0)}}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}}\right) \quad (33)$$

The interactions between the first natural frequency of the system, the principal parametric resonance and the solvability condition can be seen here. The detuning parameter is defined as based on equation (14):

$$\frac{\sigma}{(14,1)} = \frac{\frac{\omega}{(14,0,34)}}{\frac{\varepsilon}{(14,1,34)}} - \frac{\frac{2}{(13,0,34)}}{\frac{\varepsilon}{(14,1,34)}} \quad (34)$$

Then the definitions of fast time,  $T_0 = \tau$ , and slow time,  $T_1 = \frac{\varepsilon}{(2,1)} \tau$ , and equation (34) are all

substituted into equation (33). As mentioned before  $\frac{\varepsilon}{(14,1)}$  must be equal to  $\frac{\varepsilon}{(2,1)}$ . The zeroth-

order perturbation solution now emerges in the following form:

$$\theta_0 = \frac{a[T_1]}{(20,0,31)} \cos\left[\frac{1}{(9,0,31)} - \frac{\frac{2}{(21,0)} \frac{\omega}{(13,0,34)}}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}}\right] \tau + \frac{1}{\frac{2}{(21,0)} \frac{\varepsilon}{(14,1,34)}} \left[ \left( \frac{2}{(13,0)} - \frac{2}{(9,0)} + \frac{\omega}{(14,0,34)} \right) \tau + \frac{\psi}{(24,0)} \right] \quad (35)$$

It is possible to discuss the SEEM information in the argument of the trigonometric function as follows:



$$\begin{aligned}
\theta_{\downarrow 1}^{(2,1)} = & -\frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}} \frac{1}{2^3} \frac{1}{8} \gamma_{\downarrow}^{(1,1,19)} \frac{a[T_1]^3}{(20,0)} \cos\left(\frac{3}{(1,0,10,19)} T_0 + 3 \frac{\varphi[T_1]}{(20,0,36)}\right) \\
& + \frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}} \frac{q\omega^2}{(1,1,19)} \frac{1}{2} \frac{1}{8} \frac{1}{2} \cos\left(\frac{2}{(13,0,19)} T_0 + \frac{\sigma}{(14,1,19)} T_1 \right. \\
& \left. + \frac{1}{(9,0,19)} T_0 + \frac{\varphi[T_1]}{(20,0,36)}\right)
\end{aligned} \tag{37}$$

$$\begin{aligned}
\theta_{\downarrow 1}^{(2,1)} = & -\frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}} \frac{1}{2^3} \frac{1}{8} \gamma_{\downarrow}^{(1,1,19)} \frac{a[T_1]^3}{(20,0)} \cos\left(\frac{3}{(1,0,10,19)} \tau \right. \\
& \left. + \frac{3}{2} \left[ \left( \frac{2}{(13,0)} - \frac{2}{(9,0)} - \frac{2}{(13,0,34)} - \frac{\omega}{(14,0,34)} \right) \tau - \frac{\psi}{(24,0)} \right] \right) \\
& + \frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}} \frac{q\omega^2}{(1,1,19)} \frac{1}{2} \frac{1}{8} \frac{1}{2} \cos\left[ \left( \frac{2}{(13,0,19)} + \frac{1}{(9,0,19)} + \frac{2}{(13,0)} \right. \right. \\
& \left. \left. - \frac{2}{(9,0)} - \frac{2}{(13,0,34)} - \frac{2}{(13,0,34)} \right) \tau + \left( \frac{\omega}{(14,0,34)} + \frac{(14,0,34)}{2} \right) \tau + \frac{\psi}{(24,0)} \right]
\end{aligned} \tag{38}$$

Again, the interactions between the natural frequency, the principal parametric resonance and the solvability conditions are clearly visible in this equation.

At the last stage of the analysis, the solutions of the both zeroth and first order perturbation equations are substituted into the perturbation expansion equation, resulting in equation (39).

Considering the encoding information,  $\frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}}$  in the denominator of the second term can be

cancelled with  $\frac{\varepsilon_{\downarrow}^{(1,1,19)}}{\varepsilon_{\downarrow}^{(2,1,19)}}$ , given that the first two encoding digits are the same.

$$\begin{aligned}
\theta(\tau, \varepsilon) = & \frac{a[T_1]}{\binom{20,0,31}} \cos\left( \frac{1}{\binom{9,0,31}} - \frac{\binom{2}{(13,0,34)}}{\binom{2}{(21,0)}} \right) \tau + \frac{1}{\binom{2}{(21,0)}} \left[ \left( \frac{\binom{2}{(13,0)}}{\binom{2}{(13,0)}} - \frac{\binom{2}{(9,0)}}{\binom{2}{(9,0)}} + \frac{\binom{\omega}{(14,0,34)}}{\binom{2}{(14,0,34)}} \right) \tau + \frac{\binom{\psi}{(24,0)}}{\binom{2}{(24,0)}} \right] \\
& - \frac{\frac{\binom{\varepsilon}{(2,1)}}{\binom{\varepsilon}{(2,1,19)}} \frac{\binom{\varepsilon}{(1,1,19)}}{\binom{\varepsilon}{(1,1,19)}}}{\binom{\varepsilon}{(2,1,19)}} \frac{1}{\binom{2^3}{(20,0)}} \frac{1}{\binom{8}{(19,0)}} \gamma \frac{a[T_1]^3}{\binom{20,0}} \cos\left( \frac{\binom{3}{(1,0,10,19)}}{\binom{3}{(1,0,10,19)}} \tau \right. \\
& \left. + \frac{\binom{3}{(20,0,36)}}{\binom{2}{(21,0)}} \left[ \left( \frac{\binom{2}{(13,0)}}{\binom{2}{(13,0)}} - \frac{\binom{2}{(9,0)}}{\binom{2}{(9,0)}} - \frac{\binom{2}{(13,0,34)}}{\binom{2}{(13,0,34)}} + \frac{\binom{\omega}{(14,0,34)}}{\binom{2}{(14,0,34)}} \right) \tau - \frac{\binom{\psi}{(24,0)}}{\binom{2}{(24,0)}} \right] \right. \\
& \left. + \frac{\frac{\binom{\varepsilon}{(2,1)}}{\binom{\varepsilon}{(2,1,19)}} \frac{\binom{\varepsilon}{(1,1,19)}}{\binom{\varepsilon}{(1,1,19)}}}{\binom{\varepsilon}{(2,1,19)}} \frac{q\omega^2}{\binom{1,1,19}} \frac{1}{\binom{2}{(1,0,11,19)}} \frac{1}{\binom{8}{(19,0)}} \frac{1}{\binom{2}{(20,0)}} \cos\left[ \left( \frac{\binom{2}{(13,0,19)}}{\binom{2}{(13,0,19)}} + \frac{\binom{1}{(9,0,19)}}{\binom{1}{(9,0,19)}} + \frac{\binom{2}{(13,0)}}{\binom{2}{(21,0)}} \right. \right. \\
& \left. \left. - \frac{\binom{2}{(9,0)}}{\binom{2}{(21,0)}} - \frac{\binom{2}{(13,0,34)}}{\binom{2}{(21,0)}} - \frac{\binom{2}{(13,0,34)}}{\binom{2}{(13,0,34)}} \right) \tau + \left( \frac{\binom{\omega}{(14,0,34)}}{\binom{2}{(14,0,34)}} + \frac{\binom{\omega}{(14,0,34)}}{\binom{2}{(21,0)}} \right) \tau + - \frac{\binom{\psi}{(24,0)}}{\binom{2}{(21,0)}} \right] \\
& + O(\varepsilon^2)
\end{aligned} \tag{39}$$

It is possible to highlight the percentage of contribution of different equations in the solution procedure, in symbolically structuring the final response. This can be done by tracking the first digit of the SEEM for each encoding vector in the final response and compare it to the overall number of encoding vectors for the equation. Overall 45 encoding vectors for equation (39) are defined, and the percentage of contribution for each equation compared to the overall is given in Table 3-2. In this table, the first column describes the equation detail, the second column is the number of encoding vectors that are sourced to the relevant equation, and the last column shows the percentage of contribution of that specific equation number in the final response. For example considering the first row of the table, there are 6 quantities out of 45 that are addressed to equation (1), according to the first digit of the SEEM encoding. The contribution of equation of motion (1) in structuring the final response is roughly 13% ( $\frac{6}{45} \times 100$ ).

Table 3-2 The percentage of contribution of each equation for the final solution

Equations details	Total number of quantities	Percentage of contribution
Equation of Motion (1)	6	13
Perturbation expansion (2)	4	9
Solution of the zeroth-order perturbation equation (9)	5	11
Principle parametric resonance (13)	8	18
Near resonance condition (14)	4	9
Solution for the first order perturbation equation (19)	2	4
Polar form of the complex amplitude A (20)	5	11
Solvability condition (21)	8	18
Autonomous form of phase (24)	3	7

### 3.5 Summary

In this chapter, the general concept of SCD solvers is introduced. SCD solvers currently consist of two main parts; a core solver and a term-tracker. In this study, symbolic computational code for the multiple scales method [7], with many modifications, is used to create the core solver. The SEEM is defined and introduced practically as a new term-tracking method and a summary of the underlying encoding strategy of the SEEM is provided in Table 3-1.

The SEEM produces encoding information when a quantity is introduced to the analysis for the first time and also tracks how the information evolves during the solution procedure.

For demonstrating and highlighting the distinctions of the SEEM, it is applied in full to the dynamic problem of a parametrically excited pendulum. The equation of motion for this problem is derived and the assumptions that are made in both the physical modelling and the solution procedure are highlighted. The results of this analysis are listed as follows:

- The SEEM has created valuable information in each stage of the analysis, and the information can be used to identify the sources of each term and make a connection back to the physical conceptualisation of the problem.
- There is a trade-off between flexibility and user interaction in the construction and operation of SCD solvers of this nature.

- The meaning of the encoding information is directly dependent on the solution procedure, and the way it is introduced mathematically. Therefore, all SEEM information must always be presented in close association with the chosen solution procedure.
- In the early stages of the analysis the encoding information for all the quantities within an equation is reasonably similar in form and content. However, as the problem develops the SEEM-generated encoding information becomes much more complex.
- The cancellation of parameters within the solution procedure that would naturally follow an efficient and elegant algebraic process is deliberately avoided within a SEEM approach; unless both the quantity and the two first encodings digits are identical.
- The small parameter ‘epsilons’ that are introduced at different stages of the analysis do not necessarily have the same numerical value. The exception to this is  $\underset{(3,1)}{\varepsilon}$ ,  $\underset{(4,1)}{\varepsilon}$ , and  $\underset{(2,1)}{\varepsilon}$  which have different encoding information, but the same numerical value. This is because they are all sourced back to the perturbation expansion.
- To ensure a uniformly valid perturbation expansion, the  $\underset{(14,1)}{\varepsilon}$  in the detuning parameter must be numerically equal to  $\underset{(2,1)}{\varepsilon}$ .
- Applying the SEEM to many of the numbers that emerge within the perturbation analysis helps to relate the physical concept to the mathematical equations. One example of this is highlighted in equation (16), where the principal parametric resonance and the first natural frequency of the system are seen to influence each other.

# Chapter 4 Computerisation of the Source Evolution and Encoding Method

The general concept underpinning the Source Evolution and Encoding method (SEEM) is thoroughly discussed in the previous chapter. This chapter discusses the process of developing effective algorithms which can be used to computerise the SEEM concept. These algorithms are compatible with the structure of the associated Symbolic Computational Dynamics (SCD) solver. Flexibility and user control are two main guiding principles in the development of the SCD solvers; therefore in this study, great effort has been expended in satisfying these requirements.

## 4.1 The core solver

The perturbation method of multiple scales has been the preferred choice of solution used in this research, mainly because of its adaptability, transparency, and logical algorithmic structure. The core solver of the SCD in this study is based on a modified version of the multiple scales code as developed by Forehand and Cartmell [7]. In this research the powerful symbolic programming interface provided by the *Mathematica* language is used in depth but it should also be noted that it would potentially be perfectly possible to programme the algorithms in another symbolic computational language, as may be required.

## 4.2 The general procedure for encoding application

The policy of the SEEM is to collect information about the fundamental quantities in each equation therefore the encoding algorithm must be capable of decomposing each equation into expressions, terms, sub-terms, down to the fundamental individual quantities inside each sub-term. A schematic view of this process, as applied to the equation of motion of a parametrically excited pendulum, is shown in Figure 4-1. In this figure the encoding algorithm must be able to identify the different function types, to decompose the sub-terms accurately. For example, the

trigonometrical function  $\cos(\omega T_0)$  is decomposed to  $\omega$ ; whilst the power function of  $\omega^2$  is decomposed down to  $\omega$  and 2.

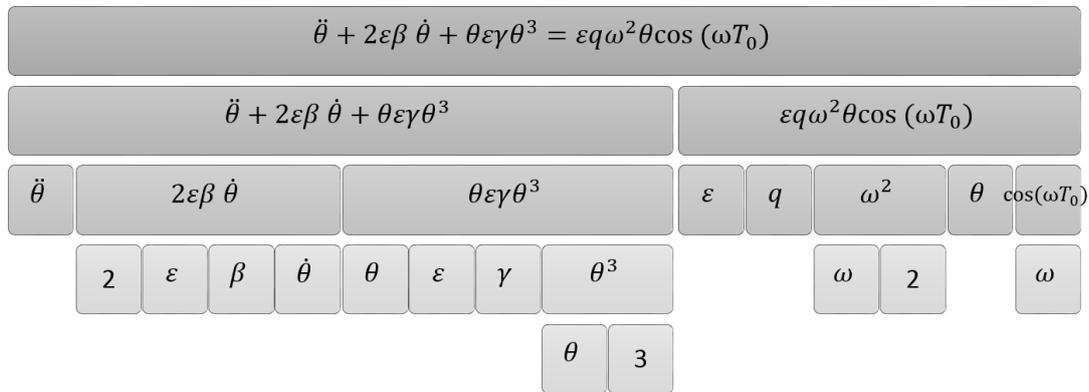


Figure 4-1 A graphical representation of the decomposition process of an equation into the fundamental quantities.

After the decomposition process has been completed the encoding information (constituting the encoding vectors) must be added into the equation structure. Considering Figure 4-2, the encoding information for each quantity is added in the form of a multiplier,  $X_n$ . As the encoding information for each fundamental quantity is unique, the index  $n$  is used to define a single piece of encoding information which is represented for each quantity. Afterwards both the quantity and its encoding information are combined as an encoded sub-term. Then, the encoded sub-terms are joined together to create encoded terms. Encoded terms are merged to give structured encoded expressions, and finally the encoded expressions are combined to create the encoded equation.

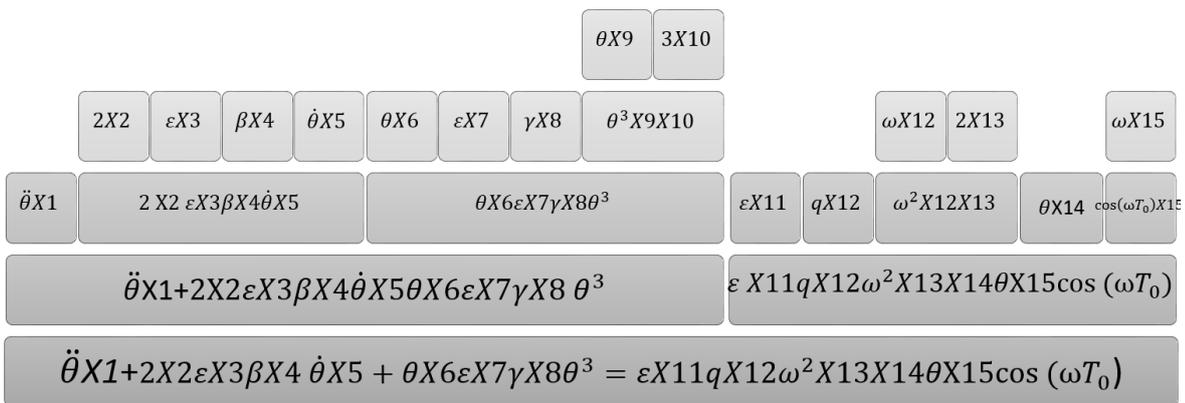


Figure 4-2 A graphical representation of the combination process; encoded quantities into the encoded equation.

When taking the SEEM methodology into consideration the encoding information must be defined and updated at each step of the solution procedure. Therefore all the equations in the SCD solver must go through both the decomposition and combination process at least once. As the results of these processes can then be combined to re-form as a new concept, which for convenience is called 'the SEEM encoding cycle'. Figure 4-3 graphically represents 'the SEEM encoding cycle' in ten steps, these steps being summarised as follows:

**Step-1:** The input equation, from the core solver, is decomposed into two expressions; the left-hand side (LHS) and right-hand side (RHS).

**Step-2:** The expressions are decomposed into terms.

**Step-3:** The terms are disintegrated down to sub-terms.

**Step-4:** The sub-terms are then decomposed into the original quantities.

**Step-5:** The encoding vector for each quantity is defined or updated, and it is multiplied by the fundamental quantity.

**Step-6:** The encoded quantities are combined to structure the encoded sub-terms.

**Step-7:** The encoded sub-terms are multiplied together to form the encoded full terms.

**Step-8:** The encoded terms are added together to form encoded expressions.

**Step-9:** The encoded expressions are combined together appropriately to form an encoded equation.

**Step-10:** The encoded equation is sent to the solution procedure, which might be re-encoded later, dependent on what is needed functionally.

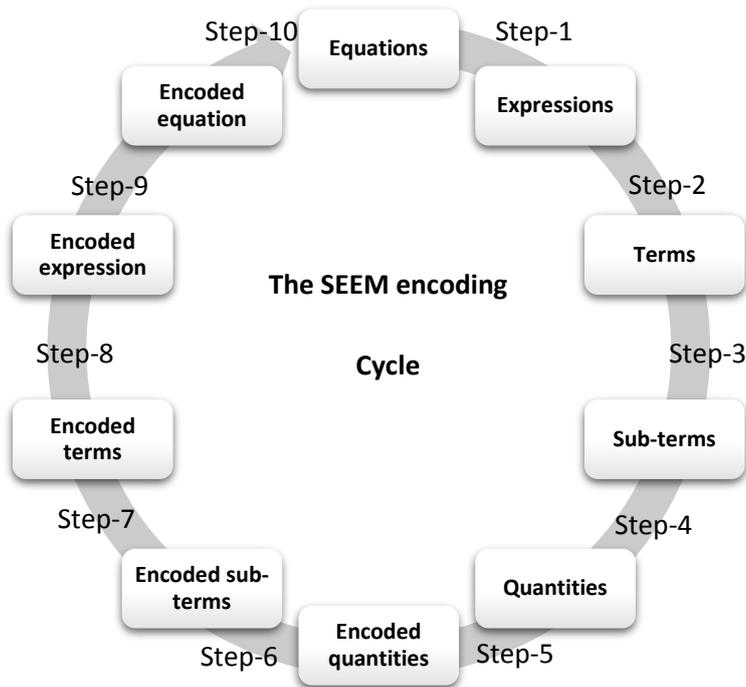


Figure 4-3 A graphical representation of ‘the SEEM encoding cycle’.

#### 4.2.1 Computerisation of the encoding vector

There are several principal challenges in computerisation of the SEEM, namely storing the encoding information in the equation structure in such a way that it does not have any influence on the mathematical solution procedures. Also, depending on the SEEM level, the encoding vectors should have the provision for variable length (§3.3). Furthermore the encoding function must be sophisticated enough to distinguish between different function types (§4.2). Despite the fact that encoding an unencoded equation can be achieved quite readily, updating the encoding information for previously encoded quantities can be a rather more demanding process.

In order to be able to identify the SEEM-related information from the actual mathematical structure of the equation, a special format is required. In the *Mathematica* language everything is an expression of one form or another, and different type of expressions can be identified by their *Head*; for example, the *Head* of the number 2 in the *Mathematica* language is given by *Integer*. As well as the built-in *Mathematica Heads* (expression types) it possible to define a new class of *Head* to store and identify the SEEM information. As the results a new class of *Head* named ‘*envec*’ has been defined; where ‘*envec*’ is an abbreviation for encoding vector. Consequently the Computerised Encoding Vector (CEV) in the format of *envec*[*identity factor*,  $n_1, n_2, n_3, n_4$ ] is created;  $n_1, n_2, n_3, n_4$  are the SEEM encoding

elements (Table 3-1), and *identity factor* is defined in the next paragraph. To eliminate any possible interaction between the CEVs with the solution procedure, all elements inside the CEVs must be in either the string or number format.

The *identity factor* is a string form of the input quantity and this element is introduced to assist the algorithm in identifying the correct CEV for a selected quantity. For example when considering an arbitrary encoded statement such as (4.1) the encoding information is converted to a CEV format in statement (4.2). During the mathematical procedure, the presentation of this statement can be altered into statement (4.3). In this situation it is impossible for the encoding algorithm to classify the correct encoding information for a selected quantity. As there is no indication of the input quantity inside the corresponding CEV, so the algorithm cannot identify the correct encoding information for a given input quantity. As the results, in statement (4.4) the *identity factor* is added as the first element in the CEV. This makes it possible for the algorithm to select the correct CEV for each quantity. Note that for the unavailable encoding digits “nan” is used, this is an abbreviation for ‘not a number’. The *identity factor* is added as the first element of the CEV, to keep the CEV flexible. Therefore, the CEV can be extended from the other side to store more information if this is required.

$$\begin{matrix} (5,0,10) & (1,1) & (1,1) & (1,1) \\ \vec{2} & \vec{\varepsilon} & \vec{\beta} & \vec{l} \end{matrix} \quad (4.1)$$

$$\begin{aligned} & 2\text{envec}[5,0,10, "nan"] \times \varepsilon\text{envec}[1,1, "nan", "nan"] \\ & \quad \times \beta\text{envec}[1,1, "nan", "nan"] \times l\text{envec}[1,1, "nan", "nan"] \end{aligned} \quad (4.2)$$

$$\begin{aligned} & 2\varepsilon\beta l \times \text{envec}[5,0,10, "nan"]\text{envec}[1,1, "nan", "nan"]\text{envec}[1,1, "nan", "nan"] \\ & \quad \times \text{envec}[1,1, "nan", "nan"] \end{aligned} \quad (4.3)$$

$$\begin{aligned} & 2\varepsilon\beta l\text{envec}["\varepsilon", 1,1, "nan", "nan"]\text{envec}["l", 1,1, "nan", "nan"] \\ & \quad \text{envec}["2", 5,0,10, "nan"]\text{envec}["\beta", 1,1, "nan", "nan"] \end{aligned} \quad (4.4)$$

### 4.3 The encoding algorithms

Two main functions are developed as a basis for computerising the SEEM term-tracking method. These are defined by *SEEMencoder* and *SEEMcore*. The *SEEMencoder* is a user-end function which formats both the input and the output of the *SEEMcore* function. Moreover, the *SEEMcore* applies the SEEM cycle to the input equation and manages the CEVs.

### 4.3.1 The *SEEMencoder* function

Considering Algorithm 4-1, the inputs of this function are defined as;  $j$ ,  $s$ , and  $k$ , where,  $j$  is the input equation number,  $s$  is the relevant side of the equation; so the LHS of the input equation is represented by 1, and RHS of the input equation is identified by 2. Finally,  $k$  is the actual source of the equation, and  $k$  and  $j$  are unique for most cases. To demonstrate how these input values are defined, equation (10) in the analysis of the parametric excitation of a pendulum (§3.4) is considered. The user can decide to introduce a new equation for the polar form of the harmonic excitation introduced in equation (1). In this case the actual equation number that needs to be encoded is 10 ( $j = 10$ ), while the true source of this equation is equation (1), ( $k = 1$ ).

$$\cos(\omega T_0) = \frac{1}{2} \left( \underset{(1,0,11)}{\underbrace{e}_{(1,0,11)}}^{\underset{(1,0,11)}{\underbrace{j}_{(1,0,11)}}} \underset{(1,0,11)}{\underbrace{\omega}_{(1,0,11)}} T_0 + \underset{(1,0,11)}{\underbrace{e}_{(1,0,11)}}^{\underset{(1,0,11)}{\underbrace{-j}_{(1,0,11)}}} \underset{(1,0,11)}{\underbrace{\omega}_{(1,0,11)}} T_0 \right) \quad (10)$$

The *SEEMencoder* function disables the differentiation operators and passes both the LHS (Lines 2 to 15) and the RHS (lines 15 to 30) of the equation to the *SEEMcore* function. Then it takes the encoded expression as the output of the *SEEMcore* and activates the differentiation operators. Finally the encoded equation is structured and then sent back to the core solver.

Algorithm 4-1 The *SEEMencoder* function algorithm.

```

1: Function SEEMencoder[j,s,k]
2:   if j == LHS,
3:     if LHS == Function,
4:       if Function == expression,
5:         Deactivate the function, out <- SEEMcore[expression,i,j,k]
6:       else
7:         SEEMcore[term,j,s,k]
8:       end
9:     else
10:      if LHS == expression,
11:        out <- SEEMcore[expression,j,s,k]
12:      else
13:        out <- SEEMcore[term,j,s,k]
14:      end
15:    end
16:  else
17:    if j == RHS,
18:      if LHS == Function,
19:        Deactivate the function, out <- SEEMcore[expression,j,s,k]
20:      else
21:        out <- SEEMcore[expression,j,s,k]
22:      end
23:    else
24:      if LHS == expression,
25:        out <- SEEMcore[expression,j,s,k]
26:      else
27:        out <- SEEMcore[term,j,s,k]
28:      end
29:    end
30:  end
31:  Restore the differentiation operators in the output
32: endFunction

```

### 4.3.2 The *SEEMcore* function

The *SEEMcore* is the main algorithm for defining or updating encoding information. This function has a complex structure; therefore it is explained in 14 separate algorithms. In the actual code these algorithms are all included in the main body of the *SEEMcore* function. As Algorithm 4-2 shows, the inputs of this function are: an expression (LHS or RHS of the input equation),  $j$ ,  $s$ ,  $k$ ., where  $j$  is the input equation,  $s$  is the relevant side of the equation, and  $k$  is the actual source of the equation.

This function extracts information from the input expression in several stages. At the first stage all relevant information from the input expression is extracted. The CEVs for each side of the equation are then stored separately (lines 2 and 3). In the next step the value for the fourth digit of the CEV,  $n_3$  is defined (lines 4 to 10). When considering the SEEM second encoding level the equation number of an explicit form equation is added to all the CEVs in that equation. Consequently line 4 is used to check if the equation is defined in an explicit form or not; this means whether a single quantity appears on the LHS and there are no differentiation operators on either side of the equation. Although this condition is not necessarily mean that  $n_3$  must be updated. Considering line 5, if the input equation number is equal to the source equation

number ( $j = k$ ), adding a third digit of encoding would be meaningless; as the encoding vectors would be of this form (40, 0, 40). If in an explicit equation the source and the input equation number are dissimilar ( $j \neq k$ ), then  $n_3$  is set to the input equation number ( $j$ ), otherwise, it is set to an empty cell, shown by "nan".

The CEVs are stored in the equation structure as multipliers; therefore having a numerical zero on either side of the equation can lead to information loss. Therefore the zero from the relevant side of the equation is temporarily removed (line 13). Afterwards all the CEVs from both sides of the equation are extracted and stored (line 14).

Following on from this, the input expression is decomposed into its fundamental terms, and the CEVs for each term are extracted and stored in the *indextrm* directory. Then the actual symbolic form of the term, without the encodings, is saved in the *trm* directory. Table 4-1 shows the *indextrm* and *trm* directories, for an arbitrary term. After separating the terms and CEVs, the  $n_2$  digit of the SEEM, this being the order of the small parameter 'epsilon', is determined. All the information that is generated in this stage is able to be reset for each term, and so nothing will actually be stored in the *SEEMcore* function.

Table 4-1 An example of the *indextrm* and *trm* for an arbitrary term.

<b>Input term</b>	$2\varepsilon\beta l \text{envec}["\varepsilon", 1, 1, "nan", "nan"] \text{envec}["l", 1, 1, "nan", "nan"]$ $\text{envec}["2", 5, 0, 10, "nan"] \text{envec}["\beta", 1, 1, "nan", "nan"]$
<b><i>indextrm</i></b>	$[[["\varepsilon", 1, 1, "nan", "nan"], ["l", 1, 1, "nan", "nan"],$ $[2, 5, 0, 10, nan], [\beta, 1, 1, nan, nan]]$
<b><i>trm</i></b>	$2\varepsilon\beta l$

In the next step each term is decomposed down to its fundamental sub-terms. Extracting the fundamental quantities from each sub-term is not a straightforward task. Each sub-term must be categorised in one of the 14 defined algorithms, note all these algorithms are placed inside the main *SEEMcore* function. After applying or updating the CEV for each fundamental quantity all the CEVs are multiplied by the corresponding *trm*, and all the  $trm * CEV$  are added together to constitute the output (this being the encoded expression).

Algorithm 4-2 The *SEEMcore* algorithm.

```

1: Function SEEMcore[expression, j, s, k]
2: lhs <- remove encodings of LHS eqn(j);
3: rhs <- remove encodings of RHS eqn(j);
4: if eqn(j) == explicit,
5:   if j == k,
6:     fourthDig <- "nan"
7:   else
8:     fourthDig <- j
9:   end
10: else
11:   fourthDig <- "nan"
12: end
13: Remove number zero from the RHS;
14: indextotal <- Extract all the encodings from both side of eqn(j);
15: for each term in the input expression
16:   indextrm <- exraxt encodings for the term
17:   trm <- remove encodings from the original terms
18:   thirdDig <- epsilon oder
19:   for each sub-term,
20:     if subterm == Power function,
21:       Check PowerFunction;
22:     else
23:       if subterm == Complex function,
24:         Check ComplexFunction;
25:       else
26:         if subterm == none zero integer,
27:           check InetegerFunction;
28:         else
29:           if subterm == Rational function,
30:             check RationalFunction;
31:           else
32:             if subterm == differentiation function,
33:               check DiffFunction;
34:             else
35:               if subterm == Trigonometric function,
36:                 check TrigFunction;
37:             else
38:               if subterm == Complex amplitude,
39:                 check ComAmpFunction;
40:             else
41:               if subterm == normalised coordinates
42:                 check NormCoFunction;
43:             else
44:               check GeneralFunction;
45:             end
46:           end
47:         end
48:       end
49:     end
50:   end
51: end
52: end
53: end
54: end
55: endFunction

```

### *Power functions*

Sub-terms in the form of power functions are decomposed into the base and index parts, and then each part is encoded independently. In this algorithm power functions are categorised into the following groups:

- Exponential functions; the index is in a summation form, Algorithm 4-3.
- Where the base contains differentiation operators, Algorithm 4-4.

- The base is in either a summation or multiplication form, and the index is an integer, Algorithm 4-5.
- The index is a negative integer, Appendix: algorithm 1.
- Exponential; the index is in multiplication form, Appendix: algorithm 2.
- The base is either a summation or a multiplication, Appendix: algorithm 3.
- The base is a derivative of the dependent variable, Appendix: algorithm 4.

### Exponential function where the index is structured in the form of a summation

a) *Where the sub-term has been encoded before and is not modified in the solver.*

This is a very important category because the compound level of the SEEM is applied at this stage. Initially considering Algorithm 4-4, here at first both the imaginary and real parts of the index are separated and stored (lines: 2 and 3). Line 4 checks if the sub-term has been encoded before and this is done by checking the *exponential identity factor* in the *indextrm*. The *exponential identity factor* is only defined for this term category; where the complete exponential sub-term is stored in a string-form in the term structure, *envec*[" $e^{i(a+b)}$ ", "nan", "nan", "nan"] (line 21). If the *exponential identity factor* exists in the *indextrm* this means that the sub-term has been encoded before and its structure remains unchanged. The relevant CEVs are selected from the *indextrm*, and after updating the third digit of encoding this is stored in the term structure (line 5). However if the *exponential identity factor* is not available in the *indextrm* (line 6) this means either that the sub-term has recently been created or that the sub-term has been modified and so the CEVs must be updated.

b) *The sub-term is recently created.*

The compound level of the SEEM term-tracking method is applied between lines 8 to 20 of Algorithm 4-3. The algorithm uses two criteria to decide whether the exponential sub-term has been encoded before, or not. The first condition is defined such that there must be no *exponential identity factor* in the *indextrm*. The second condition is given by the requirement that there must be no CEV with a string form quantity as its second digit in the *indextrm*, line 8. The reason for having the second condition is that in the SEEM term-tracking method the first element of the encoding vector uses a syntax that is stored in string form in a CEV. As Table 4-2 shows, the first element in the encoding vector is the same the second digit of the CEV due to

the presence of the identity factor as the first digit. Therefore, If the second condition is true, a new CEV for  $e$  and  $i$  is introduced which is in the form of  $envec[e^i, " * ", j, "nan", "nan"]$ , line 10. Otherwise, the encoding information must be updated, line 11. Unlike for other encoding levels in the compound level the input equation source is  $j$ , not  $k$ . The reason for this is that the compound level only shows the stages of the analysis where the exponential quantities are subsumed, not the actual source of each quantity. The actual sources can be identified from the CEVs in the index.

Table 4-2 The Source and Evolution Encoding vector format is compared to the computerised encoding. vector.

<b>Encoding vector</b>	$envec[n_1, n_2, n_3, n_4]$
<b>Computerised encoding vector(CEV)</b>	$envec[identity\ factor, n_1, n_2, n_3, n_4]$

*c) Special conditions.*

If the compound level has been defined before and the structure of the sub-term is altered then the *exponential identity factor* is not the same as the sub-term. In this situation the CEV must be updated and the algorithm must identify the CEVs that are related to the latest step of the compound level. To do this the algorithm has been provided with the list of syntaxes that have been used in the compound level in the order of their appearance; \* is the first, then #, etc. The algorithm searches for a CEV with a string form as the second digit and matches the second digit with any of the available syntaxes in the list. In the case where none of the syntaxes in the list match the string from the object, line 15, the first level of compound encoding is introduced.

Both lines 10 and 15 define the compound level for the first time, however there is a key difference between them. Taking the conditions in part (b) into consideration, the first condition is that there is no *exponential identity factor*, while the second condition is that there must be no CEV with a string form quantity as the second digit. In some rare occasions it may be possible for a CEV to have a string form quantity as the second digit and which is not related to the compound level. As the result if a sub-term fails the conditions in part (b), the second condition is again checked in part (c).

*d) The compound level is updated.*

If a sub-term fails all the conditions in both parts (b) and (c), then the compound level must be updated. The algorithm finds the matching syntax in the list, then takes the next syntax as the second digit of the CEV. For example, if the string form syntax is #, then the algorithm finds the

position of # in the list ( [\* , # , \$ , £ , \*\*]), and takes the next syntax, which is \$. The compound encoding level is either defined or updated by this section, and finally the *exponential identity factor* for this sub-term is added, line 21.

e) *Encoding the argument in the index.*

In considering a sub-term of the form of  $e^{i(a+b)}$  so far only the CEVs for  $e$  and  $i$  are encoded. Consequently the argument of the index ( $a + b$ ) must then be encoded and this is not a straightforward task. In order to illustrate the difficulty of automatically updating the encoding information a general overview is provided below:

Considering that every equation created within the solver is also passed to the term-tracker function then the encoded equation is extracted from the term-tracker and passed to the core solver. From that point onwards the encoded equation possibly goes through further complex mathematical procedures that only affect the actual quantities, not the *identity factors* within the CEVs. Therefore the information provided in the *identity factor* is always one step behind the actual quantity itself. The encoding algorithm must know what has happened in the solver in order to identify the previous state of the quantity in the *indexterm*. A series of significant and carefully chosen conditions are defined so that the *SEEMcore* function can successfully identify the state of an input quantity, as follows:

- A new quantity that has not been yet encoded.
- A quantity that has been encoded and it is repeating, with no changes within its *identity factor*.
- A quantity that has been encoded before and has also been modified during the solution procedure, so the *identity factor* does not exactly match the input quantity

It is easily possible for the algorithm to assign a new CEV for a modified quantity that has been encoded before, as the *identity factors* for both conditions do not exist in the *indexterm*. Quantities can change form within different mathematical procedures, such as in cases where:

- Differential equations are solved; i.e. finding solutions of the perturbation equations
- Changing an equation structure from trigonometrical form to polar form, and vice versa
- Defining a new definition for a single quantity and substituting it in to the equation; i.e. the polar form of the complex amplitude  $A$

This issue is graphically explained in Figure 4-4, where every equation defined in the solution procedure is passed to the term-tracker, and after being encoded it is passed to the solver. Here, a mathematical procedure has been applied to the encoded equation resulting in a new equation. Then the new equation is passed to the term-tracker. The term-tracker has no idea what has happened to the equation because it works independently of the solution procedure. It only has knowledge of the previously defined CEVs with a mixture of old and newly defined quantities to encode. The only information that the term-tracker gets is a list of the CEVs which are all one step behind the mathematical operations of the solver. The term-tracker uses this information and evaluates the mathematical procedure and hence updates the quantities accordingly.

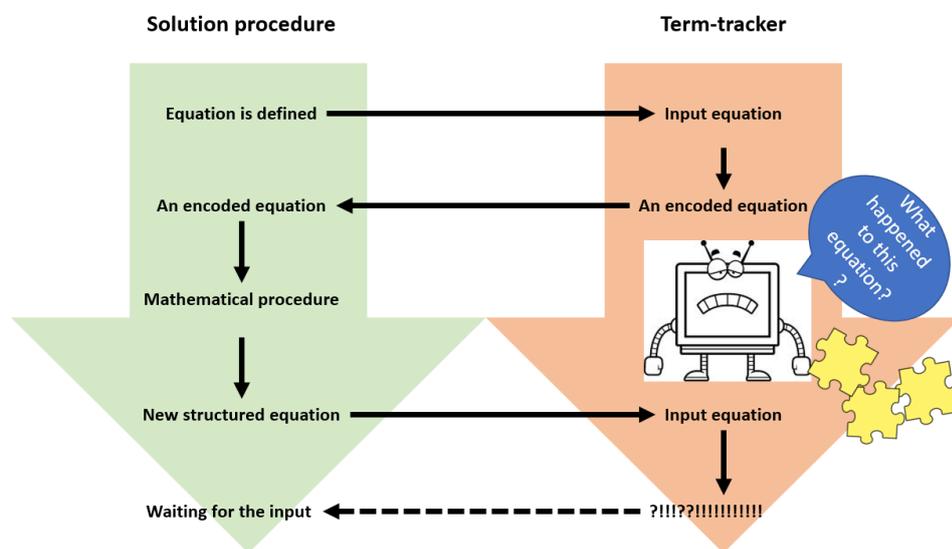


Figure 4-4 The encoded information is always one step behind the solution procedure.

Depending on the function types the algorithm makes a series of estimates to find the previous state of the input quantity in the *indextrm*. Almost the same procedure is used for each function type but the criteria that the *identity factor* selects are different.

Considering line 25, if the argument of the exponential function is in a summation form ( $a + b$ ), each fundamental quantity ( $a$  and  $b$ ) in the index must be encoded separately. For this category, the algorithm assumes that last state of  $a$  and  $b$  was  $e^{ia}$  and  $e^{ib}$ , respectively. Therefore, in line 26 this form is recreated, and it is saved in *NewVar* directory;  $[a, b] \rightarrow [e^{ia}, e^{ib}]$ .

Afterwards the algorithm must find the *identity factor* in the *indextrm* directory for each quantity. In order to do this a series of conditions are created, starting from the highly possible ones to those that are less likely; these conditions are listed in Table 4-3. If none of these options

is available in the *indextrm*, then this quantity is considered to be an unencoded quantity and the first level of encoding is applied, line 30. Otherwise the CEV is updated and the outdated version of the *identity factor* is replaced with the string form of the quantity.

Table 4-3 Criteria that are checked for finding the position of the argument of an exponential function.

Condition number	Identity factor actual → quantity
1	$e^a \rightarrow e^a$
2	$ai \rightarrow e^{ai}$
3	$a \rightarrow e^{ai}$
4	$-a \rightarrow e^{ai}$
5	$e^{-ai} \rightarrow e^{ai}$
6	$e^{-a} \rightarrow e^{ai}$

The statements such as “define a new encoding vector” or “update the encoding vector” appear several times in the *SEEMcore* function. For the first statement it means that a new CEV must be defined; *envec*[*identity factor*, *k*, *order*, *n<sub>3</sub>*,, “nan”]. The *n<sub>3</sub>*, is either equal to “nan” or *j*, as has already been explained. It should be noted that the CEV format is different to that of the CEVs that are defined in the compound level, part (b).

The general procedure needed to update a predefined CEV is summarised in three steps:

By considering the process of updating the CEV (*envec*["*e<sup>z</sup>*", *n<sub>1</sub>*, *n<sub>2</sub>*, *n<sub>3</sub>*, *n<sub>4</sub>*]) for the input quantity of *e<sup>z</sup>* as an example: (it can be noted that the updated encoding digit is shown in a bold font).

- If the CEV forth element ( *n<sub>3</sub>*) is equal to “nan”, then the CEV fourth digit of the SEEM must be updated; *envec*["*e<sup>z</sup>*", *n<sub>1</sub>*, *n<sub>2</sub>*, ***n<sub>3</sub>***, *n<sub>4</sub>*].
- Otherwise, the algorithm, checks if the fifth element ( *n<sub>4</sub>*) of the CEV is equal to “nan”, and the *identity factor* is not the same as the input quantity. In this situation *n<sub>4</sub>* must be updated; *envec*["*e<sup>z</sup>*", *n<sub>1</sub>*, *n<sub>2</sub>*, *n<sub>3</sub>*, ***n<sub>4</sub>***].
- Finally, if none of the above conditions are true, the CEV remains unaltered; *envec*["*e<sup>z</sup>*", *n<sub>1</sub>*, *n<sub>2</sub>*, *n<sub>3</sub>*, *n<sub>4</sub>*].

Finally, all the defined pieces of information in parts (a) to (e) are multiplied by the actual input sub-term and then stored in the equation structure. Going back to the example,  $e^{i(a+b)}$  is encoded as:  $envec[e^i, " * ", j, "nan", "nan"] \times envec["e^{i(a+b)}", "nan", "nan", "nan", "nan"] \times envec[e^a, 16, 0, "nan", "nan"] \times envec[e^b, 16, 0, "nan", "nan"]$ .

Algorithm 4-3 Encoding algorithm for the exponential sub-terms that contain a summation form in the index.

```

PowerFunction1(Line 20 SEEMcore)
1: if base == E and index == expression
2:   im <- imaginary part;
3:   arg <- index;
4:   if subterm part of indexterm,
5:     update encoding vector;
6:   else
7:     remove exponential identity from indexterm;
8:     if (subterm ?????????? indexterm) and indexterm ?????????? {_, String,
'_', '_', '_}
9:       the subterm has not encoded before
10:      eEncode <- envec["subterm", *, i, "nan", "nan"]
11:    else
12:      list <- {*, #, $, £, **};
13:      char <- any possible compound level signs;
14:      if char == {},
15:        eEncode <- envec[e^i, *, i, "nan", "nan"];
16:      else
17:        com <- next character after char in the list;
18:        eEncode <- envec[e^i, com, i, "nan", "nan"];
19:      end
20:    end;
21:    eEncode <- eEncode*envec[subterm, "nan", "nan", "nan", "nan"];
22:  end
23: Remove {_, _String, _, _, _} from the indexterm;
24: for each term in the index:
25:   if term == expression
26:     NewVar <- e^(each quantity in the expression);
27:     for each NewVar
28:       pos <- find the position Newvar
29:       if pos <- {},
30:         Define a new encoding vector
31:       else
32:         update the previously defined encoding vector
33:       end
34:     end
35:   else
36:     pos <- find the position Newvar
37:     if pos <- {},
38:       Define a new encoding vector
39:     else
40:       update the previously defined encoding vector
41:     end
42:   end
43: end
44: end

```

### Power function with a differentiation operator in the base

As mentioned in §4.3.1, the differentiation operator functions are deactivated during the encoding process. The deactivation is done by replacing the *Heads* of built-in derivative functions of *D* (this being the partial derivative) and *Derivative* with *dif* and *diff*, respectively. After the encoding procedure is completed, these functions are updated again.

Algorithm 4-4 shows the encoding procedure for power function sub-terms, where a differentiation operator has appeared in the base; for example,  $diff^2Z$ .

The argument of the differentiation function is extracted ( $Z$ ), line 2. Lines 2 to 8 encode the argument of the function. Then, the actual differentiation operator is encoded in lines 9 to 14. Finally the encoding vectors for both the argument and the differentiation operator are multiplied by the original input sub-term, and placed into the equation structure.

Algorithm 4-4 Power function with differentiations operator in the base

```
PowerFunction2( Line 20 SEEMcore)
1: if there is any differentiation in the base
2:   var <- argumnet of the differentiation
3:   pos <- find postion of var in the indextrm
4:   if pos == {}
5:     out1 <- define a new encoding vector;
6:   else
7:     out1 <- update the encoding vector;
8:   end
9:   pos <- find position of the differentiation operator in the indextrm;
10:  if pos == {}
11:    out2 <- define a new encoding vector;
12:  else
13:    out2 <- update the encoding vector;
14:  end
15:  out1*out2
16: end
```

### Power function where the base is either a summation or multiplication and the index is an integer

This form of sub-term can be shown by generic examples such as  $(a + b)^4$  or  $(a * b)^4$ . As the *SEEMcore* function only deals with the CEVs, then it multiplies the CEVs by the original sub-term, this function is not sensitive to the type of operation be it summation or multiplication. As Table 4-4 shows there is no difference between the CEVs for both sub-terms.

Table 4-4 An example showing that the *SEEMcore* function is independent of the input function operator.

	sub-term	CEVs	Output
(1)	$(a + b)^4$	$envec["a", 3, 0, 34, "nan"]$ $\times envec["b", 1, 1, "nan", "nan"]$ $\times envec["4", 2, 0, 33, 45]$	$(a + b)^4 \times envec["a", 3, 0, 34, "nan"]$ $\times envec["b", 1, 1, "nan", "nan"]$ $\times envec["4", 2, 0, 33, 45]$

(2)	$(a * b)^4$	$envec["a", 3, 0, 34, "nan"]$ $\times envec["b", 1, 1, "nan", "nan"]$ $\times envec["4", 2, 0, 33, 45]$	$(a * b)^4 \times envec["a", 3, 0, 34, "nan"]$ $\times envec["b", 1, 1, "nan", "nan"]$ $\times envec["4", 2, 0, 33, 45]$
-----	-------------	---	--

As the results the base of the input sub-term is decomposed into the fundamental quantities  $[a, b]$ . Then, each fundamental quantity is encoded and the final version of the expression is created and the criteria for identifying the identity factor are given in Table 4-5.

Table 4-5 Criteria that are checked for finding the position of a quantity in the base of a power function.

Number	Identity factor $\rightarrow$ actual quantity
1	$a \rightarrow a$
2	$-a \rightarrow a$
3	any identity factor that includes $a \rightarrow a$

Algorithm 4-5 Power function where the base is either a summation or multiplication and the index is an integer.

```
PowerFunction3(Line 20 SEEMcore)
1: if index == Integer and (base == expression or summation)
2:   out2 <- 1;
3:   out <- original form of the sub-term;
4:   item <- break the subterm into the fundamental quantities;
5:   for each item
6:     if item == imaginary unit number
7:       const <- I;
8:     else
9:       if item == string form
10:        convert it to expression;
11:       else
12:        do nothing
13:       end
14:     end
15:     pos <- position of the item in the indexterm
16:     if pos == {},
17:       out1 <- define a new encoding vector;
18:     else
19:       out1 <- update encoding vector;
20:     end
21:     out2 <- out2*out1;
22:   end
23:   out2
24: end
```

### 4.3.3 Non power function forms

This part of the *SEEMcore* algorithm deals with functions that are not in a power form. The process of encoding the terms is almost the same as has been explained in the previous section.

This section is divided up into the following different functional forms:

- Complex function (Appendix: algorithm 5)

- Integers (Appendix: algorithm 6)
- Rational (Appendix: algorithm 7)
- Differentiation functions (Appendix: algorithm 8 and Appendix: algorithm 9)
- Trigonometric functions (Algorithm 4-6).
- Complex amplitude functions (Appendix: algorithm 10)
- Other functions (Appendix: algorithm 11)

### Trigonometric functions

The process for encoding a trigonometric function is discussed in this section. Considering Algorithm 4-6, at first the argument of the trigonometric function is extracted and stored, line 2. Then the order of the small parameter 'epsilon' is determined, as the order of the argument is not necessary equal to the order of the corresponding term. Considering equation (1) in §3.4.2; the order of this term  $-\underbrace{\varepsilon q \omega^2}_{(1,1)} \cos(\underbrace{\omega}_{(1,0)} T_0)$  is one, however the order of the trigonometrical argument is zero (given that the frequency of excitation in most practical problems will not be definitively small). Afterwards the algorithm checks to see if the argument is a single quantity or not. For a single quantity the encoding process is done as described previously. The only difference is that the order of the argument is used as the first level of the SEEM, lines 4 to 10. For an argument that is structured in other forms (a summation or multiplication), then the argument is decomposed into the fundamental quantities through several stages.

Algorithm 4-6 The trigonometric encoding policy.

```

TrigFunction(Line 32 SEEMcore)
1: if subterm == Trig
2:   arg <- argument of the trig function without time
3:   argorder <- find the order of the argument
4:   if arg = 1= summation or expression
5:     pos <- position arg
6:     if pos == {}
7:       out <- define a new encoding vector
8:     else
9:       out <- update the encoding vector
10:      end
11:    else
12:      for each arg
13:        var <- remove time elements
14:        if var == expression or power function
15:          if var == powerfunction
16:            NewVar <- list base and index
17:          else
18:            NewVar <- list var
19:          end
20:          for each NewVar
21:            if NewVar == String
22:              convert the string from into expression;
23:              pos <- find the position of NewVar in indextrm;
24:            else
25:              if NewVar == Power
26:                pos <- find the position in indextrm;
27:              else
28:                pos <- find the position of in indextrm;
29:              end
30:            end
31:            if pos == {}
32:              out1 <- define a new encoding vector;
33:            else
34:              out1 <- update the encoding vector;
35:            end
36:            out <- out*out1;
37:          end
38:        end
39:        if var == complex
40:          check complexfunction
41:        else
42:          if var == ineteger
43:
44:            check integerfunction
45:          else
46:            check RationalFunction
47:          end
48:        end
49:        out= out*out1
50:      end
51:    end
52:  out
53: end

```

#### 4.3.4 Summary of the encoding function

The SEEM term-tracking method is computerised by defining two main functions; the *SEEMencoder* and the *SEEMcore*. As Figure 4-5 shows, the *SEEMencoder* is a user-end function, and it deactivates the differentiation functions for the input equation and prepares the expressions to be the input of the *SEEMcore* function. Then the *SEEMcore* function decomposes the expressions into the fundamental quantities, after which the CEVs for all the quantities are

defined or updated and stored in the term structures as multipliers. The encoded expression is then passed to the *SEEMencoder* function, and after activating the differentiation operators the encoded equation is sent to the solver.

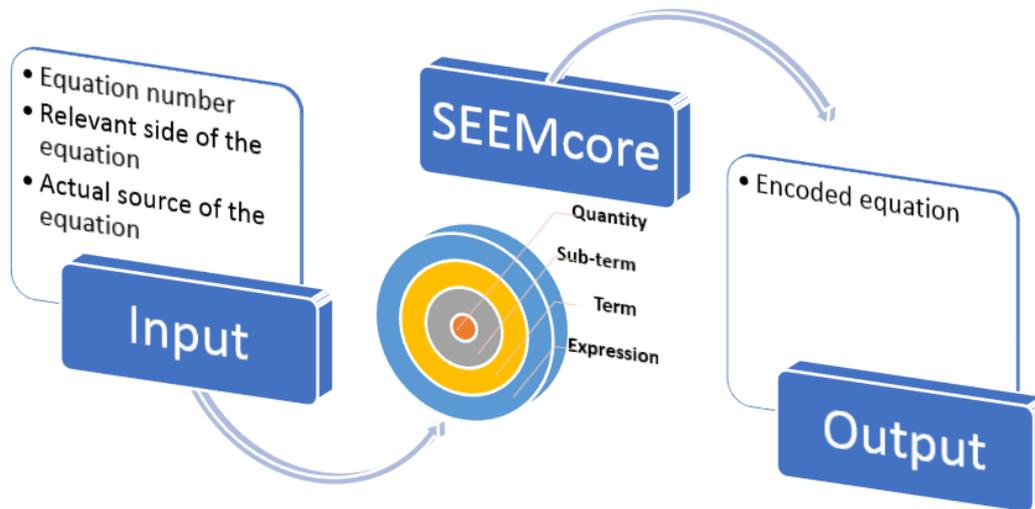


Figure 4-5 Graphical representation of the relationship between the *SEEMencoder* and the *SEEMcore* functions.

This function is computationally efficient because the information that is created during the encoding process is thereafter erased, after the input equation has been passed to the solver. In other words, all the necessary information is stored inside the input equation structure, and there is no requirement for a separate database.

This is an important analogy, as it creates flexibility for further developments, meaning that it is possible to define different post-processing functions to extract encoding information and then present them in a desired format. In this study several post-processing functions are developed for the purposes of displaying, visualising, and term evaluation.

#### 4.4 Displaying the encoding information

The *SEEMencoder* function applies the SEEM term-tracking method to the equations during the solution procedure. The encoded equations are like a row of data, and they must be processed in order to have a meaningful format for the user. An example of the output of an encoded equation is provided in Figure 4-6. The analytical solution procedure of a nonlinear problem is complex in its own right, so adding in the encoding information, either above or below the quantities, only serves to escalate this complexity. As a result, a practical method must be used

to display the encoding information in the SCD solver and this section reviews possible methods of display.

```

: eqn[1] =
  ExpandAll[
    X'[t] + ω12 X[t] -
    ε (f0 Cos[Ω t] + μ (Y'[t]2 + Y''[t] * Y[t]) - 2 ξ1 ω1 X'[t] +
    X'[t] * X'[t] γ) == 0];
: SEEMencoder[1, 1, 1]
: -f0 ∈ Cos[t Ω] envec[-1, 1, 1, nan, nan]
  envec[-1, nan, nan, nan, nan] envec[Cos[t Ω], 1, 0, nan, nan]
  envec[f0, 1, 1, nan, nan] envec[ε, 1, 1, nan, nan] +
ω12 envec[ω12, 1, 0, nan, nan] X[t] + 2 ε ξ1 ω1 envec[2, 1, 1, nan, nan]
  envec[2, nan, nan, nan, nan] envec[ξ1, 1, 1, nan, nan]
  envec[ω1, 1, 1, nan, nan] envec[ε, 1, 1, nan, nan] X'[t] -
γ ε envec[-1, 1, 1, nan, nan] envec[-1, nan, nan, nan, nan]
  envec[γ, 1, 1, nan, nan] envec[diff[1][X][t]2, 1, 1, nan, nan]
  envec[ε, 1, 1, nan, nan] X'[t]2 - ε μ envec[-1, 1, 1, nan, nan]
  envec[-1, nan, nan, nan, nan] envec[μ, 1, 1, nan, nan]
  envec[diff[1][Y][t]2, 1, 1, nan, nan] envec[ε, 1, 1, nan, nan] Y'[t]2 +
X''[t] - ε μ envec[-1, 1, 1, nan, nan] envec[-1, nan, nan, nan, nan]
  envec[μ, 1, 1, nan, nan] envec[ε, 1, 1, nan, nan] Y[t] Y''[t] == 0

```

Figure 4-6 The output of the *SEEMencoder* function format, for the equation of motion of in-plane response of the problem of the autoparametrically excited beam system.

#### 4.4.1 The Tooltip function

The *Tooltip* function is commonly used in different programming languages. It displays extra information attached to an element, usually activated when a user hovers the mouse pointer around that element. In the *Mathematica* language this function is readily defined and logically named as *Tooltip* [71]. This function in general has two inputs: the main element and the label. An application of this function is demonstrated in Figure 4-7. In this example the arbitrary encoding vectors are assigned to quantities inside an expression: *Tooltip*[quantity, "encoding vector"]. The output is displayed like a normal mathematical expression until the user decides to have more information about a particular quantity. If the user hovers the mouse pointer around the highlighted areas in the Figure the arbitrary encoding information will be displayed below the quantities and then the information disappears as the mouse pointer moves away.

$$\text{Tooltip}[x^2, "(1,0,12,30)"] * \text{Tooltip}[b, "(1,0)"] +$$

$$\text{Tooltip}[4, "(5,1,15)"] * \text{Tooltip}[x, "(1,0,13,34)"]$$

Figure 4-7 A graphical demonstration of the application of the *Tooltip* function for an arbitrary encoded expression; the highlighted area shows the mouse pointer position.

The main advantage of this method is, it does not make the appearance of the solution procedure more complex. Forehand and Cartmell [7] applied the *Tooltip* function to an SCD solver to display the SEM term-tracking information. The dependency on the specific computerised environment can be considered as the main downside of this method. The information stored in this function only can be displayed using the mouse pointer, when printed only the main element is printed without the label.

#### 4.4.2 Labeled function

For documentation purposes it is essential to have the encoding information in a printout version. One possible option is using the built-in *Labeled* function of the *Mathematica* interface. The input form of this function is the same as the *Tooltip*; comprising an element and a label. Unlike the *Tooltip* all the encoding information is represented underneath each quantity. Considering Figure 4-8, the label is displayed readily in the output but this does make the output visually more complex.

$$\text{Labeled}[x^2, "(1,0,12,30)"] * \text{Labeled}[b, "(1,0)"] +$$

$$\text{Labeled}[4, "(5,1,15)"] \text{Labeled}[x, "(1,0,13,34)"]$$

Figure 4-8 A graphical demonstration of the application of the *Labeled* function for an arbitrarily encoded expression.

#### 4.4.3 Display format options

The advantages of both the *Tooltip*, and *Labeled* functions are required on various occasions. As the results it is essential for the user to have the flexibility of using either of these methods in a given situation; perhaps using the *Tooltip* function when working in a computerised environment and the *Labeled* function for documentation purposes. From the programming point of view it is fairly easy to computerise this idea, as the inputs of these two functions are

the same, and only the *Heads* are different. Basically the main displaying post-processing function can process the quantities and encodings in the required format:  $nhead[quantity, encoding]$ , where  $nhead$  is a neutral head. As Table 4-6 shows by selecting a relevant option it is possible to replace  $nhead$  with either of these built-in functions.

Table 4-6 Display format option structures.

Display type number	Operation	Output
1	$nhead \rightarrow Tooltip$	$Tooltip[quantity, encoding]$
2	$nhead \rightarrow Labeled$	$Labeled[quantity, encoding]$

#### 4.4.4 DisplaySEEM function

The *DisplaySEEM* is a user-end function and has been developed to process the encoding information embedded in equation structures. This function has two inputs: the equation number that needs to be displayed and the display type number. As Algorithm 4-7 shows the differentiation operators are disabled and each side of the input equation is passed to the *CoreDisplayer* function. Afterwards the differentiation operators are activated for the outputs of *CoreDisplayer* function, on each side of the equation.

Algorithm 4-7 The *DisplaySEEM* function.

```

DisplaySeem[i,d]
1: if LHS eqn(i) == function
2:   deactivate the functions and send the LHS to display function
3:   else
4:     send the LHS to display function
5:   end
6: end
7: if RHS eqn(i) == function
8:   deactivate the functions and send the RHS to display function
9:   else
10:    send the LHS to display function
11:  end
12: set the encoded the LHS equal to the RHS

```

#### 4.4.5 CoreDisplayer

The main processing for displaying the SEEM term-tracking method is carried out in this function, Algorithm 4-8. The input of this function is an expression which is processed by the *DisplaySEEM* function. The general process of the *CoreDisplayer* function can be compared to the *SEEMcore*, as the expression structure is decomposed into fundamental quantities and the encoding information is extracted, layer by layer. The main difference is that the *identity factor* for each quantity always matches the input quantity, because there is no mathematical

procedure between the output of the *SEEMencoder* function and input of the *DisplaySEEM* function.

The displayer function must extract the relevant pieces of encoding information and assign them to the actual quantities, without affecting the format of the equation itself. Unlike the encoding procedure in which the CEVs were independent of the input function operator, as discussed in Table 4-4 the operators in the display function are quite important.

To summarise, after the sub-term is decomposed into the fundamental quantities the position of each quantity in the *indexterm* is located. If no *identity factor* is found it means that there is no CEV for that quantity, and the algorithm defines an empty encoding vector. Otherwise the algorithm converts the CEV into the output format. For example, considering the steps that the algorithm takes to process the encodings for *a* in the input sub-term of  $3ax$ :

- 1) The function takes the CEV from the *indexterm*;  $\rightarrow [a, 1, 0, 32, "nan"]$ .
- 2) Then it removes the empty cells  $\rightarrow [a, 1, 0, 32]$ .
- 3) After this it removes the identity factor  $\rightarrow [1, 0, 32]$ .
- 4) It then changes the formatting  $\rightarrow (1, 0, 32)$ .
- 5) From there it converts the output into the string format  $\rightarrow "(1, 0, 32)"$ .
- 6) The displaying option *nhead* $[a, "(1, 0, 32)"]$  is created.
- 7) Finally it replaces the input quantity in the sub-term with the displaying output  $3ax \rightarrow 3nhead[a, "(1, 0, 32)]x$ .

The same process as listed in steps 1 to 7 above is taken for all the quantities in that term, and at the last stage the *nhead* is replaced with the selected display type; the *Tooltip* or the *Labeled* functions. It can be noted that the sub-terms are decomposed in exactly the same order as the *SEEMcore* function

Algorithm 4-8 The algorithm for *CoreDisplyer* function

```

display[expr]
1: for each term
2:   indextrm <- extract encoding information from the term structure
3:   trm <- removing the encoding information from the term
4:   for each subterm
5:     if subterm == power function
6:       Check powerdisplay
7:     else
8:       if subterm == Rational
9:         check Rationaldisplay
10:      else
11:        if subterm == differentiation
12:          check diffdisplay
13:        else
14:          check diff2display
15:          if subterm == complex
16:            check complexdisplay
17:          else
18:            if subterm == trig
19:              check tringdisplay
20:            else
21:              if subterm == integer
22:                check integerdisplay
23:              else
24:                check other functions
25:              end
26:            end
27:          end
28:        end
29:      end
30:    end
31:  end
32: end
33: end

```

## 4.5 Symbolic Computational Dynamics solver

The SCD works like a digital interactive notebook in the *Mathematica* interface. This layout provides an essential level of flexibility for the user so that s/he can apply any assumption or simplification, where it is required. It is possible for the user to define a separate equation for introducing new mathematical concepts and even to add notes. A sample of a SCD notebook is shown in Figure 4-9, this notebook is divided into six main sections. It is possible to open each section by clicking on the highlighted area. A brief explanation for each part of this SCD notebook is provided in this section.

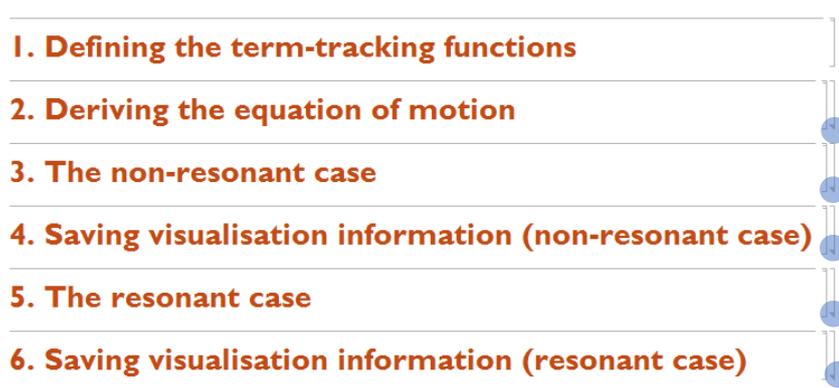


Figure 4-9 A graphical representation of the Symbolic Computational Dynamic notebook.

#### 4.5.1 Section 1: Defining functions

All the encoding and post-processing functions, which are explained in §4.3 and §4.4, are defined in the first section of the SCD, see Figure 4-10. This provides flexibility for the user to check the function structure and to modify the conditions, if this is required. As shown before the detail of the functions can be shown if the user clicks on the triangle sign on the right-hand side of the menu.



Figure 4-10 The encoding and post-processing functions defined at the beginning of a given Symbolic Computational Dynamic notebook.

#### 4.5.2 Section 2: Defining the equation of motion

In section 2 of a given SCD notebook (Figure 4-9), the procedure that has been carried out to derive the equation(s) of motion for the system is explained and the assumptions are listed. This part of the SCD is not currently automated and so the user can add in these pieces of information for documentation purposes. In future it may be possible to build in the *MultiFlex.m* [9] function in order to automate the symbolic derivation of the equations of motion.

### 4.5.3 Section 3: The non-resonant condition

The actual solution procedure starts in section 3 of the given SCD notebook. The analysis starts by introducing the symbolic form of equation of motion is defined by the user. The input equations are defined by using a head of  $eqn[i]$ , where  $i$  is the equation number. As Figure 4-11 indicates, the user defined equation 1 by using the desired symbols. It is possible to compare this equation to equation (1) in chapter 3.

$$\begin{aligned}
 eqn[1] = & \\
 & \text{Expand}[\theta''[\tau] + 2_{dp} \epsilon_{dp} \beta_{dp} \theta'[\tau] + (1 - \epsilon_{ex} q_{ex} \omega^2 \text{Cos}[\omega \tau]) \theta[\tau] + \\
 & \quad \epsilon_{ns} \gamma_{ns} \theta[\tau]^3] == 0 \\
 & \theta[\tau] - \omega^2 \text{Cos}[\tau \omega] q_{ex} \epsilon_{ex} \theta[\tau] + \gamma_{ns} \epsilon_{ns} \theta[\tau]^3 + 2_{dp} \beta_{dp} \epsilon_{dp} \theta'[\tau] + \theta''[\tau] == 0
 \end{aligned}$$

Figure 4-11 A screen-shot from a defined input equation in a Symbolic Computational Dynamics solver.

The user defines different symbols for the small parameter ‘epsilon’ within the different terms. For example, the small parameter that scales the external excitation amplitude is symbolised by  $\epsilon_{ex}$ , while the small parameter that scales the damping term is symbolised by  $\epsilon_{dmp}$ . In equation (1) in chapter 3, all these parameters are shown by  $\underset{(1,1)}{\epsilon}$ , which makes it difficult for the user to

distinguish between these parameter in later parts of the analysis. These  $\epsilon$ s in the standard multiple scales are considered to be numerically equal, however in reality they invariably have different physical sources.

Figure 4-12 is a screen-shot of the SCD solver where the user, after introducing the symbolic form of the equation of motion, has decided to add some text to help with documentation purposes. Afterwards the equation is sent to the *SEEMencoder* function: noting that the inputs are [1,1,1]: the input equation number is 1, the user wants to encode the LHS of the equation, and the actual equation source is the same as the input (1). As discussed previously, the output of this function requires post-processing, so it is suppressed by using “;”. Finally the user decides to display the equation, so the *DisplaySEEM* function is used. The first input is the equation number (1), and the second input is the displaying type, for which in this case the *Tooltip* method was selected (1). In this figure, the mouse pointer is hovering around  $\omega$ , so (1,0) is shown.

```

eqn[1] =
  Expand[ $\theta''[\tau] + 2\epsilon_{dp} \epsilon_{dp} \beta_{dp} \theta'[\tau] + (1 - \epsilon_{ex} q_{ex} \omega^2 \cos[\omega \tau]) \theta[\tau] + \epsilon_{ns} \gamma_{ns} \theta[\tau]^3$ ] == 0;

```

As equation 1 is a fundamental equation the left hand side is encoded. It is possible to observe that encoding vectors are assigned to each variable. Also, the dependent variable  $\theta[\tau]$  have not been encoded. The output is suppressed by using “;”.

```
EncodeSeem[1, 1, 1];
```

```
DisplaySeem[1, 1]
```

```

 $\gamma_{ns} \epsilon_{ns} \theta[\tau]^3 + \theta[\tau] - \omega^2 \cos[\tau \omega] q_{ex} \epsilon_{ex} \theta[\tau] + 2\epsilon_{dp} \beta_{dp} \epsilon_{dp} (\theta')[\tau] + (\theta'')[\tau] == 0$ 
(1,0)

```

Figure 4-12 A screen-shot from the SCD solver; how equations are defined, encoded and displayed.

Every small step or assumption is shown and explained directly in the SCD, so the user is fully informed and can interact where it is required. In Figure 4-13 an example is shown whereby a computational procedure must be performed in the SCD solver. Instead of hiding this procedure from the user the complete code is written within the documentation. The original solver of Forehand and Cartmell [7] showed the template for the non-resonant response and the only missing parts were the solutions for the modulation equations. These parts are included in this study.

```

diff[a_ + b_, x_] := diff[a, x] + diff[b, x];
diff[ $\epsilon_p^{n-}$  a_, x_] :=  $\epsilon_p^n$  diff[a, x];
diff[diff[a_, x_], y_] :=
  diff[a, Sort[{x, y}][[1]], Sort[{x, y}][[2]]];

temp1 = Sum[ $\epsilon_p^i$  diff[#, timeScales[[i+1]]], {i, 0, maxOrder}];
temp2 = head[temp1] /. head -> Function;

eqn[3] = dt[1] == temp2 /. {diff -> D};

EncodeSeem[3, 2, 3];

DisplaySeem[3]

dt[1] == D0 + D1  $\epsilon_p$ 

```

Figure 4-13 An example showing the transparent structure of the SCD solver.

## 4.6 Summary

In this chapter the process of computerisation of the SEEM term-tracking method has been explained in some detail. The *Mathematica* programming language is used throughout due to its extremely powerful capabilities for symbolic computation. It would obviously be possible to programme the developed algorithms in any other language, but it should be borne in mind that the main criteria for the algorithms created here have been generality and flexibility. Examples of these requirements can be identified in the logistics of all the functions in this chapter. For example, the computerised encoding vector (CEV) is designed to have a flexible length so it has the potential for future extension. Two separate functions for the encoding purposes are developed. Firstly the *SEEMencoder* is presented, where this is a user-end function that transforms the input equation to the required format for the *SEEMencoder* function. The *SEEMencoder* applies 'the SEEM cycle' to each expression of the input equation. The pieces of encoding information are saved within a format of strings and numbers so that they do not have any interaction with the actual solution procedure. One form of post-processing the encoded equations was introduced as the *DisplaySEEM* function. This function has the flexibility for adjusting the display function type; this being either the *Tooltip* or *labeled* function. It is therefore possible to summarise the findings of this chapter in the following points:

- The method of multiple scales has been used for generating the approximate analytical solution due to its flexibility and adaptability.
- The whole SCD solver has been programmed in the *Mathematica* interface.
- The process of the SEEM encoding cycle has been explained in some detail, Figure 4-3.
- The Computerised Encoding Vector (CEV) has been introduced, Table 4-2.
- The inherent obstacles that have been encountered in the automation of the SEEM encoding have been summarised, Figure 4-4.
- A summary of the computerisation of the SEEM term-tracking method has been given, Figure 4-5.
- An overview of the SCD solver has been provided.
- Overall, 28 algorithms have been developed in this study.



# Chapter 5 Visualisation Methods

## 5.1 Introduction

A novel proposal for a computationally implementable term-tracking methodology in the form of the Source and Evolution Encoding Method (SEEM) was fully introduced in chapter 3 of this thesis. The computerisation and implementation of this method into a Symbolic Computational Dynamics (SCD) solver was discussed in detail in chapter 4. Post-processing functions are henceforward required to extract the SEEM term-tracking information from the encoded equations, and to present this in a meaningful, high-impact format for the user. In chapter 4, *DisplaySEEM* was introduced as a computational process for displaying the encoding information either in the form of a *Tooltip* or as *Labeled* functions. It has already been established that a nonlinear solution procedure is computationally complicated in its own right so displaying extra information in and around the equations adds a considerable and challenging layer of additional computation to the process. Therefore, a robust visualisation method which enables the user to interact with the SEEM information is required. Figure 5-1 shows a graphical summary of the development of the SCD method in this thesis. It is possible to notice the classical hand-written solution procedure in chapter 3 was developed to a computerised format in chapter 4. This chapter reviews the process of developing the very first generation visualisation method for a practically implementable SCD solver.

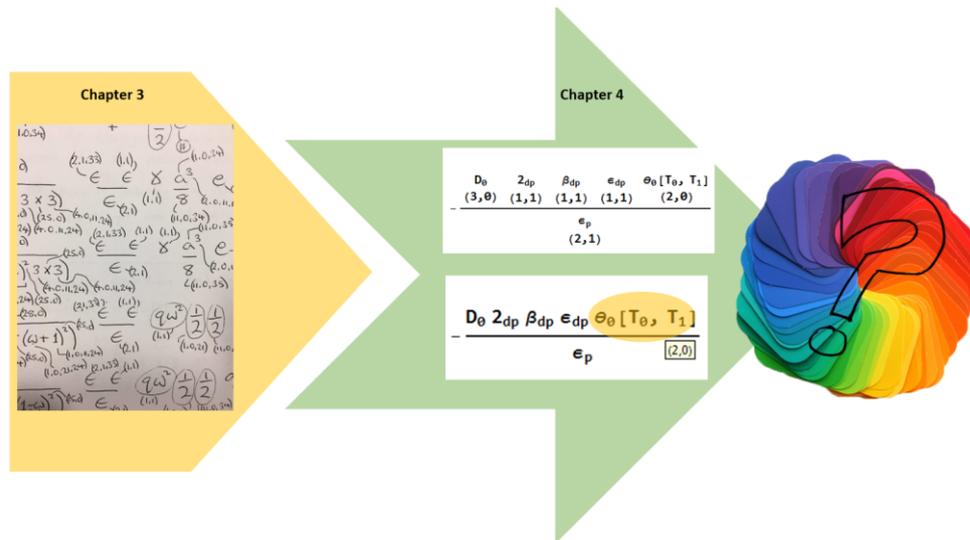


Figure 5-1 A graphical review of the development of Symbolic Computational Dynamics in chapter 3 and 4.

## 5.2 Successful visualisation criteria

The core solver and the term-tracker can be considered as two independent procedures that are exchanging information. The SCD solver is intended to generate the symbolic form of the solution procedure and the term-tracker is intended to display the encoding in a practically useful format, in a form that is potentially highly customisable and most informative. The novelty of the work described within this chapter is the efficient computation required to achieve the foregoing aims and the new graphical results that have emerged from. It is possible to combine the solution procedure and the term-tracking results in a user-definable single graph form. The essential properties of a successful visualisation method are as follows:

- It should have the ability to show all the SEEM term-tracking information in an analysis.
- It can highlight the position of each quantity inside the relevant equation; it is possible for an equation to have the same quantity appearing in different terms. It is important for the user to distinguish the specific source of the quantity, not just the equation number.
- A clear identification of the path for each quantity that is taken from its source to the final response. This could help the user to follow the path for each assumption or physical concept in a given analysis.
- Enough information should be provided about various stages of the analytical analysis.

Considering the above conditions, two different visualisation methods are suggested: the *ColourMap* and the *Blueprint* methods.

### 5.3 The *ColourMap* visualisation method

The *ColourMap* is the first generation visualisation method that was proposed for use with SCD solvers, Figure 5-2. The information assembled in an SCD analysis is visualised in the form of a table, where all the quantities in the analysis are listed in the first row of the table, and the equation numbers are given in the first column. The encoding information is colour coded; in this figure the zeroth-order quantities are shown in blue, the first order quantities in pink, and the second level of the SEEM in green.

Eqn	$\delta$	$\epsilon$	$\theta$	$\zeta$	$\omega$	$\lambda$	$\beta$	$\epsilon$	$\theta$	$\rho$	$e$	$D$	$\rho$	$\lambda$	$\epsilon$	$D$	$D$
1							Blue										
2									Blue								
3																	
4																	
5																	
6																	
7																	
8																	

Figure 5-2 Initial overview of the *ColourMap* visualisation method.

Then in this study some modifications have been applied to the method to increase its functionality. This comprises the use of different shapes presenting all the SEEM encoding levels, for example the zeroth-order terms are shown with an oval shape, while the first order quantities are shown using a rectangular shape. Furthermore, distinct colours are used to represent the different encoding levels; the first level is shown in purple, second level in blue, etc. To distinguish between various stages of an analytical analysis, the quantities in each equation are given in a single row. This method is computerised and implemented into the SCD solvers by means of the *ColourMap* function. The colours and shapes allocated to the SEEM information can be altered for the convenience of the user. Considering Figure 5-3, the quantities in equation (1) are shown in the first row, the quantities for the second equation are shifted to the second row and so on.

$\theta$	-1	$\text{Cos}[t \omega]$	q	$\omega^2$	$\epsilon$	$\gamma$	$\theta[t]^3$	2	$\beta$	$\text{Cos}[T_0 \omega]$	$e^{-1}$	$e^i$	$e^{-3i}$	$e^{3i}$	1	$e^\omega$	$e^{-i\omega}$	$e^{i\omega}$	i	$\omega$					
1	■	●	■	■	■	■	■	■	■																
2																						●	■	■	
3																									
4																									
5																									
6	■		■	■	■	■		■	■	●														■	●
7																									●
8	■		■	■	■	■		■	■	●															●
9											●	●													
10											●	●	●	●											●
11											●	●			●	●									
12	■		■	■	■	■		■	■		●	●	●	●	●	●	●	●							
13																									
14	■		■	■	■	■		■	■																
15	■		■	■	■	■		■	■		●	●	●	●	●	●	●	●							
16	■		■	■	■	■		■	■	●	●	●	●	●	●	●	●	●	●	●	●	■			
17																									
18																									
19																									
20																									
21	■				■	■		■	■																
22	■				■	■		■	■																
23															■										

Figure 5-3 A graphical representation of the *ColourMap* visualisation method.

There are several possible advantages to using the *ColourMap* visualisation method such as the possibility for quick recognition of various stages of the analysis where a single quantity appears and disappears. Also the SEEM term-tracking levels can be highlighted by adjusting the shapes and colours of each cell in the *ColourMap*. Finally, the user can identify the exact source of each quantity with relative ease.

Despite these advantages this method still fails to highlight the position of each quantity in an equation. Also information relating to a repeating quantity in an equation is not available. Furthermore this method, as conjectured, fails to show the path that a quantity has taken from the beginning to the end of an analysis. Finally, the table size is likely to grow exceptionally large for a complex problem. Therefore this method has been rejected and no further developments of it are considered in this research.

### 5.4 The *Blueprint* visualisation method

By considering the necessary criteria that have been identified for a robust visualisation, a new and somewhat more comprehensive and inclusive visualisation method can be proposed. In this further attempt the principal aim is to show all the sources and connections between the quantities in the solution procedure, like a map. In other words a blueprint for the solution procedure is developed. The initial idea of the *Blueprint* visualisation method is shown in Figure 5-4. In this visualisation format each equation is shown in a line, and each term in the equation is allocated a block in the equation line. The interactions of the quantities are highlighted with lines between different blocks. It is possible to change the lines colours and patterns to show different encoding information.

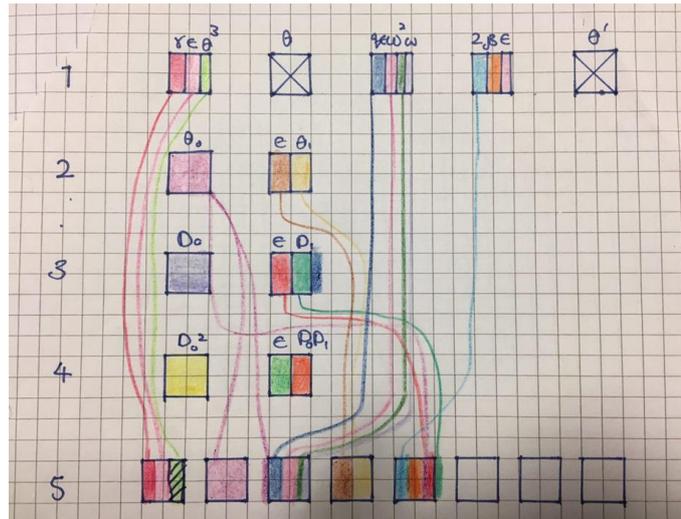


Figure 5-4 A graphical representation of the initial idea for the *Blueprint* visualisation method.

This method has several advantages, such as the possibility for highlighting the path for each quantity from its source to the ultimate point of the analysis. This is essential in case the user wants to follow a given assumption through the solution procedure. Also it is possible to use a unique colour shade for each system element (such as damping for example) so the user can visualise how the damping terms (or other system elements) are distributed within the solution procedure. Damping characteristic of systems has a considerable effect on the response of the system, as energy dissipates through damping. Generally stronger excitation results a higher response amplitude and wider resonant region. However, small excitation in resonant condition can cause high amplitude response even with significant damping (5%). This is also noticed in parametric excitation where small excitation can cause large response in presence of damping.

Furthermore, it shows the position of each quantity in the equations so that the user can distinguish between similar quantities in different terms. More importantly, the steps of the solution procedure are identifiable in an analysis. The SEEM information can be highlighted by assigning a unique colour or pattern to the connecting links.

The main disadvantage of this method is that the number of links increases with the complexity of the analysis. This can be prevented by defining controlling functions for the visualisation interface, such as temporarily hiding information that is not required, zooming in on that which is of particular interest, or also showing only a particular quantity or element. It can be noted that element here means mechanical system elements (mass/inertia, damping, etc.) and quantity means any symbol used in the equation structures.

## 5.5 Computerisation of the visualisation method

The visualisation method is implemented within the SCD solver by defining *ColourCode* and *Blueprint* functions. This section summarises the development process relating to the *Blueprint* function.

### 5.5.1 Automated panel

The first challenge in the computerisation of the *Blueprint* was to find a way to assign colours and patterns for different levels of the SEEM encodings. In the early attempts a panel as shown in Figure 5-5 was developed so that the user could select different options before running the code. These options are described as the function notebook, the input core solver notebook, and setting the specifications for each level of the SEEM term-tracking method. For example, the user can select the predefined functions notebook and press run functions. Then the solution procedure notebook, where the problem is defined, must be run.

After a few minutes of running the solver the visualisation panel would be populated and made visible to the user. There were found to be several issues with this form of visualisation; firstly the solution procedure and functions were hidden from the user. The second issue was that when changing the visualisation settings the user would necessarily have to re-run the program. These two issues are against to the main philosophy of the SCD solver - flexibility and control. As the results, this form of visualisation is not considered for the main function.

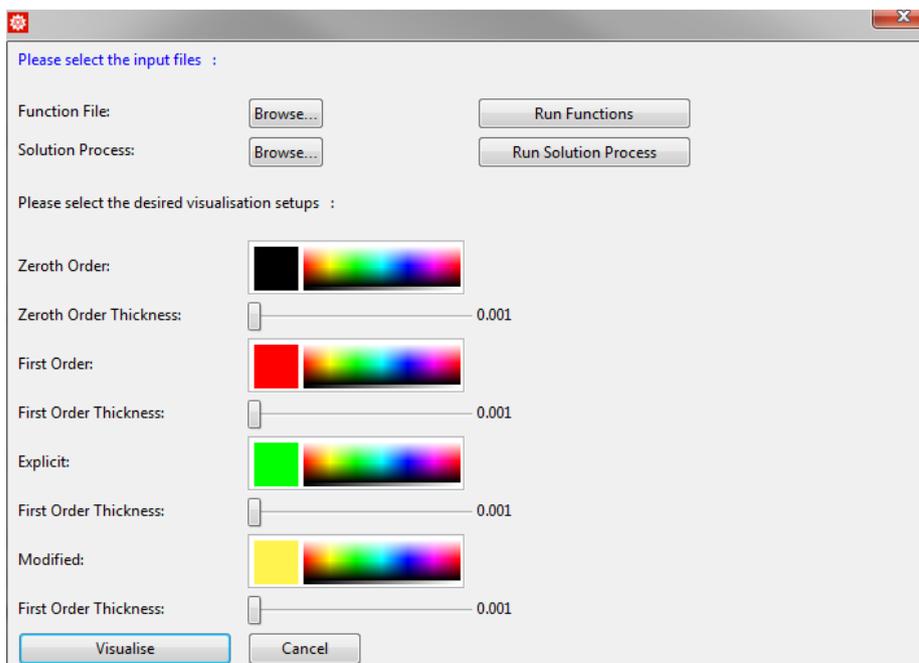


Figure 5-5 User panel for setting up the visualisation method.

### 5.5.2 Two-dimensional panel

The first version of the *Blueprint* was in two-dimensional (2D) form. A control panel was defined on the left-hand side (LHS) of the visualisation interface, Figure 5-6. The user could set the equation number with the first selection bar, and the solution procedure would run based on the settings input to the system by means of this panel. Any changes in the setting, the equation number, encoding colours, or a wish to change fonts, would all require the re-run of the solver process.

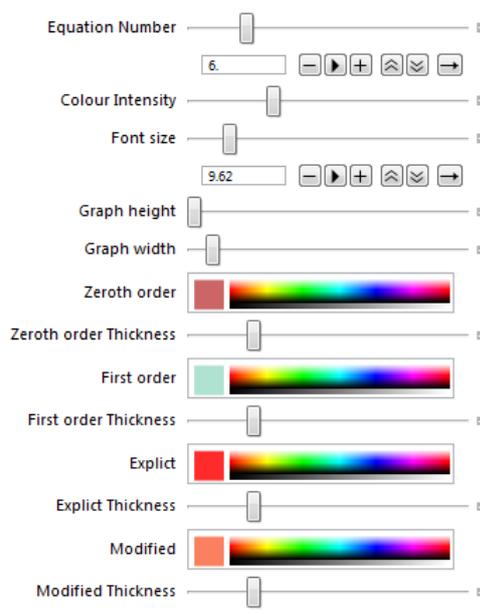


Figure 5-6 Control panel of the visualisation interface.

Figure 5-7 is an example of the output of the panel noting that there is a specific line for each equation and the detail for the equation is stated besides the equation number. The pink coloured links representing zeroth-order are defined to be denser than the blue links which depict the first order. The colour intensity of the links is adjustable so that the user can hide the links in order just to have the symbolic form of the solution procedure.

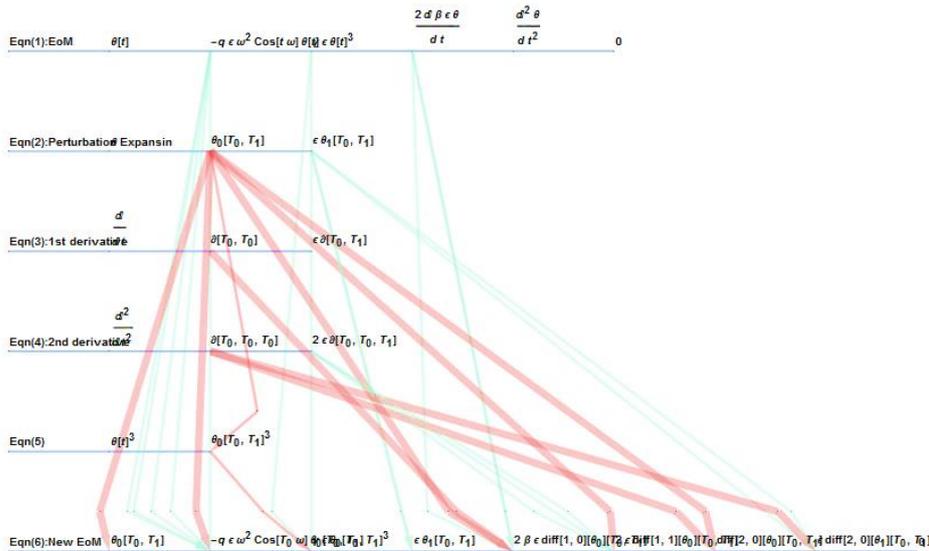


Figure 5-7 The output of the two-dimensional *Blueprint* visualisation method.

### 5.5.3 Special considerations

As the analysis progresses the number of links increases and it quickly becomes essential to have an alternative approach to display the SEEM information. It was therefore decided to add the encodings in the form of either the *Tooltip* or *Labeled* displaying function on each link. Implementing this facility allowed the user to be able to hover the mouse pointer around a particular link to visualise the quantity and its associated encoding vectors. This seems to be readily achievable using *Mathematica* built-in functions; however, as Figure 5-8 shows, there is a bug in a specific application of this function. One can define this problem by considering that  $a$  and  $b$  represent two terms in two separate equations, and two different quantities from term  $a$  are linked to the term  $b$ . As visible in the graph, the colour assigned to the link  $a \rightarrow b$  is red and the tag is  $A$ , while for the other link the colour is blue and the tag is  $B$ . Due to the *Mathematica* bug these different links are shown as being the same.

```
Graph[{Labeled[Style[a -> b, Red], "A"],
      Labeled[Style[a -> b, Blue], "B"]}, EdgeStyle -> Thick,
      VertexLabels -> "Name"]
```

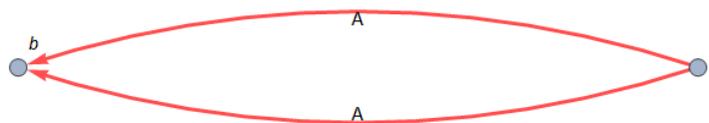


Figure 5-8 *Mathematica* bug in visualising different link tags.

To fix this issue a middle point between each links is created as shown in Figure 5-9. Therefore, in the visualisation function for each defined link, a connection point must be defined automatically, and its coordinates must be adjusted to the source and destination links.

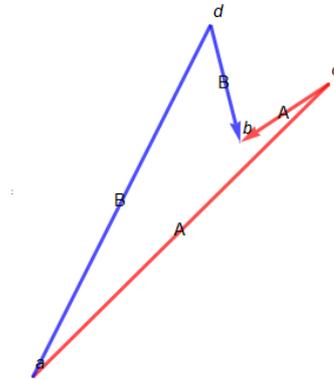


Figure 5-9 Solving the *Mathematica* bug in tagging links.

#### 5.5.4 The three-dimensional Blueprint visualisation method

The next phase in the development of the visualisation was to create a three-dimensional (3D) version of the *Blueprint* method. This version enables the user to have a productive interaction with both the SEEM encoding and the analytical results and comparing to that of the 2D version the control panel functionality is significantly more advanced. As Figure 5-10 shows the user can select the desired equation number and extract the development pattern for that specific equation. The results of the analysis are stored in a database in order to enhance computational efficiency, and there is no need for the solution procedure to be re-run for every small modification made at the control panel. Finally, it is possible for the user to extract the generated graph by pressing the 'save graph as .pdf' button.

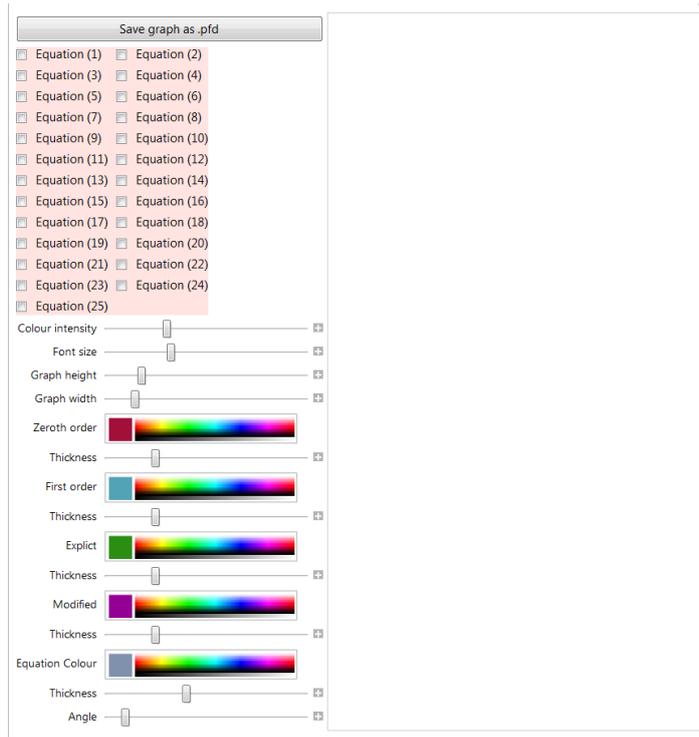


Figure 5-10 The three-dimensional *Blueprint* visualisation panel.

Figure 5-11 is an example of the output of the visualisation graph, for the problem of the parametrically excitation of a pendulum. This figure is provided just for showing the overall interface of the solver and a detailed presentation is provided in this chapter. The underlining theory of this case study was introduced in Chapter 3, and part of the SCD generated solution procedure is given in §5.5. The full symbolic form of the solution procedure is provided on the right-hand side of the panel, where the SEEM term-tracking information can be shown using the *Tooltip* function.



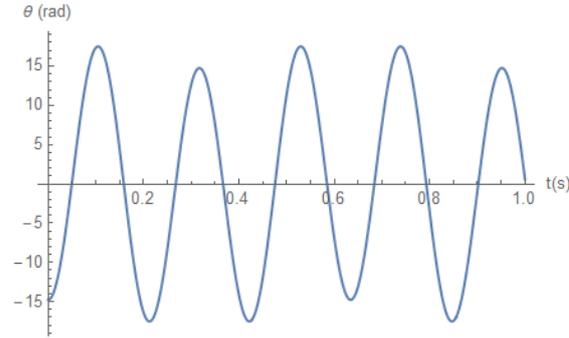


Figure 5-13 The numerical response of the case study is plotted based on the symbolic general solution.

As well as the 2D response plot, an animation of the problem is also included in the visualisation notebook (Figure 5-14). In this format by changing the system characteristics, both the response and animation would be updated (Figure 5-14-d). For example considering Figure 5-14-a and Figure 5-14-b, where the support displacement as fraction of the pendulum length ( $q_{ex}$ ) is decreased from 0.61 to 0.3 respectively. In another example, the length of the pendulum is doubled from Figure 5-14-b to Figure 5-14-c.

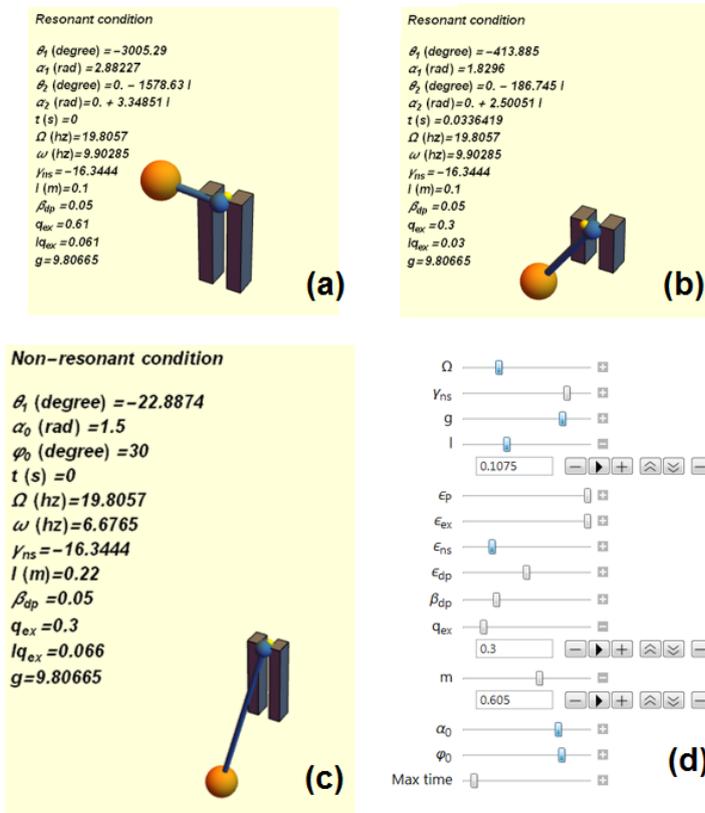


Figure 5-14 An interactive animation of the response based on the symbolic solution.

### 5.6 A case study

The application of the visualisation panel is discussed by investigating the dynamic problem of a parametrically excited pendulum. The same case study as in chapter 3 is used to enable the reader to compare the different features of the SCD solver.

The equation of motion (1) is introduced by the user and encoded (Figure 5-15). The labelled type of displaying is used so that the encoding information can be visible in the report format. The user can decide to add indices to each quantity; for example the quantities in the nonlinear term have *ns* in the index, *ex* is for external excitation, and *dp* is used for the damping term. These indices help the user to distinguish the different sources of the small parameters in each term. The dependent variable,  $\theta$ , does not have any encoding vector because the perturbation expansion series has not yet been introduced, §3.3.1.

$$\begin{aligned}
 - \underset{(1,1)}{q} \cos \left[ \underset{(1,\theta)}{\tau} \underset{(1,\theta)}{\Omega} \right] \underset{(1,1)}{\epsilon_{ex}} + \underset{(1,\theta)}{\omega_1^2} X[\tau] - \underset{(1,1)}{\gamma} \underset{(1,1)}{\epsilon_{nsf}} X'[\tau]^2 - \\
 \underset{(1,1)}{\mu} \underset{(1,1)}{\epsilon_{cplf}} Y'[\tau]^2 + \underset{(1,1)}{2_{dmpf}} \underset{(1,1)}{\epsilon_{dmpf}} \underset{(1,1)}{\xi_1} \underset{(1,1)}{\omega_1} X'[\tau] + X''[\tau] - \underset{(1,1)}{\mu} \underset{(1,1)}{\epsilon_{cplf}} Y[\tau] Y''[\tau] = 0
 \end{aligned}$$

Figure 5-15 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem.

The perturbation expansion up to the first order perturbation equation is introduced, equation (2). Figure 5-16 shows the encoding information for this equation and it can be seen that the perturbation parameter ( $\epsilon_p$ ) is introduced for the first time in this problem. The index *p* is used as an abbreviation for perturbation expansion. Any reference to this equation in the later analysis highlights the choice of the solution procedure expansion, and is not defined through physical modelling.

$$\theta = \underset{(2,0)}{\theta_0}[\mathbf{T}_0, \mathbf{T}_1] + \underset{(2,1)}{\epsilon_p} \underset{(2,1)}{\theta_1}[\mathbf{T}_0, \mathbf{T}_1]$$

Figure 5-16 The automatically generated Source and Evolution Encoding Method information for equation (2) of the parametrically excited pendulum problem.

The first time derivative function is created based on the perturbation expansion, and is stated in D-operator notation.

$$\mathbf{dt}[1] = \underset{(3,0)}{D_0} + \underset{(3,1)}{D_1} \underset{(3,1)}{\epsilon_p}$$

Figure 5-17 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem.

It is convenient to assume that the epsilon shown in equation (4) is numerically equal to the epsilon in the perturbation equation. However, to highlight the source of the epsilons, this instance of the small parameter is shown by a unique symbol;  $\epsilon_{dr}$ .

$$dt[2] = \begin{matrix} D_0^2 & + & D_0 D_1 & 2_{dr} & \epsilon_{dr} \\ (4, 0) & & (4, 1) & (4, 1) & (4, 1) \end{matrix}$$

Figure 5-18 The automatically generated Source and Evolution Encoding Method information for equation (1) of the parametrically excited pendulum problem.

Then the explicit form of  $\theta^3$  is derived, and the third digit of the SEEM encoding is added accordingly, as shown in Figure 5-19.

$$\theta[\tau]^3 = \theta_0[T_0, T_1]^3 \quad (2, 0, 5)$$

Figure 5-19 The automatically generated Source and Evolution Encoding Method information for equation (5) of the parametrically excited pendulum problem.

The first four equations that are shown in Figure 5-20 are generally considered to be fundamental equations; therefore, there are no connection links visible in the graph.

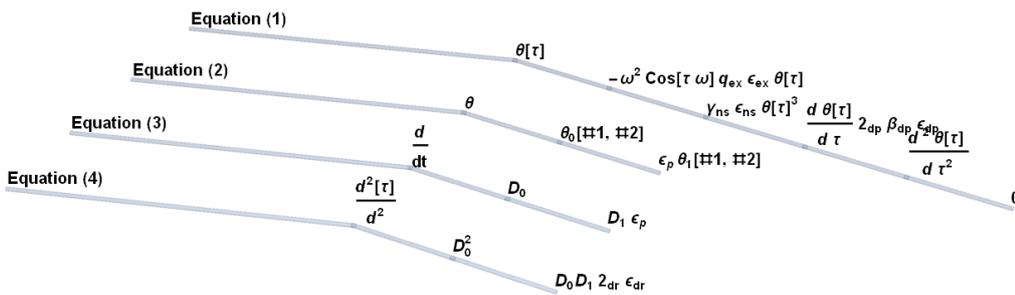


Figure 5-20 The *Blueprint* visualisation for the fundamental equations in the analysis of the parametrically excited pendulum problem.

Equation 5 is the first stage in this analysis, where a new equation based on the fundamental equations is created. Consequently, from this point onwards the sources and links are highlighted using the *Blueprint* method. For consistency, a unique colour for each level of the SEEM is used, considering the first SEEM encoding level; zeroth-order and first order quantities

are shown in pink and blue, respectively. The second level in green, the third in purple, and the compound level in yellow. To emphasize the strength of the zeroth-order terms, the zeroth-order links are shown as denser than those for the first order terms. Note that it is possible in the implementation to change these definitions at any point of the analysis. Considering Figure 5-21, the pink link shows that the zeroth-order  $\theta_0$  coming from the perturbation expansion has been cubed in this equation.

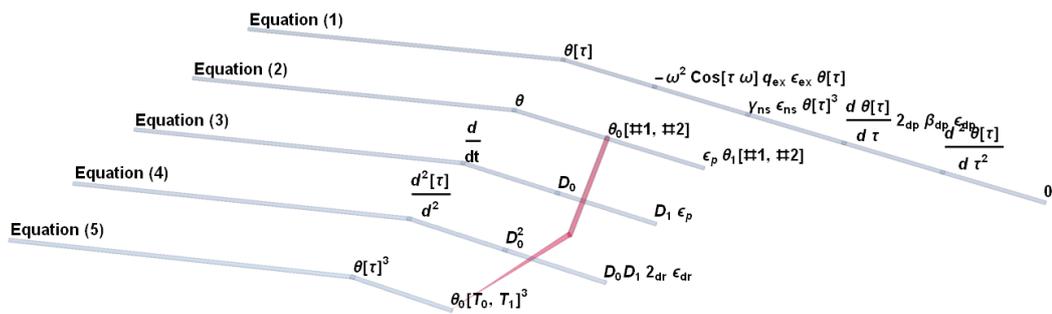


Figure 5-21 The visualisation graph up to as far as equation (5) for the analysis of the parametrically excited pendulum problem.

It has already been explained that the format of the information generated using the SEEM term-tracking method is sensitive to the solution method. Therefore the visualisation graph generated with the *Blueprint* method is specific to the user choice of solution. Some prefer a very detailed analysis, where a single equation is defined for each minor step of the analysis, while others will prefer a concise approach where only the main steps are provided. Obviously the SEEM information generated for a detailed analysis has a higher resolution comparing to that for a concise version. For example, it is possible to skip equation (5) and substitute equation (2) into equation (1) in a single step, which is mathematically correct but results in very different encoding information, and therefore a different visualisation pattern. The benefit of defining a separate equation for the explicit form of  $\theta^3$ , is that the user can distinguish the exact step at which the nonlinear restoring force is defined. The general inference from this is that the encoding strategy should tend to favour maximised detail at the outset as this can always then be suppressed if it is not wanted.

Equations (2), (3), (4) and (5) are substituted into equation (1), and the result is expanded, with terms of higher than first order being discarded, according to the decision already made about the accuracy that is needed from the perturbation expansion. Figure 5-22 shows the overall sources and links for equation (6). The symbolic forms of the terms in this equation are showed on the equation line, with no particular order. The source of each quantity is shown upstream of the equation line. It is possible for the user to hover the mouse pointer around any link to

display that quantity and its encoding information, using the *Tooltip* function. The choice of the colour and thickness of the links can clearly highlight the powerful zeroth-order quantities from the first order ones. Moreover it is possible to identify the path for the  $\theta_0$ , noting that a pink link connects this quantity from the perturbation equation into equation (5), highlighted in orange. Afterwards, the colour of the link changes to green from equation (5) to (6), highlighted in yellow. This highlights the change of the first SEEM encoding level, to second; as the explicit form of the  $\theta_0^3$  was determined in equation (5), Figure 5-19.

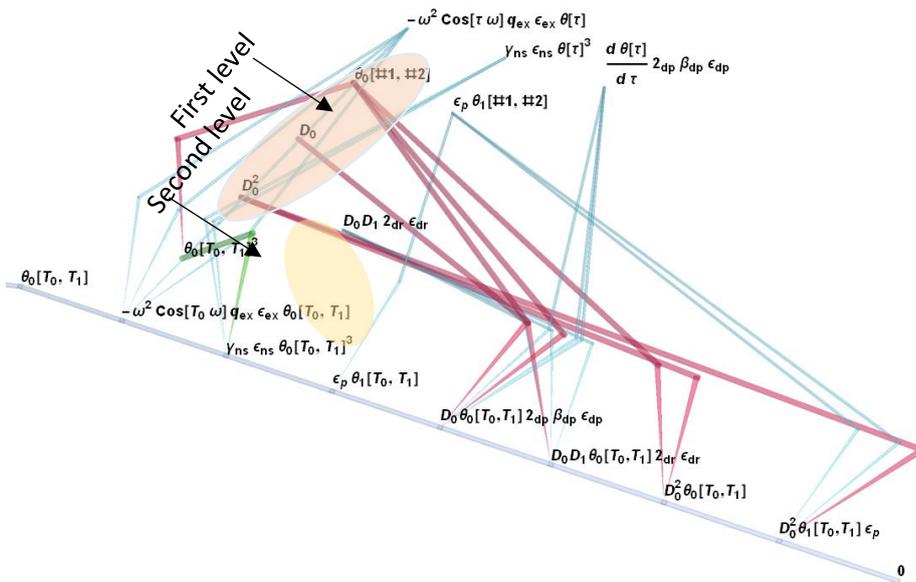


Figure 5-22 The overall *Blueprint* visualisation graph for equation (6). The first level of the encodings for  $\theta_0^3$  is highlighted in orange, and the second level in yellow.

The *Blueprint* visualisation method can show an overall view for each equation in the solution procedure, as well as detailed information for each term. As Figure 5-23 shows the user can zoom in on each term to investigate the structure of each term; by pressing ctrl and hovering the mouse pointer in upward or downward directions. In this Figure the SEEM encoding information are shown with the *Labeled* function. Figure 5-23-a shows the nonlinear stiffness term, which it is introduced based on a modelling assumption. The explicit form of  $\theta_0^3$  is substituted from equation (5), the link is shown in green. The order of  $\theta_0^3$  is zero, so the link thickness is larger than the first order links. Furthermore,  $\epsilon_{ns}$  is introduced in the equation of motion to scale the nonlinear restoring force. Figure 5-23-b is the external excitation term, the term  $\epsilon_{ex}$  is introduced through the assumption that the support displacement is much smaller than the rod length ( $q_{ex}$ ). This is based on a physical interpretation at the modelling phase. The external excitation frequency,  $\omega$ , and  $\theta_0$  are both zeroth-order quantities. Moreover, Figure

5-23-c presents the first order perturbational correction term,  $\epsilon_p$  is introduced based on the perturbation expansion. All the quantities in this term are first order and therefore links are in blue. The damping term is shown in Figure 5-23-d,  $\epsilon_{dp}$  is introduced by physical modelling to scale the damping term. Also,  $\theta_0$  is sourced from the perturbation equation and the first time derivative is linked to equation (3). Furthermore, Figure 5-23-e is a perturbational correction term and considering the first digit of the encoding, there is no quantity sourced from the equation of motion. This term structure is highly depending on the choice of the solution method. Finally, Figure 5-23-f is the powerful inertia term, all quantities are zeroth-order.

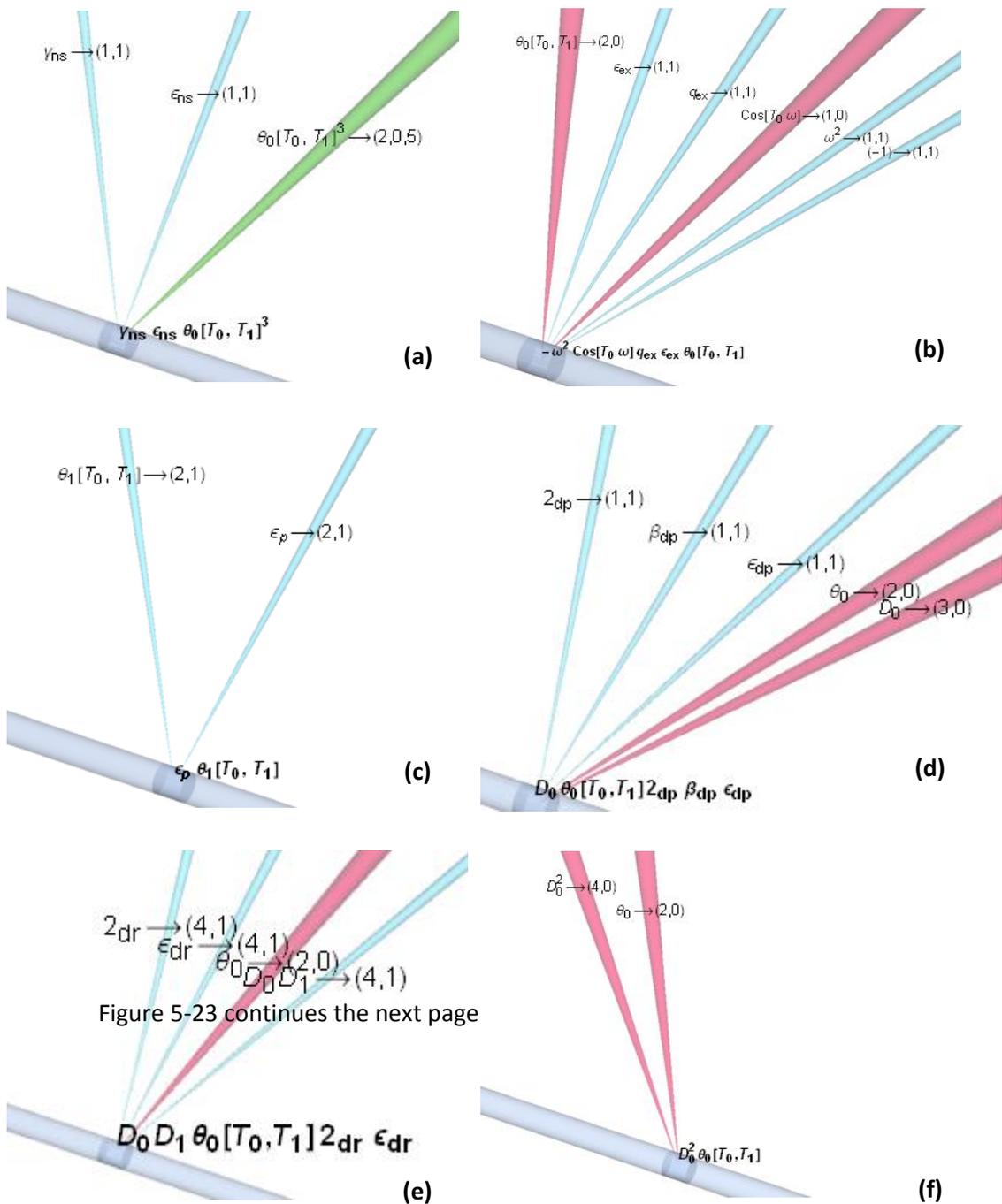


Figure 5-23 continues the next page

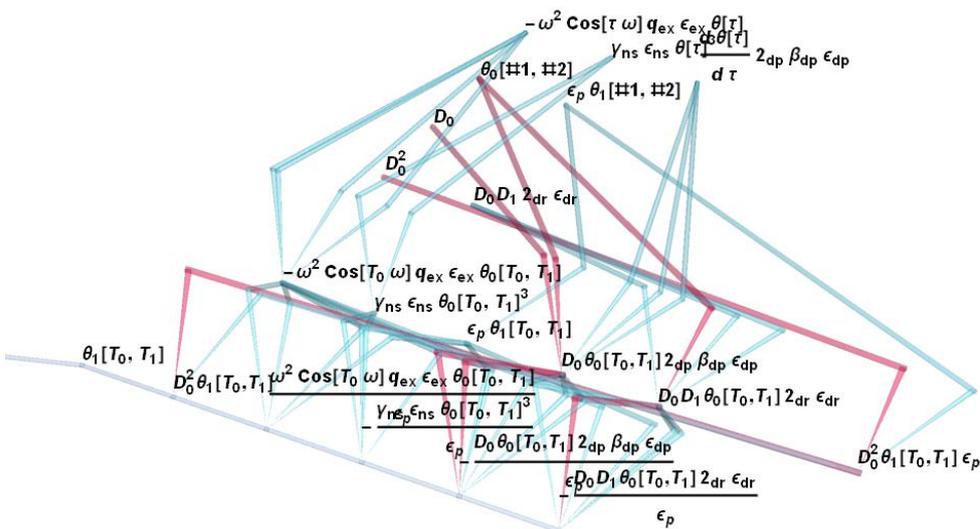
Figure 5-23 Detailed discussion for selected terms in equation (6), where; (a) is nonlinear stiffness term, (b) is the external excitation term, (c) is a perturbational correction term, (d) damping term, (e) is a perturbational correction term, and (f) is the inertia term.

Figure 5-24 presents the zeroth-order perturbation equation, which is obtained by setting the coefficient of  $\epsilon^0$  in equation (6) to zero. As this is a single degree of freedom problem, with just one generalised coordinate, then  $\theta_0$  is in fact the main part of the motion that the chosen generalised coordinate undertakes. Physically this equation represents the swinging motion of the pendulum in this case study. The D-operator notation is used to represent the time derivative with respect to the time scales  $T_i$ . Physically the second order time derivative relates directly to the inertia of the problem, which in turn is based purely on the kinetic energy within the pendulum.

$$\begin{array}{ccc}
 D_0^2 & \theta_0 [T_0, T_1] + \theta_0 [T_0, T_1] & = 0 \\
 (4, 0) & (2, 0) & (2, 0)
 \end{array}$$

Figure 5-24 Encoding information for equation (7).

The first order perturbation equation is derived by taking the coefficient of  $\epsilon^1$  terms out from equation (6) out and set to zero, resulting equation (8). Both the SEEM encoding information and the overall *Blueprint* visualisation are given in Figure 5-25.

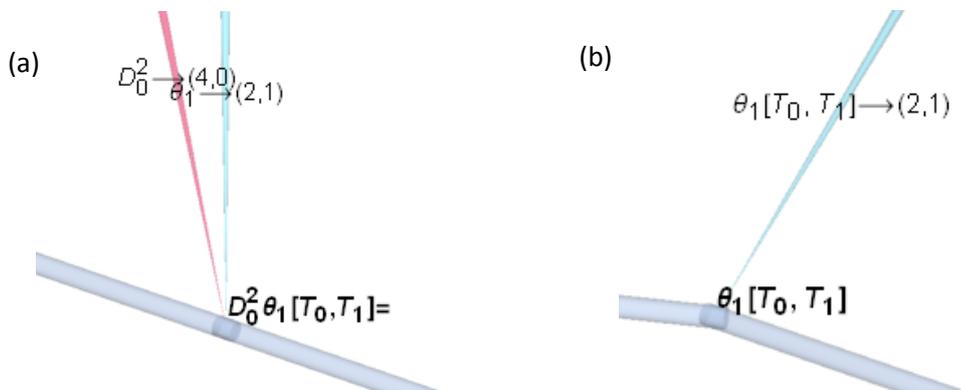


$$\begin{array}{c}
 D_0^2 \theta_1 [T_0, T_1] + \theta_1 [T_0, T_1] = | \\
 (4, \theta) \quad (2, 1) \\
 (2, 1) \\
 \hline
 \begin{array}{ccccccc}
 D_0 & 2_{dp} & \beta_{dp} & \epsilon_{dp} & \theta_0 [T_0, T_1] & D_0 D_1 & 2_{dr} & \epsilon_{dr} & \theta_0 [T_0, T_1] \\
 (3, \theta) & (1, 1) & (1, 1) & (1, 1) & (2, \theta) & (4, 1) & (4, 1) & (4, 1) & (2, \theta)
 \end{array} \\
 \hline
 \begin{array}{ccc}
 \epsilon_p & & \epsilon_p \\
 (2, 1) & & (2, 1)
 \end{array} \\
 \hline
 \begin{array}{ccccccc}
 \gamma_{ns} & \epsilon_{ns} & \theta_0 [T_0, T_1]^3 & \omega^2 & \cos [T_0 \omega] & q_{ex} & \epsilon_{ex} & \theta_0 [T_0, T_1] \\
 (1, 1) & (1, 1) & (2, \theta, 5) & (1, 1) & (1, \theta) & (1, 1) & (1, 1) & (2, \theta)
 \end{array} \\
 \hline
 \begin{array}{ccc}
 \epsilon_p & & \epsilon_p \\
 (2, 1) & & (2, 1)
 \end{array}
 \end{array}$$

Figure 5-25 Overall visualisation graph for equation (8).

Figure 5-26 shows the detailed investigation of the selected terms in equation (8). Figure 5-26-a is the inertia term, in this term the differentiation operator source is equation (4), whilst the source of  $\theta_1$  is the perturbation equation. Furthermore, Figure 5-26-b shows the first order perturbation term,  $\theta_1$ . In the standard multiple scale analysis the small epsilon parameters such as  $\epsilon_p$  and  $\epsilon_{ex}$  (in Figure 5-26-c) would cancel each other out. However, the epsilons are reserved here in order to plot their history,  $\epsilon_{ex}$  is introduced to scale the external excitation, and  $\epsilon_p$  is the perturbation small parameter.

Figure 5-26-d shows the damping term in a conventional multiple scales analysis. The nonlinear restoring force is shown in Figure 5-26-e,  $\epsilon_p$  is the perturbation parameter and the  $\epsilon_{ns}$  is introduced in the equation of motion. Finally, in Figure 5-26-f both  $\epsilon_s$  are sourced from the perturbation equation, and it might be possible to cancel these two epsilons without losing any vital information. The zeroth-order perturbation term is the only strong quantity in this term, shown in pink.



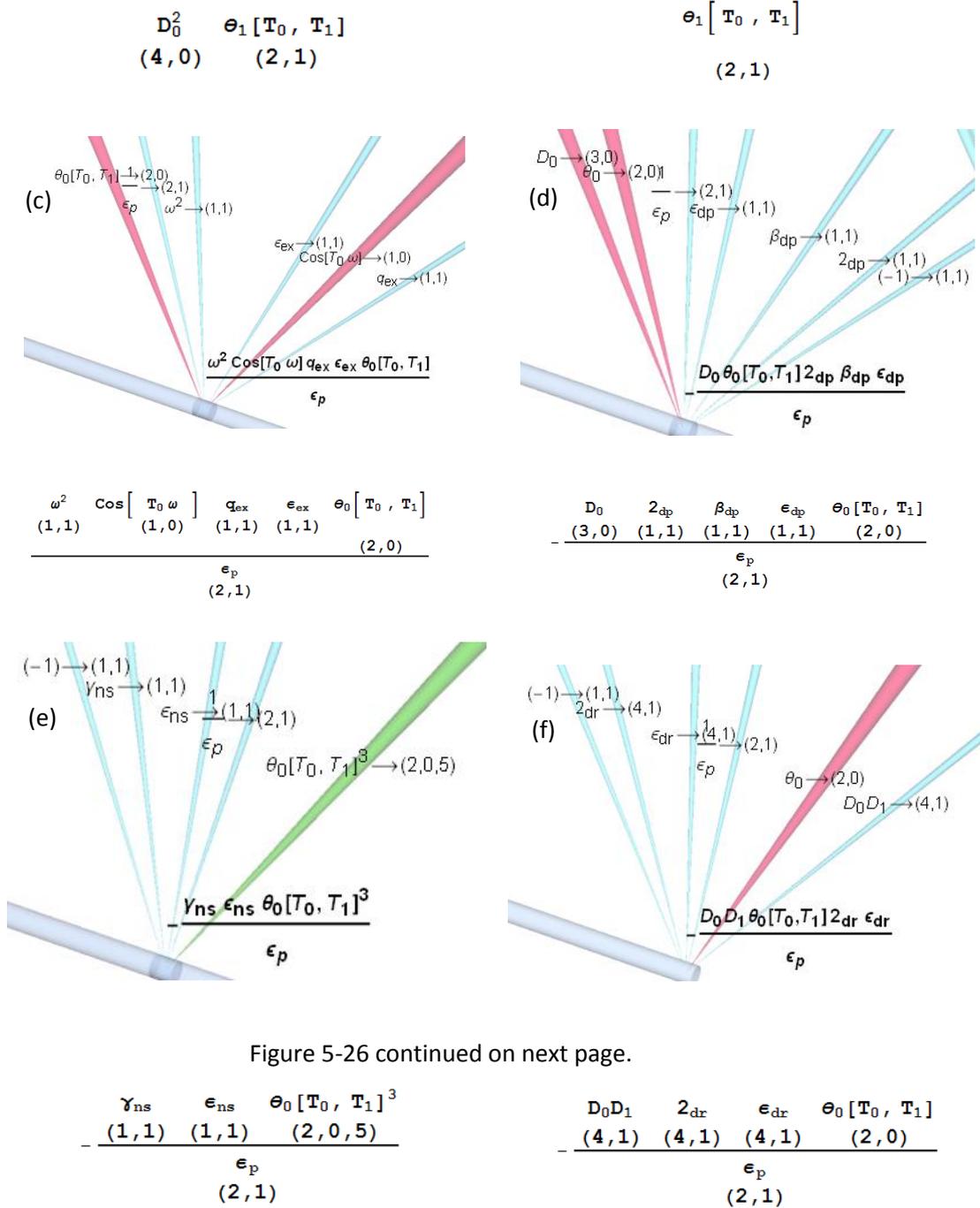


Figure 5-26 continued on next page.

Figure 5-26 The detailed encoding information for selected terms in equation (8); where, (a) is the inertia term, (b) is the first order perturbation term, (c) is the external excitation term, (d) is the damping term, (e) is the nonlinear stiffness term, and (f) is a perturbational correction term.

It is possible to hide other equations and only visualise the solution of the zeroth-order perturbation equation is shown in Figure 5-27. This figure shows that all the links are zeroth-order (pink) and there are no other links related to other SEEM encoding levels. The arbitrary



$$\theta_0 [T_0, T_1]^3 = 3 \times \frac{A[T_1]^2 \bar{A}[T_1]}{(2,0,9,10)} e^{i T_0 (9,0,10)} + \frac{3 \times A[T_1] \bar{A}[T_1]^2}{(10,0) (2,0,9,10) (9,0,10)} e^{i T_0 (9,0,10)} + \frac{\bar{A}[T_1]^3}{(2,0,9,10) (9,0,10)} e^{-3 T_0 (1,0,10) (9,0,10)} + \frac{A[T_1]^3}{(2,0,9,10) (9,0,10)} e^{i T_0 (9,0,10) (2,0,9,10)}$$

Figure 5-28 Overall visualisation graph for equation (10).

Figure 5-29 shows the thorough detail of the encodings for equation (10). Figure 5-29-a shows the explicit nonlinear restoring term; where the encoding link is given in green. The user can quickly identify the history of this term, such as the source of  $\theta_0$  is from equation (2) which is the perturbation expansion, and its order of epsilon is zero. It becomes explicit in equation (5) where the perturbation expansion has been used to get the cubic form of  $\theta_0$  into first order epsilon form.

Furthermore, the solution of the zeroth-order perturbation has been involved in the term shown in Figure 5-29-b. There is no first order quantity in this term and the visualisation links and SEEM encoding vector can clearly show the history of each quantity. In summary  $\bar{A}$  is referenced to the perturbation expansion which is shown in explicit format in equation (9). Furthermore,  $\theta_0$  is introduced in the form of the amplitude  $A$  in equation (9) for the first time. Also  $e^i$  is introduced in the solution of the zeroth-order perturbation equation. Furthermore, 3 relates to equation (1), from which it can be deduced that the nondimensional excitation frequency and then multiplied by 3 in equation (10). As the complex amplitude is powered to 3 in this term, modified from the original form, a fourth digit of encoding is added to signify that fact.

Figure 5-29-c shows the interactions of the amplitude  $A$  and its conjugate, and as the results of this the third level of the SEEM method is applied to these quantities.

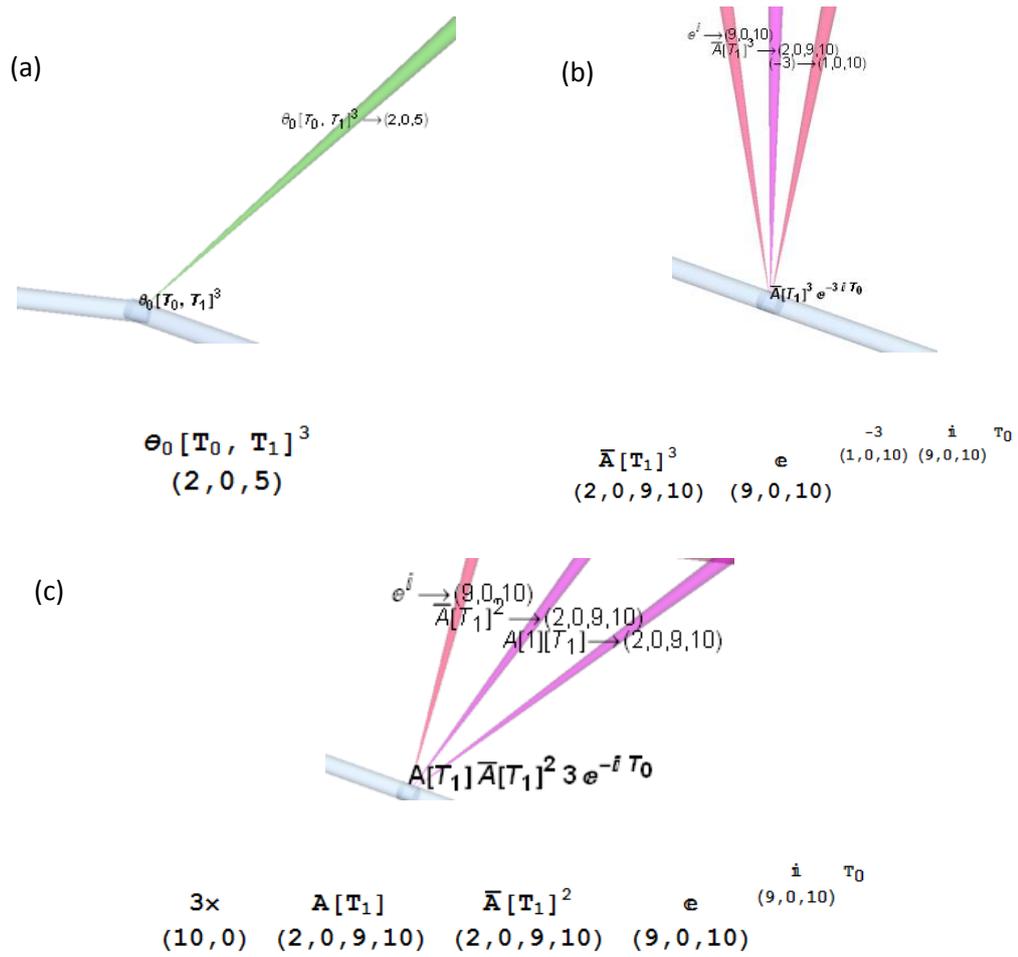


Figure 5-29 A selected detailed discussion of the SEEM information for equation (10); where, (a) is the nonlinear restoring force, (b) and (c) are the definition of the restoring force based on the zeroth-order perturbation solution.

The exponential form of the cosine function (defined as Cos here) is shown in Figure 5-30. This step is added for the convenience of the user and is not necessarily a fundamental step of the multiple scales analysis (eqn (11)). Considering the *Blueprint* visualisation graph in this figure, the function has correctly linked this term to the external excitation term in the equation of motion.

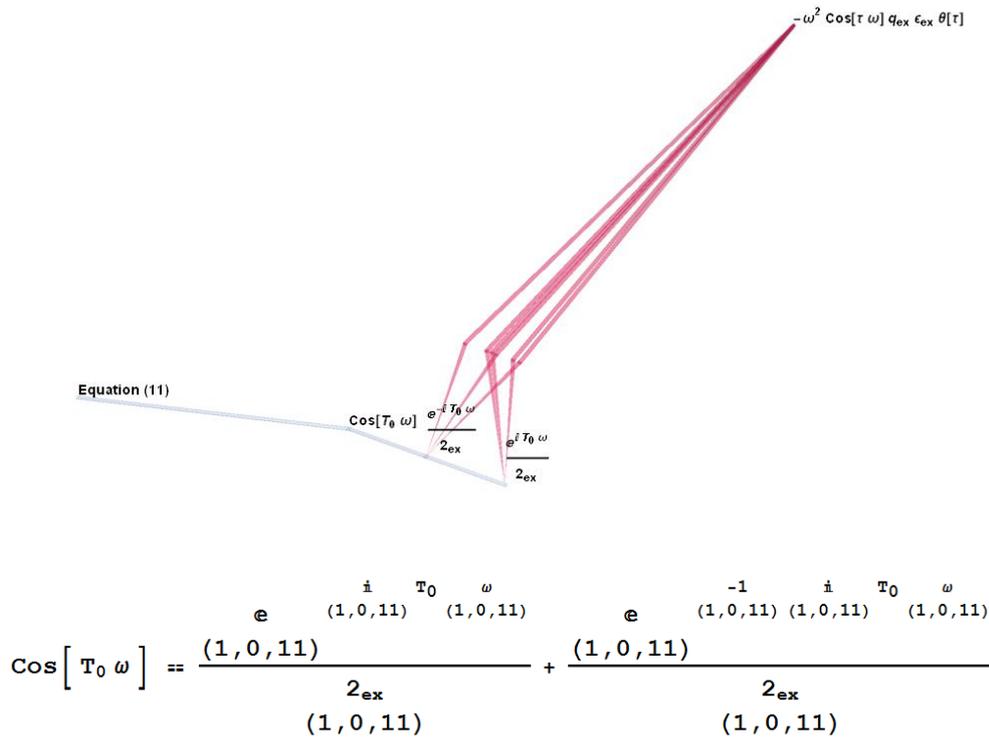


Figure 5-30 Encoding information for equation (11).

### 5.7 Summary

In this chapter the need for a robust visualisation method for the SCD solvers is discussed. The main criteria for possible visualisation methods are introduced, and two approaches have been suggested; the *ColourMap* and the *Blueprint*. The *ColourMap* method is fully introduced and computerised. However, due to its lack of practicality at anything but the smallest scale modelling, any future development of this method has been terminated.

The *Blueprint* method concept is fully introduced, and the obstacles in computerising this method have been investigated. This idea has been developed in three stages; an automated panel, a 2D visualisation panel, and the 3D visualisation panel. The final version of the 3D *Blueprint* visualisation panel is applied to the problem of a parametrically excited pendulum, and the results for the first 11 equations of this analysis are provided in this chapter. The main conclusions from this case study are listed as follows:

- Generally visualisation links makes the interpretation of the complex mathematical solution procedure easier for the user.

- The visualisation relates the complex mathematical notations of the solution to the practical physical reasoning very early on in the modelling and solution phases.
- Colour coding and modifying the width of the links clarifies the significance of the relevant quantity to the analysis quickly.
- For more information, the user can zoom in on each node and check the encoding information in the finest grained level of detail.

It can be noted that this method has a huge potential for the future developments. It is possible to introduce the controlling functions which enable the user to visualise information layer by layer. These functions can assist the user to validate the solution procedure just by investigating the generated visualisation results, a thorough discussion on the future plan for the *blueprint* method is provided in §7.2 of this thesis.



# Chapter 6 An Evaluation Method for Symbolic Terms

## 6.1 Introduction

The development of the first version of the *Blueprint* visualisation method for Symbolic Computational Dynamics (SCD) solvers was discussed in the previous chapter. The *Blueprint* visualisation method enables the user to obtain an overview of the complete analysis as well as generating a detailed investigation of the history of each term. The detailed investigation is achieved by zooming in on each term in the equation, and then analysing the encoding links to create a solid connection between the physics of the system and the mathematical solution procedure. Interesting observations can be made in these types of analysis, starting off by considering Figure 6-1 one can see the variability of the distribution of the SEEM links in terms of colours and thicknesses. The first term has two zeroth-order and four first order quantities, and the second term has one zeroth-order and five first order quantities (the zeroth-order term are shown in pink and the first order term are shown in blue). The key question here is: “is a term with a larger number of zeroth-order quantities, regarded as having a higher significance when compared to the other terms in the same equation?”.

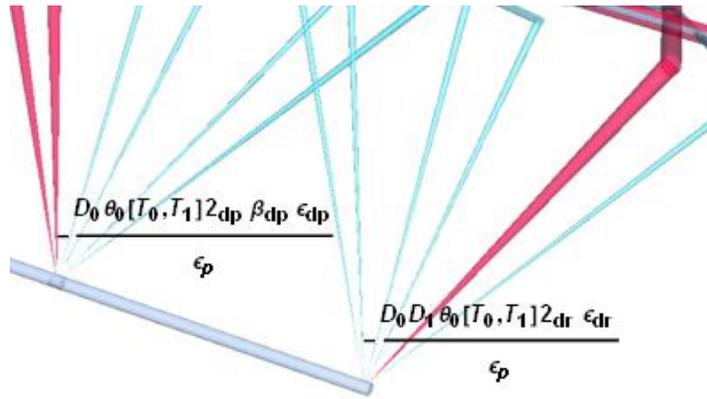


Figure 6-1 A graphical example of two terms with different *Blueprint* encoding structures.

In order to be able to derive a mathematically and physically correct solution for the final response when using an adaptable perturbation method such as multiple scales, there is a necessity for the user to be able to apply both formal and informal interventions to the method, to ensure the uniformity of the perturbation expansion and also to eliminate any spurious dynamic effects that may otherwise arise. In the formal case one considers key issues such as secularity, autonomy, and slow modulations of the amplitude and phase. However other interventions, which are somewhat informal, such as the identification of terms that are principally relevant (or not) to a resonance condition of interest, usually require some degree of user experience, and there are not necessarily general guidelines for doing this. Therefore, there could be merit in developing a term evaluation concept which is based on the symbolic formulations, and which applies ideas of the SEEM hierarchy to aid such processes. Ideally this could be a reference method for the user to compare the significance of terms in a single equation, and even explore how they may contribute to the solution based on a chosen resonance condition. Developing the SF theory can ultimately increase the popularity of the perturbation method as they can assist the analyst to specify possible negligible terms.

Figure 6-2 summarises the development stages of the SCD solver up to this chapter of the thesis and may be interpreted as follows. Chapter 3 introduces the concept of the SEEM term-tracking method, then chapter 4 summarises the algorithms developed for implementation of the SEEM into the SCD solver. The first version of the visualisation method for the SCD solver is introduced in chapter 5, and this current chapter introduces a possible term evaluation method for the SCD solvers.

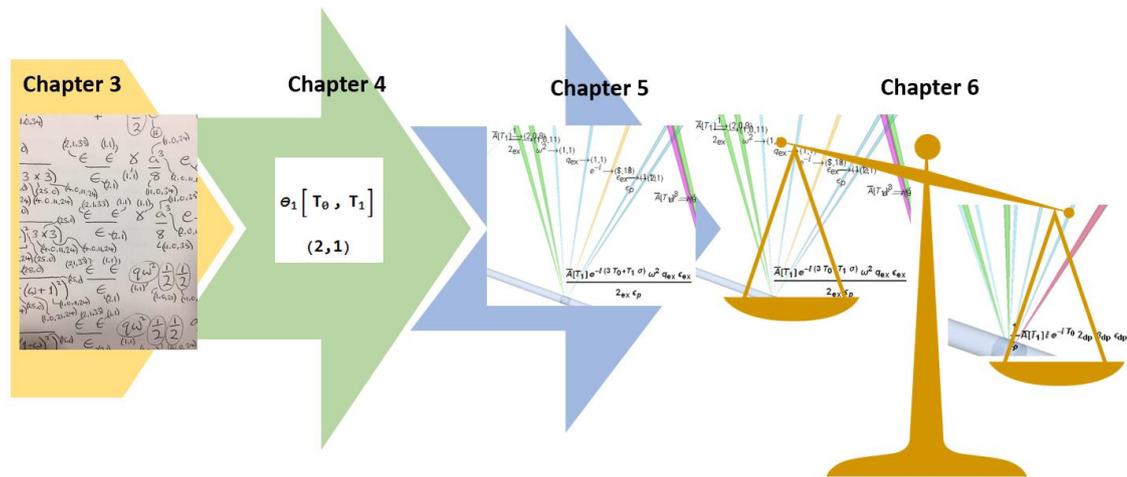


Figure 6-2 The development procedure of the Symbolic Computational Dynamics solver in this thesis.

## 6.2 The Strength factor

A new metric based on the SEEM information is proposed in order to highlight the significance of each term in an equation. This metric is in the form of a calculated number and this is henceforth called the 'Strength Factor' (SF). The SF is based on both the order and the SEEM hierarchy of the constituent quantities in each term. The first level of the SEEM represents the order of quantity. The SEEM hierarchy, Table 3-1, for each quantity is categorised into two classes: *simple* or *complex*. The quantities with the first SEEM encoding level are classified as *simple*, and the second and higher SEEM encoding levels are placed in the *complex* classification.

A quantity that classified as *complex* has possibly more involvement in a particular analysis, compared with a quantity with the *simple* classification. For example, the contribution of a quantity that has been part of the equation structure of the explicit definition of the amplitude response contribution more than a quantity that has the first level of the SEEM.

In order to quantify this concept a points system based on order and SEEM hierarchy has been defined. Table 6-1 is a guideline for both the order-associated points and the SEEM hierarchy that are used in this research. Based on experience, it is assumed the effectiveness of a zeroth-order term is twice that of the first order, and quantities with *complex* SEEM hierarchy have twice the points value compared to quantities with the *simple* classification. It can be noted that these conditions can be adjusted according to user preference.

Table 6-1 A points system in calculation of the Strength Factor.

<b>Order point</b>	<b>Zeroth-order</b>	<b>First order</b>
	2	1
<b>SEEM hierarchy point</b>	<b>Simple</b>	<b>Complex</b>
	2	4

Equation (6.1) shows the proposed SF formulation for each quantity in a term; noting that for each quantity the order point is multiplied by the SEEM hierarchy point, and then normalised. The maximum value that a  $SF_{Quantity}$  can have is eight and this defines a zeroth-order term with a *complex* classification. The SF formulation for a term is given in (6.2), the  $SF_{Quantity}$  values in a term are added and divided by the number of quantities in a term.

$$SF_{Quantity} = \frac{Order\ point * SEEM\ hierarchy\ point}{8} \tag{6.1}$$

$$SF = \frac{\sum SF_{Quantity}}{\sum quantities} \tag{6.2}$$

Table 6-2 shows the process of calculating the SF value for  $\underbrace{A}_{(2,0,9)} \underbrace{e}_{(9,0)} \underbrace{\hat{i}}_{(9,0)}^\tau$ , this term is taken from equation (9) in chapter 3. For each quantity in this term both the order and SEEM hierarchy points are defined, then the  $SF_{Quantity}$  is determined. Finally, all three  $SF_{Quantity}$  values are added together and divided by three for normalisation. It should be noted from this point forward all the reference to SF is regarded to SF value calculated for terms.

Table 6-2 Calculating the Strength Factor value for a term.

<b>Quantity</b>	<b>Order point</b>	<b>SEEM hierarchy</b>	<b><math>SF_{Quantity}</math></b>	<b>SF</b>
$\underbrace{A}_{(2,0,9)}$	Zeroth → 2	<i>Complex</i> → 4	$(2 \times 4)/8$	$\frac{2}{3}$
$\underbrace{e}_{(9,0)}$	Zeroth → 2	<i>Simple</i> → 2	$(2 \times 2)/8$	
$\underbrace{\hat{i}}_{(9,0)}$	Zeroth → 2	<i>Simple</i> → 2	$(2 \times 2)/8$	

### 6.3 Implementation of the Significance Factor

It is essential to implement the SF number within the *Blueprint* visualisation method. The SF value is shown graphically as a sphere around each term node in the visualisation pallet, and the

diameter of the sphere is equal to the SF value of the term. As Figure 6-3 illustrates this allows the user to grasp quickly the significance of each term by merely inspecting the visualisation graph.

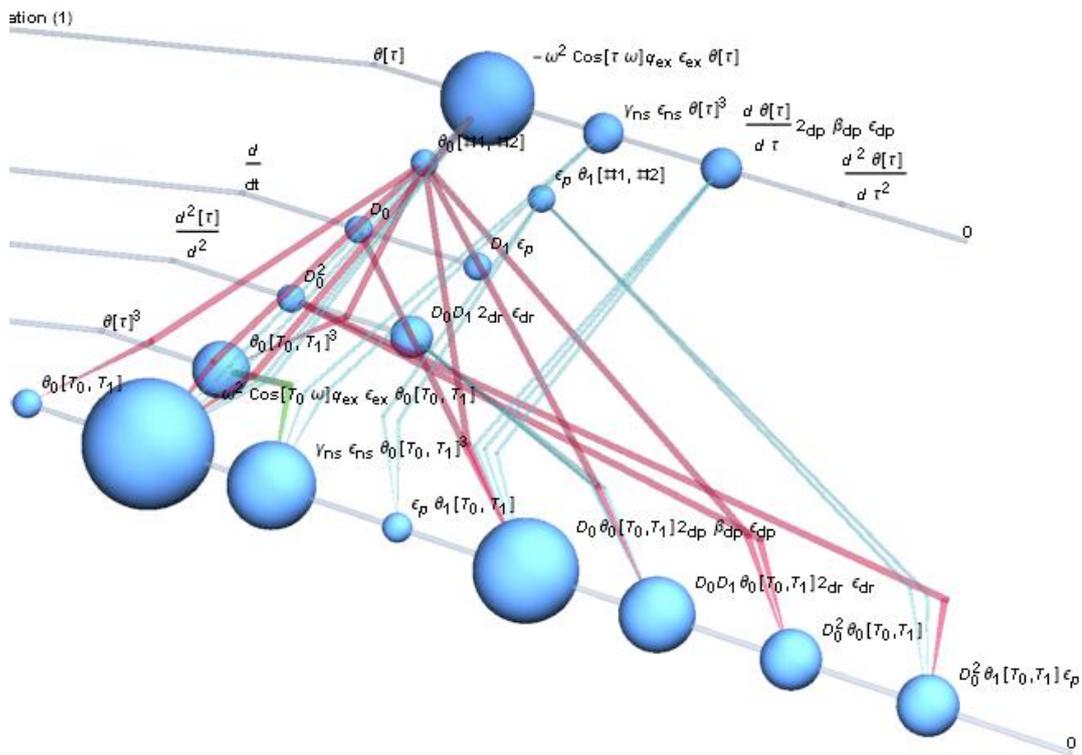


Figure 6-3 A graphical example of implementation of the Strength Factor number in the *Blueprint* visualisation graph.

### 6.4 An autoparametric case study

An autoparametric system is now used as a basis for further testing of the SCD solver. Figure 6-4 shows the well-known problem of a pair of beams coupled in such a manner that the forced response of the horizontal primary beam acts as a parametric excitation of the vertical secondary beam [68]. A single frequency external excitation is applied to the primary beam (AB) at the first natural frequency of free undamped vibration, resulting in a first bending mode response. The secondary beam (BC) is considered to have a high flexural stiffness ratio; therefore, the bending deformation of BC in the ABC plane is negligible, and so the response of the secondary beam is out of the plane of the diagram. This depends on the system being configured in such a way that the response of the fundamental primary mode is around twice the frequency of the fundamental secondary mode, in which case the condition for autoparametric resonance is satisfied.

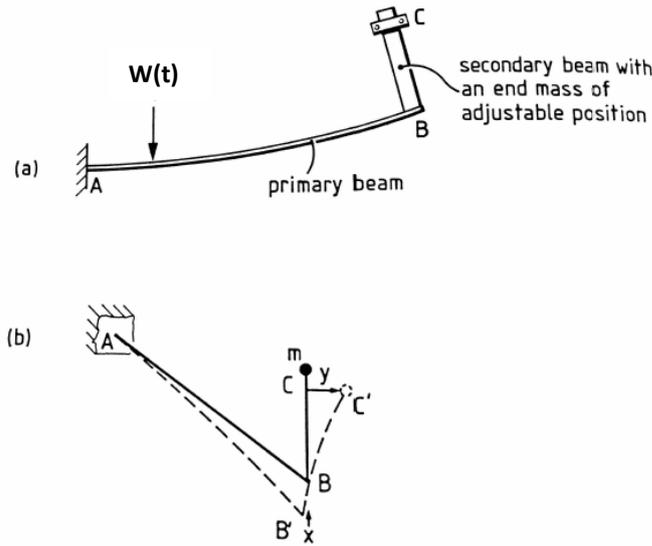


Figure 6-4 Coupled beam system based on Cartmell [70]. AB is the primary beam, and BC is the secondary beam.  $W(t)$  represents the external excitation.

#### 6.4.1 Physical modelling

The autoparametric system shown in this sort of systems can be modelled as follows [68]:

$$X\ddot{[\tau]} + 2\xi_1\omega_1\dot{X}[\tau] + \omega_1^2X[\tau] - \varepsilon\mu(\dot{Y}[\tau]^2 + \dot{Y}[\tau]Y[\tau]) - \varepsilon\gamma\dot{X}^2[\tau] = Q_0\cos[\Omega\tau] \quad (6.3)$$

$$\dot{Y}[\tau] + 2\xi_2\omega_2\dot{Y}[\tau] + \omega_2^2Y[\tau] - \varepsilon Y[\tau]\ddot{X}[\tau] = 0 \quad (6.4)$$

In these equations  $X$  and  $Y$  represent the physical in-plane and out-of-plane modal responses of the primary and secondary beams respectively. The first linear natural frequency for the primary and secondary systems are shown by  $\omega_1$  and  $\omega_2$ . The coupling between primary and secondary beam is modelled by  $\varepsilon\mu(\dot{Y}[\tau]^2 + \dot{Y}[\tau]Y[\tau])$  in equation (6.3). Considering equation (6.4),  $\ddot{X}$  is the parametric coefficient for the coordinate  $Y$ . The damping coefficients for both systems, up to the first order, are presented with  $\xi_1$  and  $\xi_2$  symbols. The damping coefficients for the primary and secondary system can be shown as  $\varepsilon\zeta_1$  and  $\varepsilon\zeta_2$ . Furthermore, the external excitation amplitude is shown by  $Q_0$  which is equal to  $\varepsilon q$ . Unlike the problem of the parametrically excited pendulum in chapter 3 and 5, where it was equal to nondimensional time based on the natural frequency ( $\tau = \omega_0 t$ ), in this case study it is equal to the real time ( $\tau = t$ ). The complete process of derivation of the equations of motion for this system can be found in [72].

#### 6.4.2 Solution procedure

A part of the solution procedure for the problem of the autoparametrically excited coupled beam system based on the developed SCD solver is given in this section. For this case study a

unique label of ‘eqn’ is used for addressing the equation numbers. The in-plane equation of motion for a coupled beam, based on the Roberts and Cartmell [72] formulation, is given in eqn (1) In this equation the external excitation amplitude is shown by  $q$ ,  $\varepsilon_{ex}$  is a small parameter introduced for scaling the external excitation,  $\omega_1$  is the first natural frequency of free undamped vibration,  $\gamma$  is the coefficient of centripetal acceleration,  $\varepsilon_{nsf}$  is a small parameter that scales this particular nonlinear stiffness term,  $\mu$  is the effective mass ratio,  $\varepsilon_{cplf}$  is a small parameter added to scale the quadratic coupling terms, and  $\varepsilon_{dmpf}$  is a small parameter introduced to scale the damping term for the primary system. Finally,  $\zeta_1$  is the damping coefficient for the primary system.

$$\begin{aligned} \ddot{X}[\tau] + 2\varepsilon_{dmpf}\varepsilon_{dmpf}\zeta_1\omega_1\dot{X}[\tau] + \omega_1^2X[\tau] \\ - \varepsilon_{cplf}\mu(\dot{Y}[\tau]^2 + \ddot{Y}[\tau] * Y[\tau]) - \dot{X}[\tau] * \dot{X}[\tau]\gamma\varepsilon_{nsf} \\ = \varepsilon_{ex}q\cos[\Omega\tau] \end{aligned} \quad \text{eqn (1)}$$

The overall visualisation graph for this equation is shown in Figure 6-5. In this case study this equation is regarded as a fundamental equation and is introduced as an input to the SCD solver by the user; therefore, no links are visible in this figure. The SF value for each term is graphically shown using a sphere placed at each term node, with the diameter equal to the SF number. As the perturbation expansion has not yet been introduced  $\ddot{X}[\tau]$  is not encoded, and no SF representation is generated for this term. It can be noted that the  $\varepsilon$  and  $\epsilon$  in this chapter are the same.

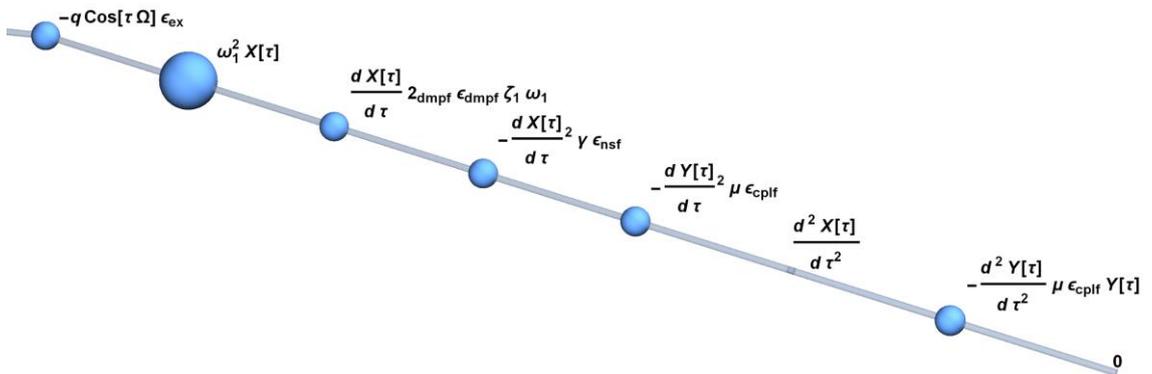


Figure 6-5 The visualisation graph for the primary beam's equation of motion.

The automatically generated SEEM term-tracking information for this equation is shown in Figure 6-6 and it should be noted that the terms of the differential equation have been rearranged from left to right in order to suit the SCD process. The external excitation term,  $\varepsilon_{ex}$  is introduced to scale the excitation amplitude ( $q$ ). The external excitation frequency ( $\Omega$ ) is

clearly independent of  $\varepsilon_{ex}$  and considered a zeroth-order quantity. Also, the coefficient of centripetal acceleration ( $\gamma$ ) is scaled by the small parameter  $\varepsilon_{nsf}$  through the process of modelling. The first time derivative of  $X$  is not encoded because the perturbation expansion has not been yet introduced to the analysis. Furthermore,  $\varepsilon_{cplf}\mu\dot{Y}[\tau]^2$  is a quadratic coupling term for which it is assumed that the effective mass ratio ( $\mu$ ) is scaled by  $\varepsilon_{cplf}$ . Finally,  $2\varepsilon_{dmpf}\varepsilon_{dmpf}\zeta_1\omega_1\dot{X}[\tau]$  is the damping term for the primary beam, and it is assumed that the damping ratio ( $\zeta_1$ ) is small enough to be scaled by introducing  $\varepsilon_{dmpf}$ .

$$\begin{aligned}
 & - \underset{(1,1)}{q} \underset{(1,0)}{\text{Cos}} \left[ \underset{(1,0)}{\tau} \quad \underset{(1,0)}{\Omega} \right] \underset{(1,1)}{\varepsilon_{ex}} + \\
 & \underset{(1,0)}{\omega_1^2} X[\tau] - \\
 & \underset{(1,1)}{\gamma} \underset{(1,1)}{\varepsilon_{nsf}} X'[\tau]^2 - \\
 & \underset{(1,1)}{\mu} \underset{(1,1)}{\varepsilon_{cplf}} Y'[\tau]^2 + \\
 & \underset{(1,1)}{2\varepsilon_{dmpf}} \underset{(1,1)}{\varepsilon_{dmpf}} \underset{(1,1)}{\zeta_1} \underset{(1,1)}{\omega_1} \\
 & X'[\tau] + X''[\tau] - \underset{(1,1)}{\mu} \\
 & \underset{(1,1)}{\varepsilon_{cplf}} Y[\tau] Y''[\tau] = 0
 \end{aligned}$$

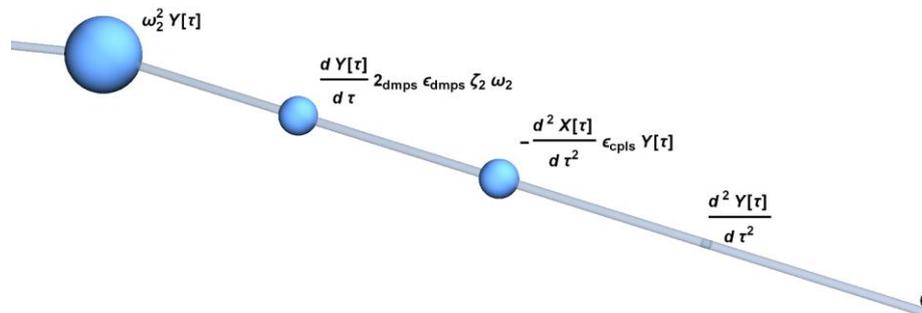
Figure 6-6 A screenshot of the automatically generated encodings for eqn (1).

The equation of motion for the secondary beam is taken from Roberts and Cartmell [72] and shown in eqn (2). where,  $Y$  represents the out-of-plane motion of the secondary beam,  $\omega_2$  is natural frequency of free undamped vibration for the secondary beam,  $\varepsilon_{dmps}$  scales the damping term,  $\zeta_2$  is the damping coefficient for the secondary system, finally  $\varepsilon_{cpls}$  scales the coupling term.

$$\omega_2^2 Y[\tau] + 2_{dmps} \epsilon_{dmps} \zeta_2 \omega_2 \dot{Y}[\tau] - \epsilon_{cpls} Y[\tau] \ddot{X}[\tau] + \dot{Y}[\tau] = 0 \quad \text{eqn (2)}$$

Figure 6-7 shows the automatically generated SEEM information, where;  $\omega_2^2 Y[\tau]$  is the linear stiffness term for the secondary beam, and  $2_{dmps} \epsilon_{dmps} \zeta_2 \omega_2 \dot{Y}[\tau]$  is the damping term which is considered to be weak due to the introduction of  $\epsilon_{dmps}$ . The coupling term,  $\epsilon_{cpls} Y[\tau] \ddot{X}[\tau]$ , is scaled by introducing  $\epsilon_{cpls}$ .

It could easily be argued that the small coupling parameters ( $\epsilon_{cplf}$ ) in eqn (1) and this equation ( $\epsilon_{cpls}$ ) are numerically the same but in order to track the parameter paths in the solution procedure different symbols are selected. Finally  $\dot{Y}[\tau]$  appears in the linear inertia term and represents the principal kinetic part of the out-of-plane motion of the secondary beam. No sphere is defined for this node, as the perturbation expansion has not yet been introduced.



$$\begin{array}{ccc} \omega_2^2 & Y[\tau] + & 2_{dmps} \quad \epsilon_{dmps} \\ (2,0) & & (2,1) \quad (2,1) \end{array}$$

$$\begin{array}{ccc} \zeta_2 & \omega_2 & Y'[\tau] - \\ (2,1) & (2,1) & \end{array}$$

$$\begin{array}{ccc} \epsilon_{cpls} & Y[\tau] & X''[\tau] + \\ (2,1) & & \end{array}$$

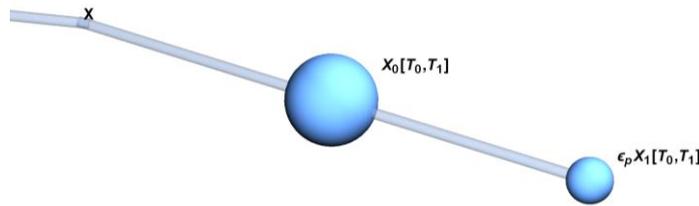
$$Y''[\tau] = 0$$

Figure 6-7 Overall visualisation graph for eqn (2).

The multiple scales perturbation expansion is applied to the dependent variables  $X$  and  $Y$ , up to the first order correction. The result is given in eqn (3), where; the small perturbation parameter is shown by  $\epsilon_p$ , fast time scale is shown by  $T_0$  and slow time scale by  $T_1$ .

$$X = X_0[T_0, T_1] + \epsilon_p X_1[T_0, T_1] \tag{eqn (3)}$$

Considering Figure 6-8, the sphere representing the zeroth-order term is considerably larger than the first order. This is a fundamental term therefore no visualisation link is shown here. In this figure  $X$  is the dependent variable and has not yet been encoded. Furthermore, the zeroth-order perturbation term,  $\epsilon_p X_1$ , is introduced for the first time. The first level of the SEEM can be identified in this Figure. As expected the sphere for the zeroth-order term is much larger than that for the first order perturbational correction term.



$X ==$

$$\begin{matrix} X_0 [T_0, T_1] & + & \epsilon_p & X_1 [T_0, T_1] \\ (3, 0) & & (3, 1) & (3, 1) \end{matrix}$$

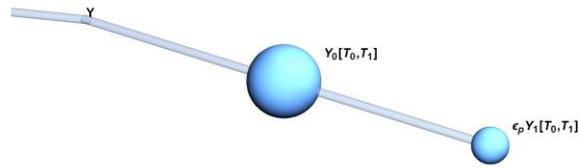
Figure 6-8 Overall visualisation graph for eqn (3).

The solution procedure progresses by applying the multiple scales perturbation expansion (up to the first order correction) to the dependent variable  $Y$ , resulting in eqn (4). As both epsilons in eqn (3) and eqn (4) are sourced from the multiple scales perturbation expansion the same numerical value and symbol have been used for them both in this analysis.

$$Y = Y_0[T_0, T_1] + \epsilon_p Y_1[T_0, T_1] \tag{eqn (4)}$$

Figure 6-9 shows the visualisation graph describing eqn (4). In this Figure  $Y_0$  is the zeroth-order perturbation term for the transverse displacement of the secondary beam and the  $\epsilon_p Y_1$  is the first order correction for the transverse displacement of the secondary beam. The spherical

shape representing the SF number for  $Y_0$  being considerably larger than  $\epsilon_p Y_1$ . This is a fundamental equation therefore no visualisation link is introduced.



$Y ==$

$$Y_0 [T_0, T_1] + \epsilon_p Y_1 [T_0, T_1]$$

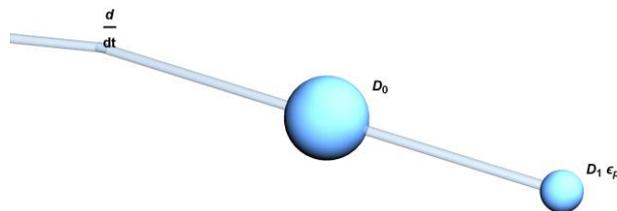
$(4, 0) \quad (4, 1) \quad (4, 1)$

Figure 6-9 Overall visualisation graph for eqn (4).

After these initial declarations, the first time derivative is introduced using the multiple scales perturbation expansion resulting in eqn (5) where the D-operator notation is used to define the partial derivatives:  $D_i = \partial / \partial T_i$ .

$$dt[1] = D_0 + D_1 \epsilon_p \tag{eqn (5)}$$

The visualisation graph for this equation is shown in Figure 6-10. Considering the SEEM term-tracking information,  $dt[1]$  is the total first time derivative and this is decomposed into a power series sum of partial derivatives by means of the perturbation expansion.  $D_0$  is the time derivative with respect to the time scale  $T_0$  and the perturbation order of this term is zero and it is encoded to the first level of the SEEM.  $D_1 \epsilon_p$  is the perturbational correction term which is to first perturbation order here, noting that the whole perturbation scheme for the problem is truncated after all first order perturbation correction terms. This term is also expressed to a first level of SEEM encoding. Clearly the SF value for  $D_0$  is much larger than  $D_1 \epsilon_p$  which indicates that the point scaling for the SF value appears to be numerically reasonable.



$$dt[1] == D_0 + D_1 \epsilon_p$$

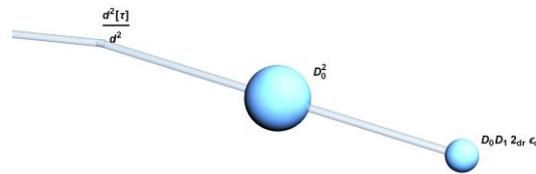
$(5, 0) \quad (5, 1) \quad (5, 1)$

Figure 6-10 Overall visualisation graph of eqn (5).

The second time derivative is determined using the multiple scales perturbation expansion, eqn (6), again based on the algebra of the D-operator notation. A unique symbol for the small perturbation parameter,  $\epsilon_{dr}$ , is introduced by choice of the user. The numerical value of  $\epsilon_{dr}$  is clearly equal to  $\epsilon_p$ , as both are part of the same perturbation expansion, but once again different sub-scripting is employed in order for clarity. This series is also truncated after the first order perturbation correction, as shown in eqn (6).

$$dt[2] = D_0^2 + \epsilon_{dr} 2_{dr} D_0 D_1 \tag{6}$$

The visualisation and encoding information for this equation are shown in Figure 6-11; where,  $dt[2]$  is the total second time derivative and  $D_0^2$  is the second time derivative with respect to time scale  $T_0$ , which is clearly to zeroth-order perturbation. One can see that  $\epsilon_{dr} 2_{dr} D_0 D_1$  is the next and final term in this series. In this term all the quantities are to first perturbation order and at the first level of SEEM encoding. The SF value for the zeroth-order ( $D_0^2$ ) term is much higher than that for the first order term ( $\epsilon_{dr} 2_{dr} D_0 D_1$ ), again as one would expect.



**dt [2] ==**

$$\begin{matrix}
 \mathbf{D_0^2} & + & \mathbf{D_0 D_1} & \mathbf{2_{dr}} & \mathbf{\epsilon_{dr}} \\
 \mathbf{(6,0)} & & \mathbf{(6,1)} & \mathbf{(6,1)} & \mathbf{(6,1)}
 \end{matrix}$$

Figure 6-11 Overall visualisation graph of eqn (6).

Figure 6-12 Shows the fundamental equation in the analysis of the autoparametrical excitation of a coupled beam system. Obviously no visualisation link yet has been introduced.

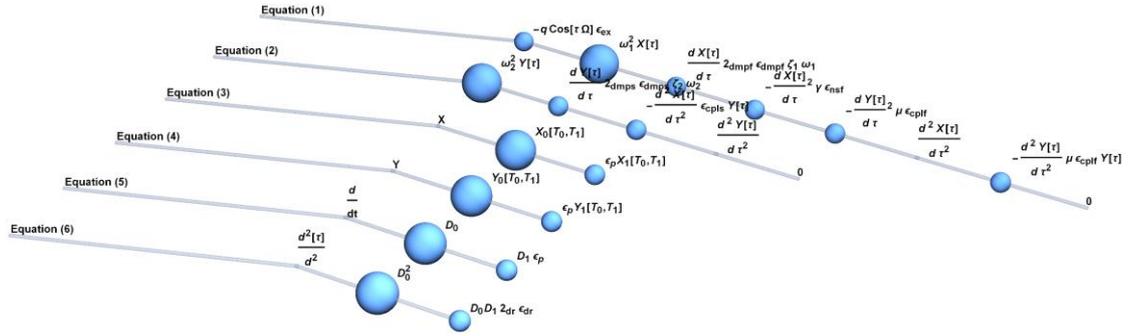


Figure 6-12 Fundamental equations for the problem of autoparametrically excitation of a coupled beam system.

The multiple scales perturbation expansion and the series forms for the first and second time derivatives are substituted into the equation of motion of the primary beam, resulting in:

$$D_0^2 X_0 + D_0 2_{dmpf} X_0 \epsilon_{dmpf} \zeta_1 \omega_1 + \omega_1^2 X_0 + D_0 D_1 2_{dr} X_0 \epsilon_{dr} - q \text{Cos}[T_0 \Omega] \epsilon_{ex} \quad \text{eqn (7)}$$

$$- D_0^2 \gamma X_0^2 \epsilon_{nsf} + D_0^2 X_1 \epsilon_p + \epsilon_p \omega_1^2 X_1 - D_0^2 \mu Y_0 \epsilon_{cplf} Y_0 = 0$$

Figure 6-13 shows the general visualisation graph for this equation. In this Figure the dense pink links are used to show the strong terms (to zeroth perturbation order), whilst the slim blue links are representing the weaker terms (to first perturbation order). Furthermore it is possible to identify the exact source of each quantity from the upstream terms.

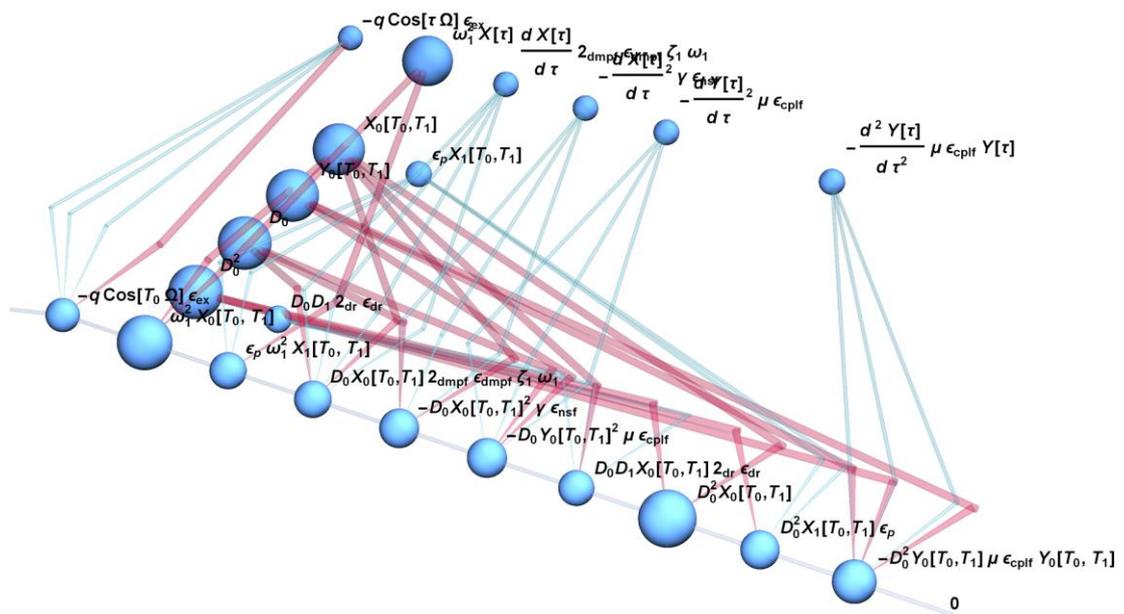


Figure 6-13 Overall visualisation graph for eqn (7).

A detailed analysis of both the SEEM term-tracking and the *Blueprint* visualisation methods for each term is shown in Figure 6-14. The external excitation term is shown in Figure 6-14-a where the amplitude of excitation  $q$  is scaled by introducing a small parameter  $\varepsilon_{ex}$  and the order of the excitation is shown by one. The frequency of the external excitation ( $\Omega$ ) is to zeroth perturbation order and its visualisation link is shown in pink and is denser than the other links.

Figure 6-14-b represents the linear stiffness term and the natural frequency is directly linked to the equation of motion, whilst  $X_0$  is linked to the perturbation equation. All the links for this term are in pink and the encoding information is shown by the *labeled* function on the links.

Figure 6-14-c shows the stiffness term which is introduced purely through the choice of the perturbation expansion. The first linear natural frequency is the only zeroth-order quantity in this term and is linked to the equation of motion for the primary beam.

Figure 6-14-d shows the damping term which is scaled by introducing  $\varepsilon_{dmpf}$ . The time derivative operator is linked to eqn (5) and shown in pink. Also Figure 6-14-e depicts the coefficient of centripetal acceleration term defined by  $\gamma$ , and this is scaled by the use of  $\varepsilon_{nsf}$ . There are three zeroth perturbation order and two first perturbation order quantities in this term.

Furthermore Figure 6-14-f presents the coupling term. The effective mass ratio  $\mu$  considered to be small by introducing  $\varepsilon_{cplf}$ . Figure 6-14-g is the inertia term and in the perturbation scheme this is influenced by the introduction of the perturbation series form for second order differentiation with respect to time. It is possible to notice that all the quantities, except  $X_0$ , are linked to equation (6). In Figure 6-14-h the inertia term can be seen and this defines the principal part of the kinetic effects, which in turn represent the main part of the motion of the primary beam. This term is purely to zeroth perturbation order and all the links are shown in pink.

Finally, Figure 6-14-i is the second and final term emanating from the perturbation scheme and directly affecting the inertial quantity for the secondary beam. This term is a function of the multiple scales method, and two out of three quantities in this term are linked to eqn (3).

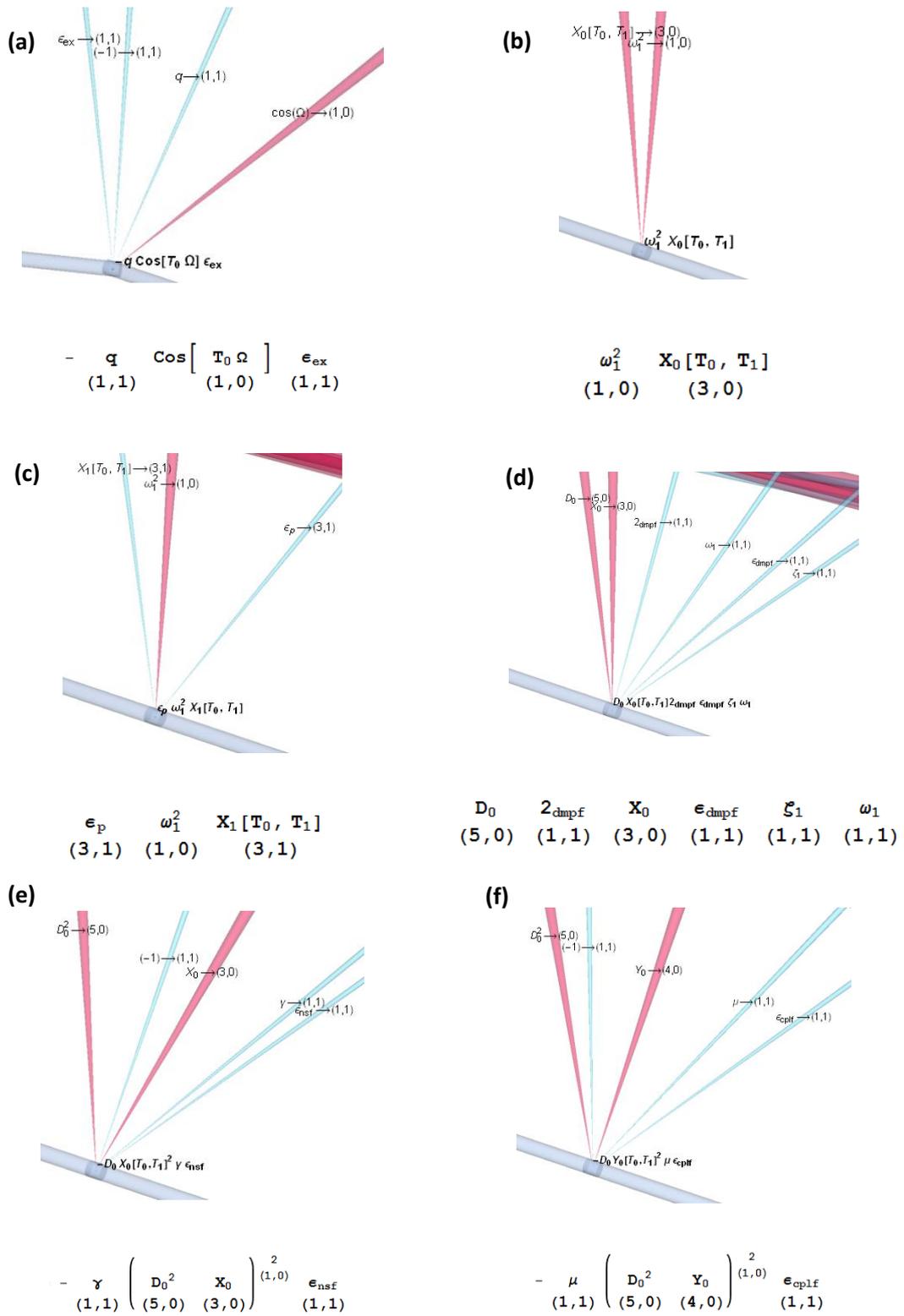


Figure 6-14 continued on next page.

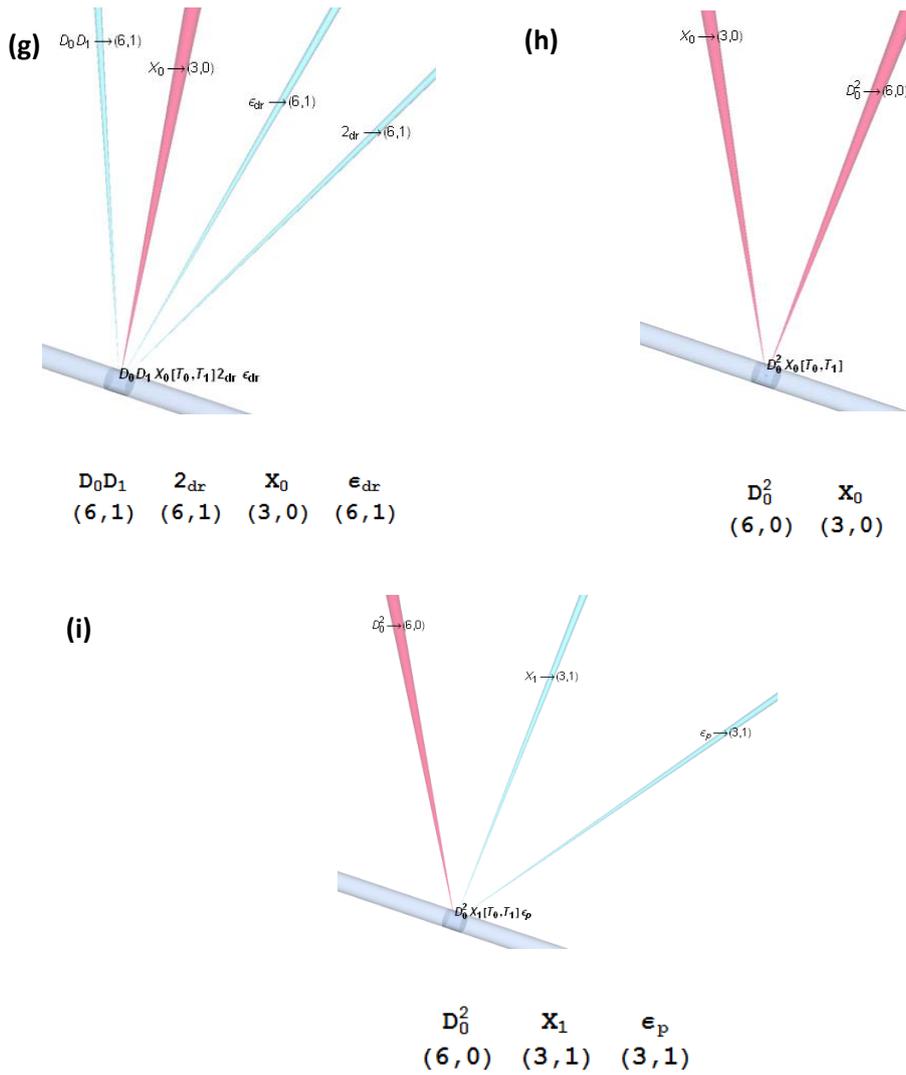


Figure 6-14 A detailed investigation of the encoding for eqn (7); (a) is the external excitation term, (b) is the linear stiffness, (c) is a stiffness term, (d) is the damping term, (e) and f are coupling terms, (g) is a perturbational correction term, and finally (h) and (i) are the inertia terms.

The same procedure as above was carried out to derive the new form of the equation of motion for the secondary beam, up to the first perturbation order correction. The perturbation equations (4), (5) and (6) are substituted into equation (2), resulting in:

$$D_0^2 Y_0 + D_0 D_1 2_{dr} Y_0 \epsilon_{dr} + D_0^2 Y_1 \epsilon_p + D_0 2_{dmps} Y_0 \epsilon_{dmps} \zeta_2 \omega_2 - D_0^2 X_0 \epsilon_{cpls} Y_0 + \omega_2^2 Y_0 + \epsilon_p \omega_2^2 Y_1 = 0 \tag{eqn (8)}$$

Figure 6-15 shows the sources and links for each term in eqn (8). The inertia and linear stiffness terms have larger SF spheres as compared with the other terms. It is possible to follow each link to the source terms located in the upstream direction of the equation line.

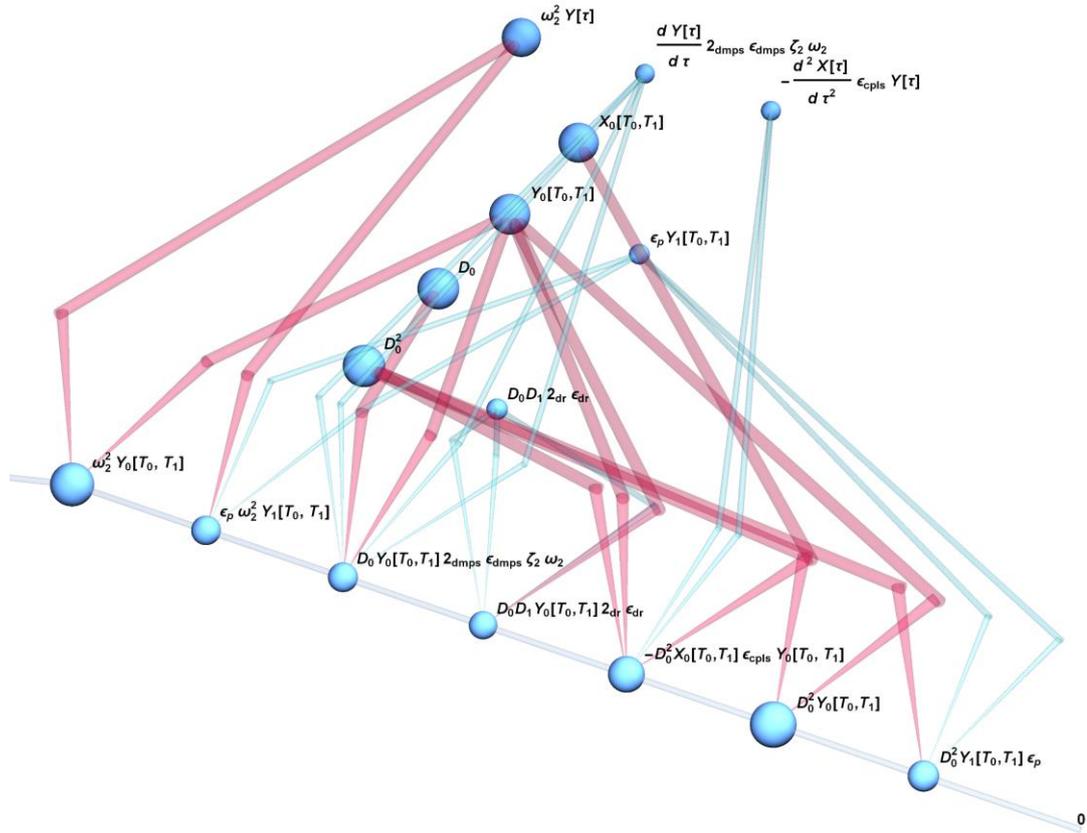


Figure 6-15 Overall visualisation graph for eqn (8).

A detailed investigation of some selected terms from this equation is provided in Figure 6-16. Figure 6-16-a is the stiffness term for the secondary beam, and obviously plays an important role in the response of the system. Figure 6-16-b is the part of the stiffness term for the secondary beam relating to the first order correction in the perturbation series. There are two first order quantities and one zeroth-order quantity in this term.

Figure 6-16-c shows the damping term structured from equations 2, 4, and 5. Figure 6-16-d is a perturbational correction term generated by the series used in the multiple scales method. It has only one strong zeroth-order quantity, and the rest of the terms are linked in different ways to the multiple scales process.

Figure 6-16-e is a coupling term. It is assumed that the coupling term is small through introducing  $\epsilon_{cpIs}$  at this point. This term has three zeroth perturbation order quantities in its structure.

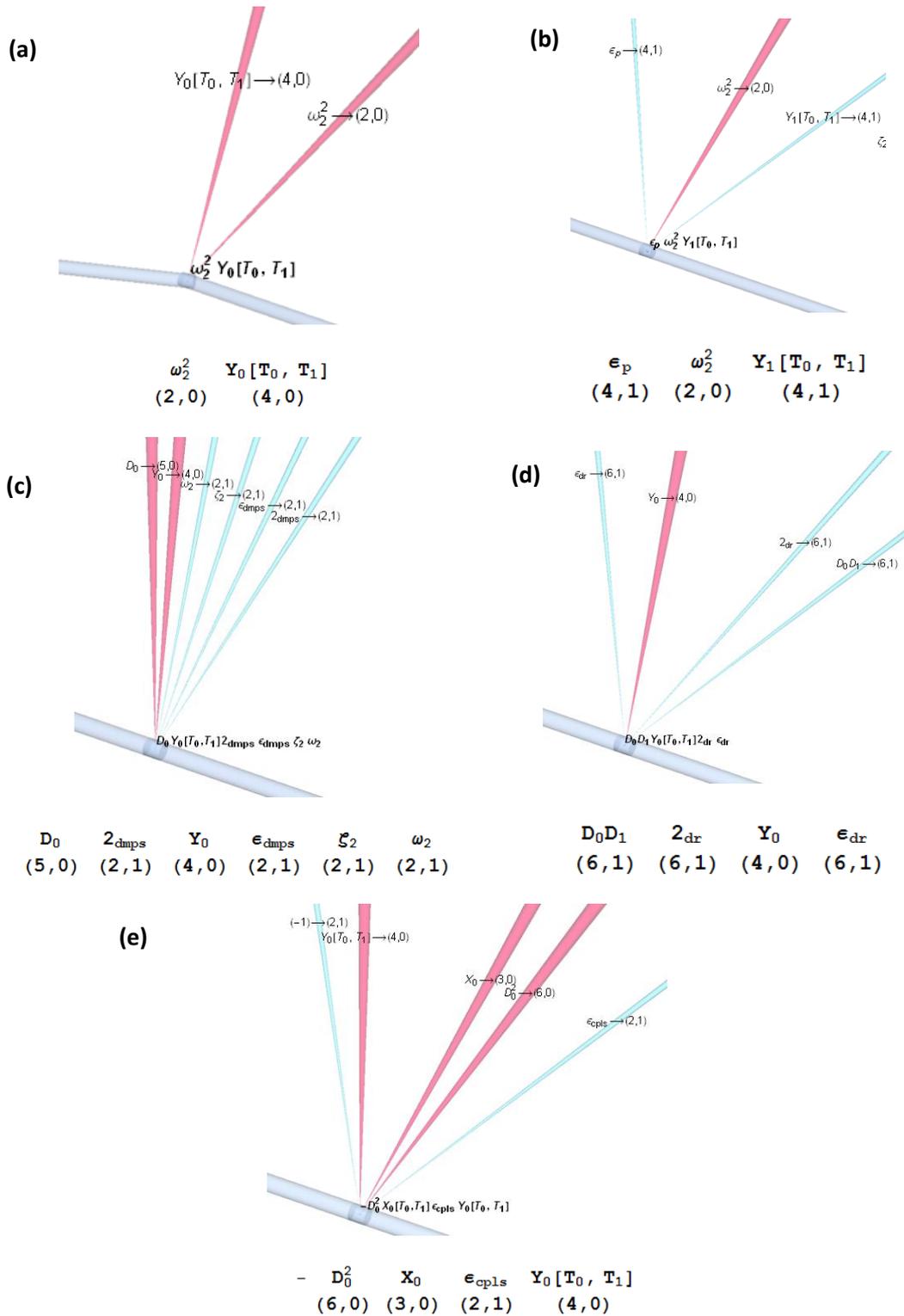
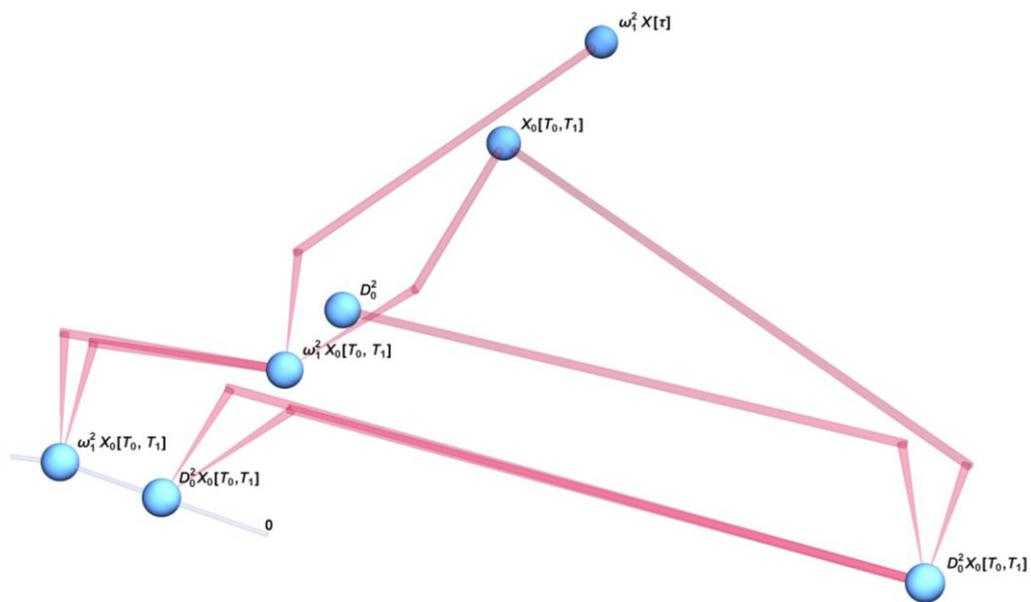


Figure 6-16 A detailed investigation of the encoding information for eqn (8); where, (a) is a stiffness term, (b) is a perturbational correction term, (c) is the damping term, (d) is a perturbational correction term, and (e) is the coupling term.

The zeroth-order terms from eqn (7) are selected to form the zeroth-order perturbation equation, as follows:

$$D_0^2 Y_0 + \omega_2^2 Y_0 [T_0, T_1] = 0 \tag{eqn (9)}$$

Figure 6-17 shows all the sources and links contributing in the zeroth-order perturbation equation structure for the primary system. The *Blueprint* visualisation method successfully shows all the links in dense pink format and the SF values for all terms are approximately the same. Considering the SEEM information the inertia term is linked to eqn (3) and (6), and the stiffness term has a direct link back to the equation of motion.



$$\begin{array}{c}
 D_0^2 \quad X_0 \quad + \\
 (6, \theta) \quad (3, \theta) \\
 \omega_1^2 \quad X_0 [T_0, T_1] == \theta \\
 (1, \theta) \quad (3, \theta)
 \end{array}$$

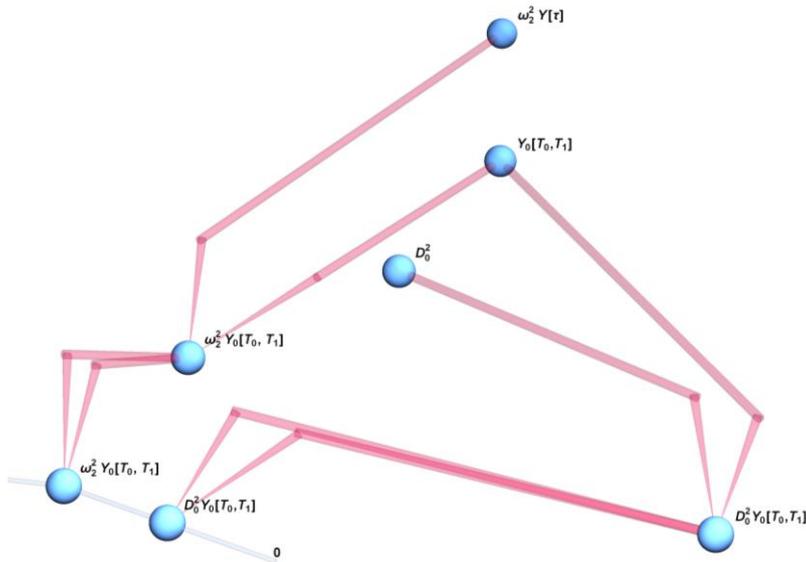
Figure 6-17 A detailed investigation of the encoding information for eqn (9).

The zeroth perturbation order terms from eqn (8) are selected and set to zero, to form the zeroth-order perturbation equation, as follows:

$$D_0^2 Y_0 + \omega_2^2 Y_0 [T_0, T_1] = 0 \tag{eqn (10)}$$

Figure 6-18 shows all the sources and links that constitute the zeroth-order perturbation equation for the secondary beam. All the links are displayed with dense pink lines and the SF

values for all terms are roughly the same. Again the user can quickly get the importance of this equation by considering this Figure.



$$\begin{matrix}
 D_0^2 & Y_0 & + \\
 (6, \theta) & (4, \theta) & \\
 \omega_2^2 & Y_0 [T_0, T_1] & = 0 \\
 (2, \theta) & (4, \theta) &
 \end{matrix}$$

Figure 6-18 A detailed investigation of the encoding information for eqn (10).

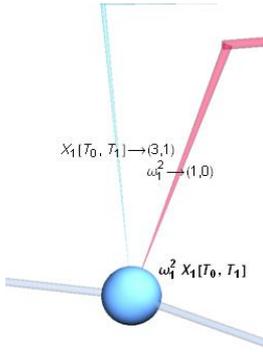
The first order perturbation equation for the primary beam is shown in eqn (11). This equation is created by taking the terms containing  $\epsilon^1$  out from eqn (7) and setting them to zero.

$$\begin{aligned}
 D_0^2 X_1 + \omega_1^2 X_1 [T_0, T_1] & \qquad \qquad \qquad \text{eqn (11)} \\
 = \frac{D_0^2 \mu Y_0^2 \epsilon_{cplf}}{\epsilon_p} - \frac{D_0 D_1 2_{dr} X_0 \epsilon_{dr}}{\epsilon_p} + \frac{q \text{Cos}[T_0 \Omega] \epsilon_{ex}}{\epsilon_p} \\
 + \frac{D_0^2 \gamma X_0^2 \epsilon_{nsf}}{\epsilon_p} - \frac{D_0 2_{dmpf} X_0 \epsilon_{dmpf} \zeta_1 \omega_1}{\epsilon_p} + \frac{D_0^2 \mu Y_0 \epsilon_{cplf} Y_0}{\epsilon_p}
 \end{aligned}$$

The overall visualisation graph for eqn (11) is given in Figure 6-19 and it is possible to notice that terms with more than one zeroth perturbation order quantity have a noticeably larger SF representation. The contribution from the primary equation of motion and the fundamental equations used within the method of multiple scales can be noticed in the upstream of the equation line. Furthermore, it is possible to interpret that highlighted terms in eqn (7) are merely re-arranged and then appeared in equation (11).

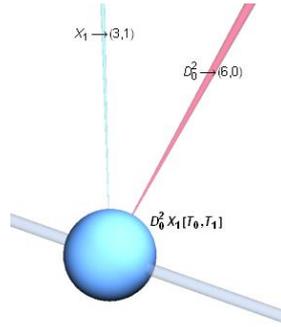


(a)



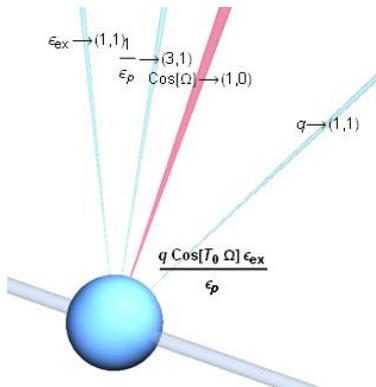
$$\frac{\omega_1^2 \quad \mathbf{X}_1 [T_0, T_1]}{(1, 0) \quad (3, 1)}$$

(b)



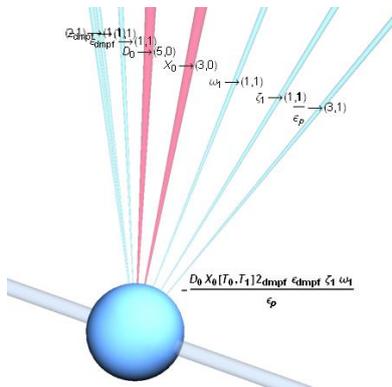
$$\frac{D_0^2 \quad \mathbf{X}_1}{(6, 0) \quad (3, 1)}$$

(c)



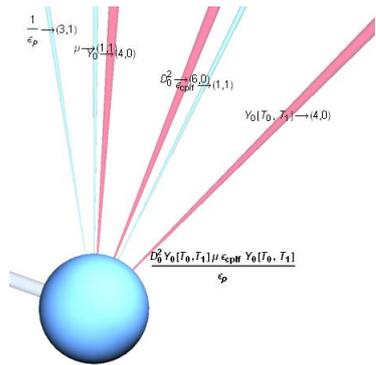
$$\frac{q \quad \text{Cos} [ T_0 \Omega ] \quad \epsilon_{ex}}{(1, 1) \quad (1, 0) \quad (1, 1)} \\ \epsilon_p \\ (3, 1)$$

(d)



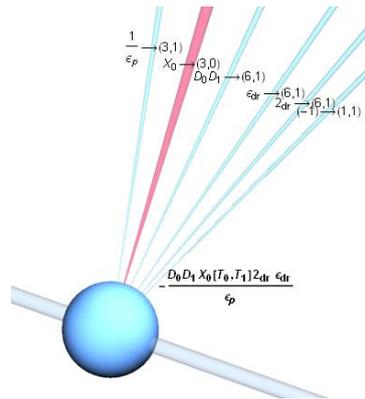
$$\frac{D_0 \quad 2 \text{dmp} f \quad \mathbf{X}_0 \quad \epsilon_{\text{dmp} f} \quad \xi_1 \quad \omega_1}{(5, 0) \quad (1, 1) \quad (3, 0) \quad (1, 1) \quad (1, 1) \quad (1, 1)} \\ \epsilon_p \\ (3, 1)$$

(e)



$$\frac{D_0^2 \quad \mu \quad \mathbf{Y}_0 \quad \epsilon_{\text{epl} f} \quad \mathbf{Y}_0 [T_0, T_1]}{(6, 0) \quad (1, 1) \quad (4, 0) \quad (1, 1) \quad (4, 0)} \\ \epsilon_p \\ (3, 1)$$

(f)



$$\frac{D_0 D_1 \quad 2 \text{dr} \quad \mathbf{X}_0 \quad \epsilon_{\text{dr}}}{(6, 1) \quad (6, 1) \quad (3, 0) \quad (6, 1)} \\ \epsilon_p \\ (3, 1)$$

Figure 6-20 A detailed investigation of eqn (11); (a) is a linear stiffness, (b) is the inertia, (c) is the external excitation, (d) is a damping term, (e) is a coupling term, and finally (f) is a perturbational correction term.

Eqn (12) is the first order perturbation equation for the secondary beam. This equation is obtained by taking terms with the coefficient of  $\varepsilon^1$  from eqn (8), resulting in:

$$D_0^2 Y_1 + \omega_2^2 Y_1 [T_0, T_1] = -\frac{D_0 D_1 2_{dr} Y_0 \varepsilon_{dr}}{\varepsilon_p} - \frac{D_0 2_{dmps} Y_0 \varepsilon_{dmps} \zeta_2 \omega_2}{\varepsilon_p} + \frac{D_0^2 X_0 \varepsilon_{cpls} Y_0}{\varepsilon_p} \tag{eqn (12)}$$

Figure 6-21 is a general overview of the sources and links for eqn (12). Equation (8) is highlighted in this graph, is a transitional area because no new terms are introduced at this stage.

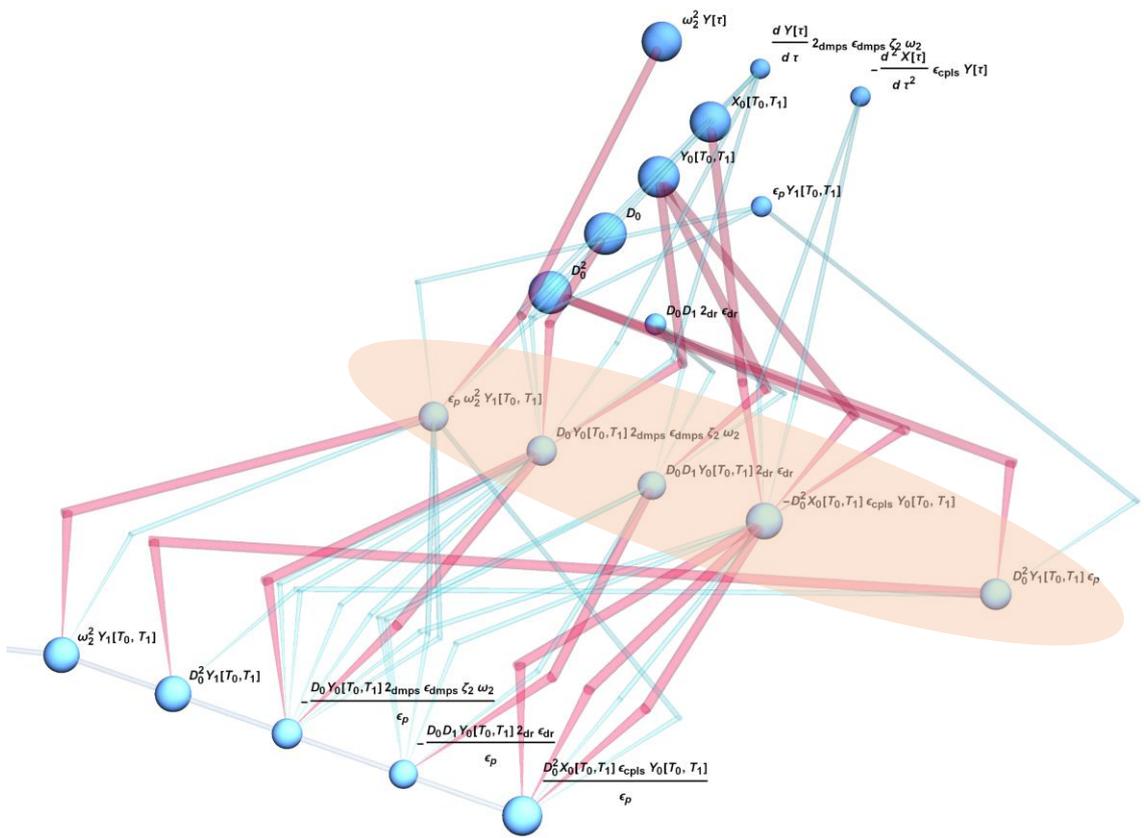


Figure 6-21 Overall visualisation graph for eqn (12).

Figure 6-22 shows a detailed process behind the visualising links for selected terms in this equation. The stiffness term is given in Figure 6-22-a, and the first linear natural frequency is linked to the equation of motion for the secondary beam. Also the inertia term is shown Figure 6-22-b where the time derivative is shown in pink and linked to eqn (6). The first order perturbation term for the secondary beam is shown in blue and linked to the perturbation expansion. Both terms in this equation are sourced to the perturbation expansion. Figure 6-22-

c shows the damping term for the secondary beam. This term is structured from equations 2, 4, and 5. Most of the links in this Figure are to first perturbation order and are shown by blue and the user can quickly see that the damping term is not strong. Finally, the perturbational correction term for this equation is given in Figure 6-22-d. This term is sourced to equations 4 and 6, and changing the solution method will obviously highly affect this term.

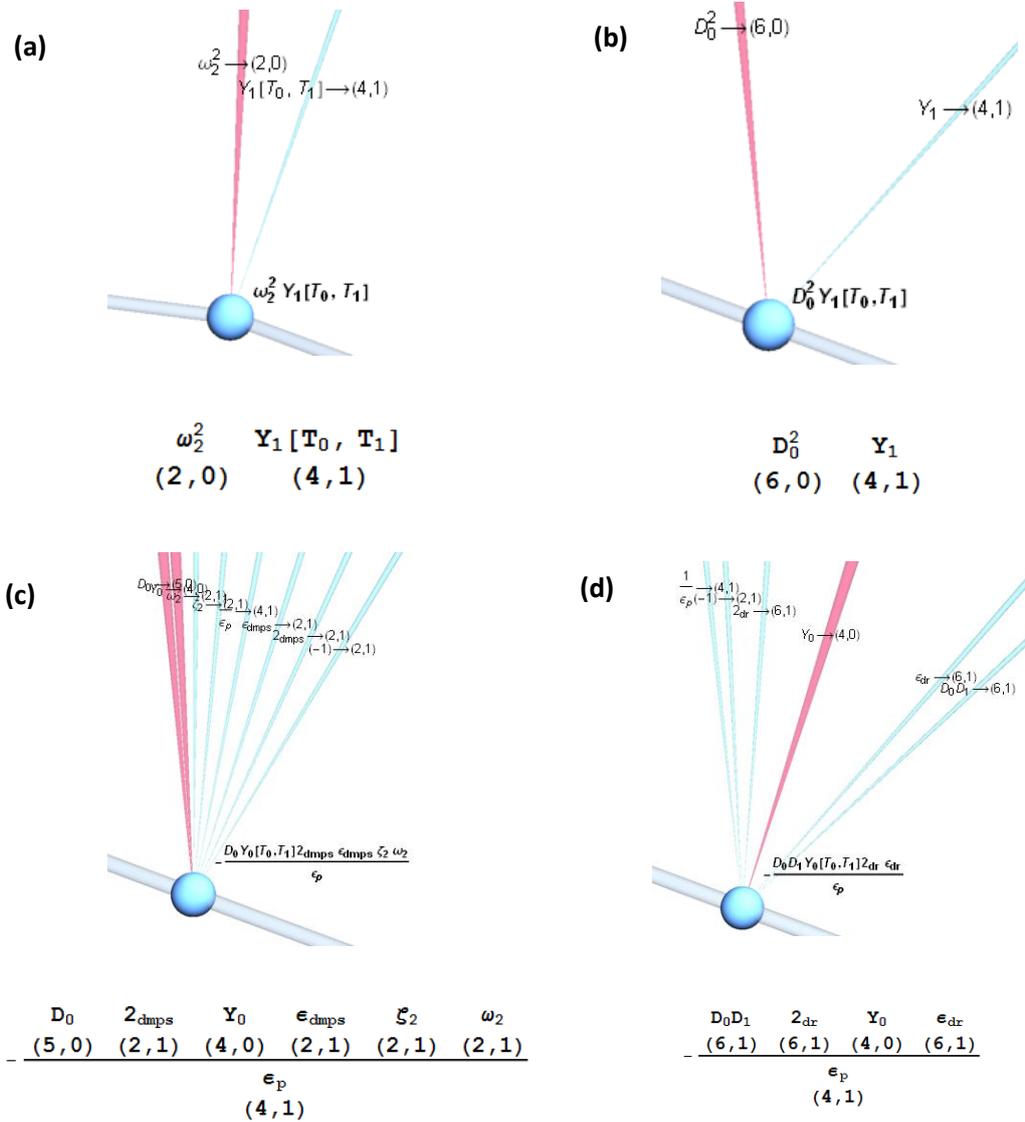


Figure 6-22 A detailed investigation of the encoding information for eqn (12); where, (a) is the stiffness term, (b) is the inertia, (c) is the damping term, and (d) is the perturbational correction term.

The solution procedure continues in Appendix-C of the thesis.

## 6.5 Post-processing the Strength Factor information

Considering the main logic of the SF value, which is identifying the less effective term in each equation, a scatter plot based on the terms and their SF values is created. As Figure 6-23 shows, this graph is added into the SCD solver. The user can change the equation number by adjusting the bar on the top of the graph, and the SF values for each term, in the selected equation, are given on the y-axis of the plot as functions of the term on the x-axis. Moreover, the exact SF value for each term is labelled near its node. This is just a presentation of the SF values and does not confer an automated decision process due to the algorithm.

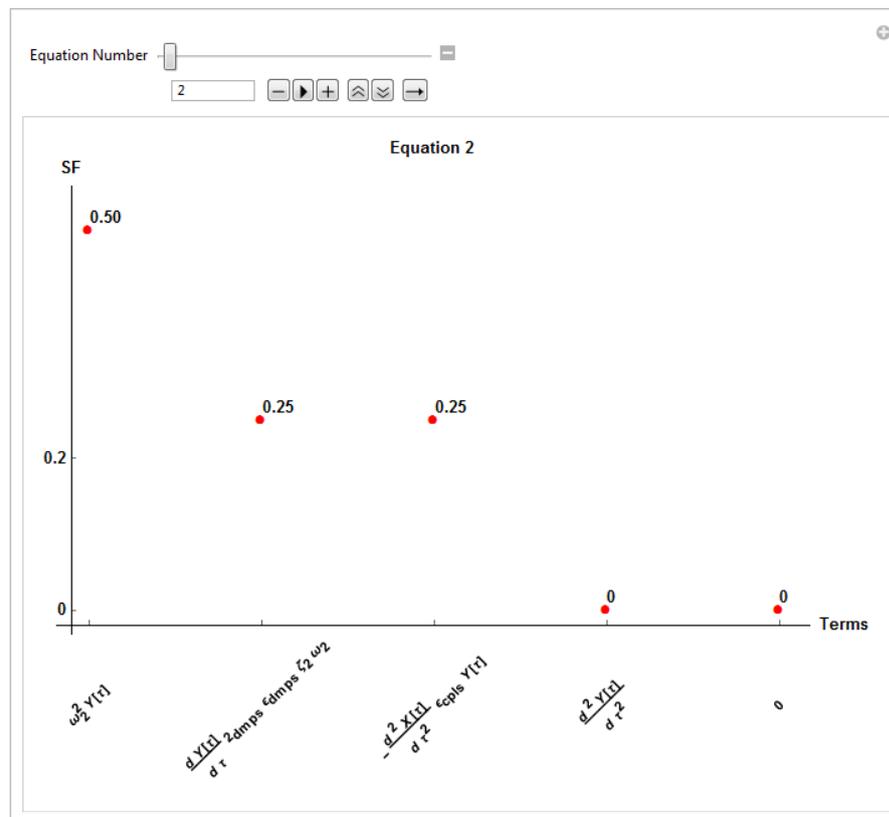


Figure 6-23 A typical Strength Factor graph for the eqn (2) of the autoparametrically excited coupled beam system.

### 6.5.1 The Strength Factor guideline

The main goal of this procedure is to develop an algorithm that can automatically identify the less important term in any equation based on both the terms SF number and the solution procedure concept. The fundamental aim has been to create a series of conditions whereby a quantity cannot be justifiably removed from an equation. This initially has done by manually checking the SF graph for each equation and then developing a guideline proposal. The SF guideline is summarised below:

1. No terms from the perturbation expansion and the differentiations equations must be removed. This is because the higher order terms are already truncated based on the mathematical model.
2. The user must define a threshold from which the SF number below that threshold can be considered to represent an insignificant term. The higher range of SF values indicate terms that are more important, because they have higher points in either, or both, the order and the SEEM hierarchy. As the SF value for each term is a number between 0 and 1, a threshold of 0.5 is regarded as a conservative decision point for removing the associated term. It is possible for the user to change this sensitivity, if this is required.
3. The algorithm must not remove the inertia term, or any external excitation terms, or indeed any fundamental dynamical term from any of the perturbation equations.
4. As explained in §3.3.1, there are some quantities that the SEEM encoding information is not defined intentionally. If there is no encoding information, the SF cannot be calculated for that quantity and makes the SF for that term equal to zero. As the results terms with zero SF value should stay in the analysis.

### 6.5.2 The Strength Factor analysis for autoparametric system

The SF graph for all equations in the coupled beam case study is plotted and the SF guideline is used to select negligible terms in each equation. The *original* response is calculated, by the SCD solver, without removing any terms. While, the response after removing a selected term based on the SF guideline is called *modified* response.

For each case to compare the *original* and *modified* responses, a set of numerical values are substituted in to these symbolic responses and the results are plotted in a single graph. The numerical values are initially adopted from experimental data provided in Cartmell [70], which is given in Table 6-3.

Table 6-3 Experimental data for the coupled beam system parameters [70].

Parameter	Numerical value
$\mu$	0.3
$\varepsilon$	0.5
$\omega_1$ (Hz)	50
$\omega_2$ (Hz)	25
$\xi_1$	0.01
$\xi_2$	0.001

It is important to show that the qualitative forms of the generated responses are not dependent on the specific numerical values of the data used to calculate them. Therefore, the *original* and the *modified* responses are plotted in a dynamic interface where the user can adjust the numerical values for the system.

As Figure 6-24 shows, the *original* and *modified* responses, based on the numerical data provided in Table 6-3. These response plots show all the classical characteristics of an autoparametrically resonant system [70]. Both *original* and *modified* responses based on this numerical data set are the same.

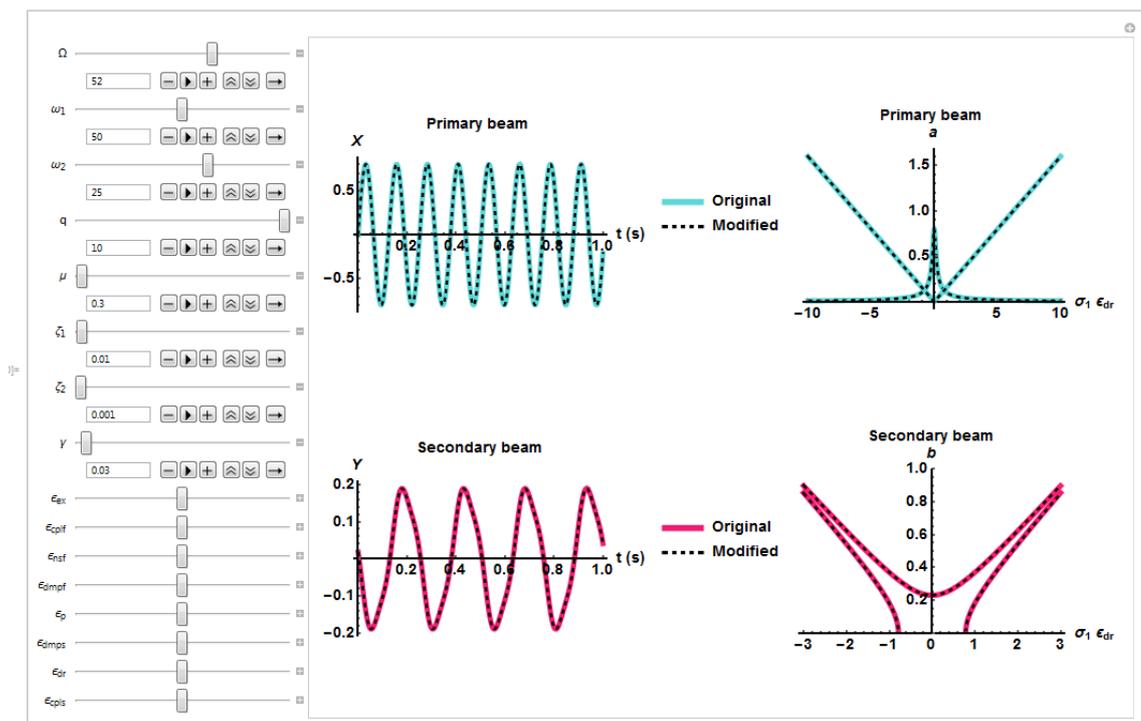


Figure 6-24 The *original* and *modified* amplitude responses for the primary (a) and secondary (b) beams (based on the numerical values given in Table 6-3).

It was then decided to change the numerical data values to ensure that plots shown in Figure 6-24 are indeed qualitatively independent from them, see Figure 6-25. It is possible to notice that the *original* and *modified* graphs are qualitatively identical in both plots, which strongly implies (but does not prove) that they are qualitatively independent of the numerical values used to calculate them.

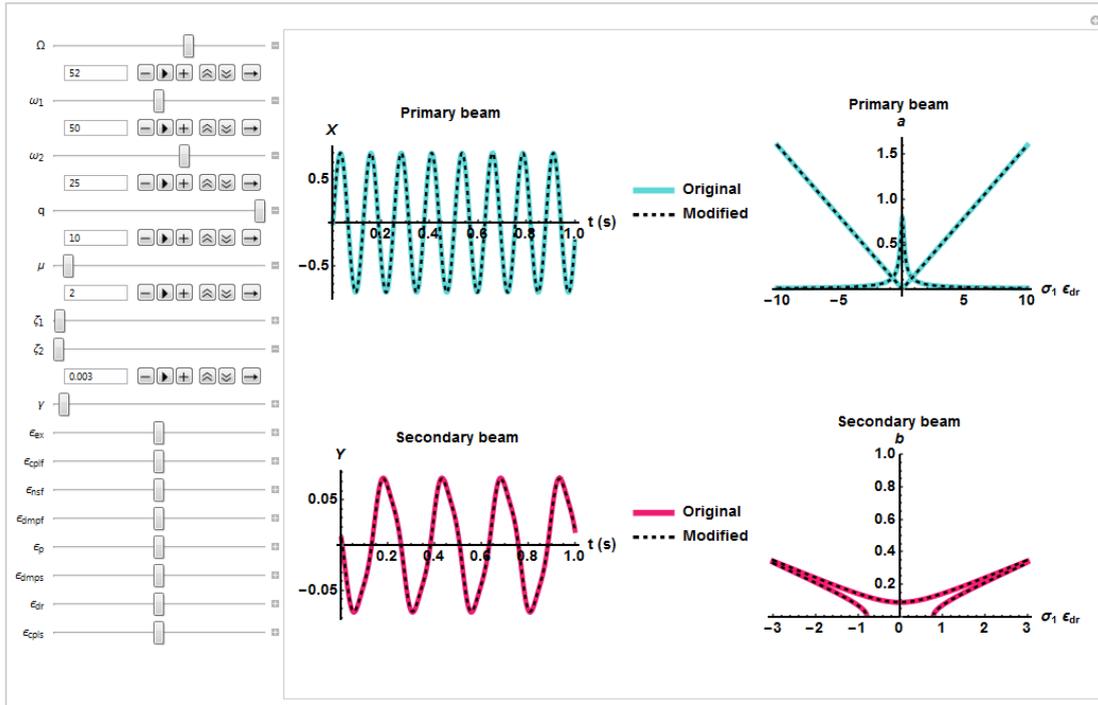


Figure 6-25 The *original* and *modified* amplitude responses for the primary and secondary beams (based on varied numerical input data values).

The SF plot for eqn (1) is provided in Figure 6-26. No term is selected under the first rule of the SF guideline, as this not either a perturbation expansion or the definition of time derivatives. Considering the second rule the chosen threshold is 0.5, therefore  $\omega_1^2 X[\tau]$  should stay in the equation. The third rule stops the user for removing fundamental terms such as; damping  $2\epsilon_{dmpf}\epsilon_{dmpf}\zeta_1\omega_1\dot{X}[\tau]$ , external excitation  $\epsilon_{ex}q\cos[\Omega\tau]$ , and principal coupling terms  $\epsilon_{cplf}\mu(\dot{Y}[\tau]^2 + \dot{Y}[\tau] * Y[\tau])$ . Finally, the fourth rule of the SF guideline indicates retaining the inertia term,  $\ddot{X}[\tau]$  which it is not yet encoded. Consequently, the guideline suggests removing the  $\dot{X}[\tau] * \dot{X}[\tau]\gamma\epsilon_{nsf}$  term. These results are also categorised in Table 6-4.

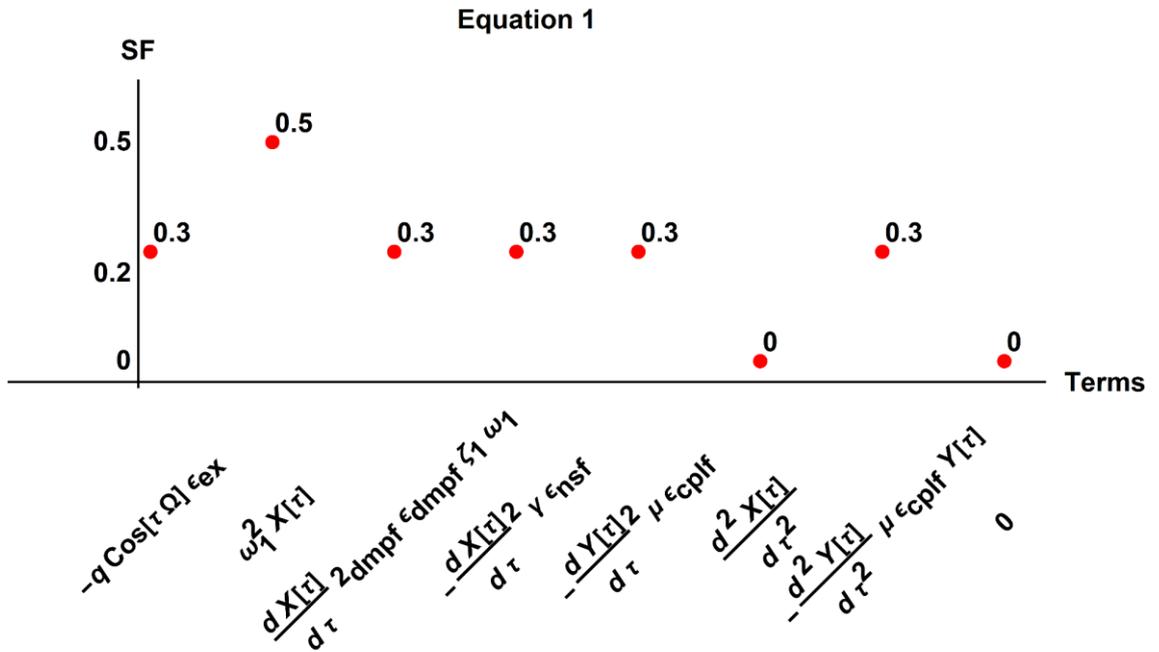


Figure 6-26 The Strength Factor graph of eqn (1).

Table 6-4 The Strength Factor guideline applied to eqn (1).

The Sf guideline rule number	Terms
1	-
2	$\omega_1^2 X[\tau]$
3	$2_{dmpf} \epsilon_{dmpf} \zeta_1 \omega_1 \dot{X}[\tau]$
	$\epsilon_{cplf} \mu (\dot{Y}[\tau]^2 + \ddot{Y}[\tau] * Y[\tau])$
	$\epsilon_{ex} q \cos[\Omega \tau]$
4	$\ddot{X}[\tau]$

Considering the SF selection, the *modified* response by removing  $\dot{X}[\tau] * \dot{X}[\tau] \gamma \epsilon_{nsf}$  from eqn (1) is then calculated. Then *original* and *modified* responses based on the numerical values are plotted, resulting in Figure 6-27 . Both the *original* and *modified* responses remain the same, and it is possible to remove this term permanently from this analysis.

It should be noticed that this term is also considered negligible in Cartmell [70], through observations mainly based on repeated numerical explorations in different problems, typified by differential equations incorporating autoparametric resonances .

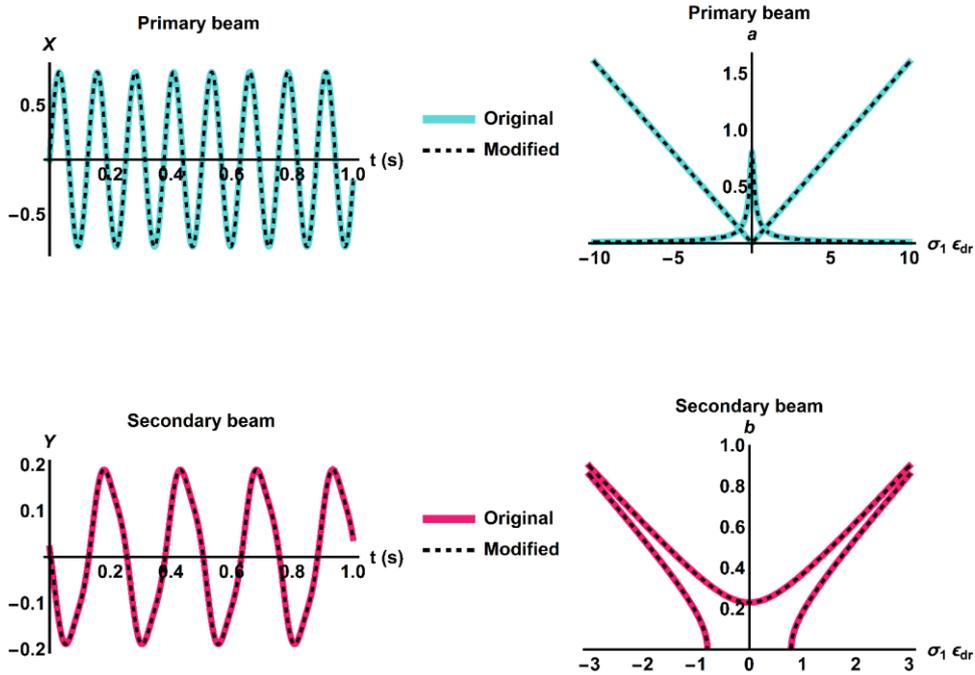


Figure 6-27 The *original* and *modified* responses after removing  $\dot{X}[\tau] * \dot{X}[\tau] \gamma \epsilon_{nsf}$ .

The same procedure as eqn (1) is repeated for eqn (7) in the study of the autoparametric excitation of coupled beam systems. The SF graph for this equation is provided in Figure 6-28. As this equation is not categorised as the definition of either perturbation expansion or the differentiation equations, rule number one of the SF guideline is not valid. Based on the second rule of the SF guideline, terms with the SF higher than 0.5 should remain in the equation structure, which means,  $\omega_1^2 X_0 [T_0, T_1]$  and  $D_0^2 X_0 [T_0, T_1]$ . The third rule of the SF guideline dictates the fundamental terms; such as, damping  $D_0 2_{dmpf} X_0 \epsilon_{dmpf} \zeta_1 \omega_1$ , external excitation  $q \text{Cos}[T_0 \Omega] \epsilon_{ex}$ , and the coupling term  $D_0 Y_0^2 \mu \epsilon_{cplf}$  must stay in the equation structure. Finally, number zero based on the fourth rule is retained in this equation. These results are also summarised in Table 6-5.

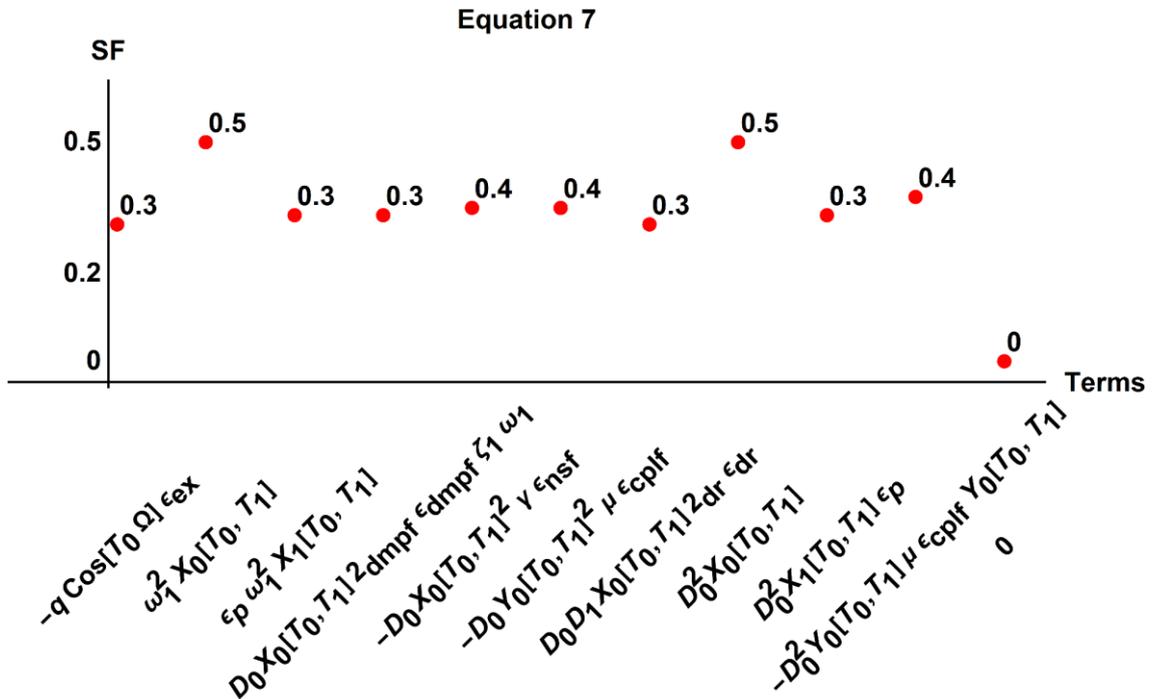


Figure 6-28 The Strength Factor plot of eqn (7).

Table 6-5 The Sterength Factor guideline applied to eqn (7).

The Sf guideline rule number	Terms
1	-
2	$D_0^2 X_0$
	$\omega_1^2 X_0$
3	$D_0 2_{dmpf} X_0 \epsilon_{dmpf} \zeta_1 \omega_1$
	$-q \text{Cos}[T_0 \Omega] \epsilon_{ex}$
	$D_0 Y_0^2 \mu \epsilon_{cplf}$
4	0

As the results,  $2_{dr} \epsilon_{dr} D_0 D_1 X_0$ , and  $\dot{X}[\tau] * \dot{X}[\tau] \gamma_{\epsilon_{nsf}}$ ,  $D_0^2 X_1 \epsilon_p$ , and  $\epsilon_p \omega_1^2 X_1$  are considered less effective terms in this equation. As discussed in Figure 6-29 removing  $\dot{X}[\tau] * \dot{X}[\tau] \gamma_{\epsilon_{nsf}}$  has no particular effect on the response of the system.

The *modified* repose of the system by removing  $2_{dr} \epsilon_{dr} D_0 D_1 X_0$ , term is investigated. Figure 6-30 shows the *original* and *modified* responses, it is possible to notice removing  $2_{dr} \epsilon_{dr} D_0 D_1 X_0$  significantly altered the secondary beam response of the system. Consequently, this term should not be removed from this equation.

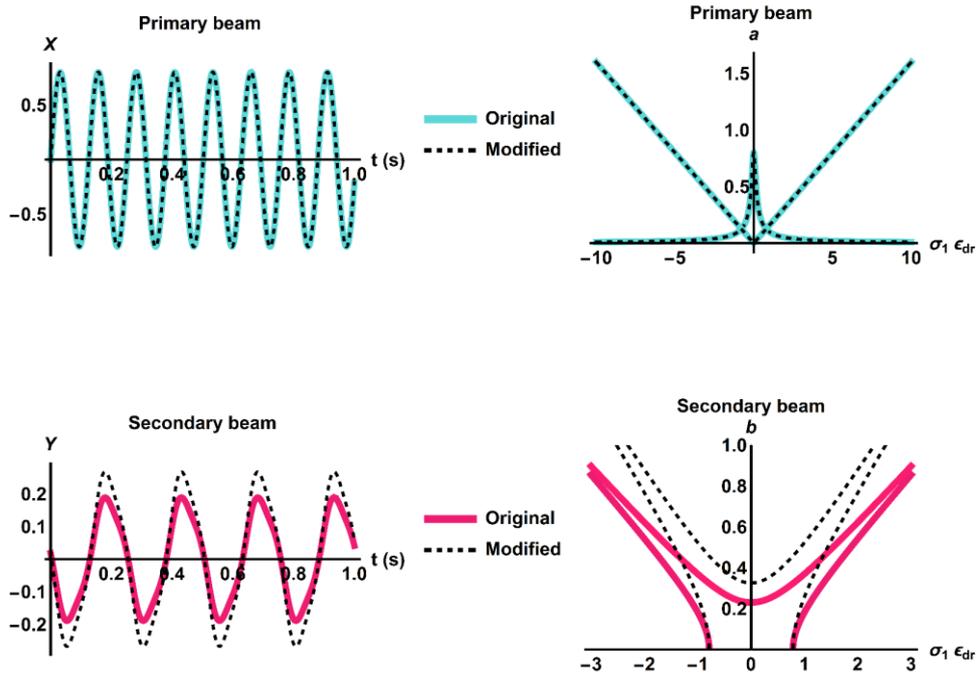


Figure 6-30 The *original* and *modified* responses after removing  $2_{dr}\epsilon_{dr}D_0D_1X_0$ .

Then the *modified* responses for  $D_0^2X_1\epsilon_p$ , and  $\epsilon_p\omega_1^2X_1$  individually are calculated. As Figure 6-31 and Figure 6-32 show, removing either of these terms do not affect the response of this system. As the results  $D_0^2X_1\epsilon_p$ , and  $\epsilon_p\omega_1^2X_1$  can be neglected, while  $2_{dr}\epsilon_{dr}D_0D_1X_0$  should be retained in this equation.

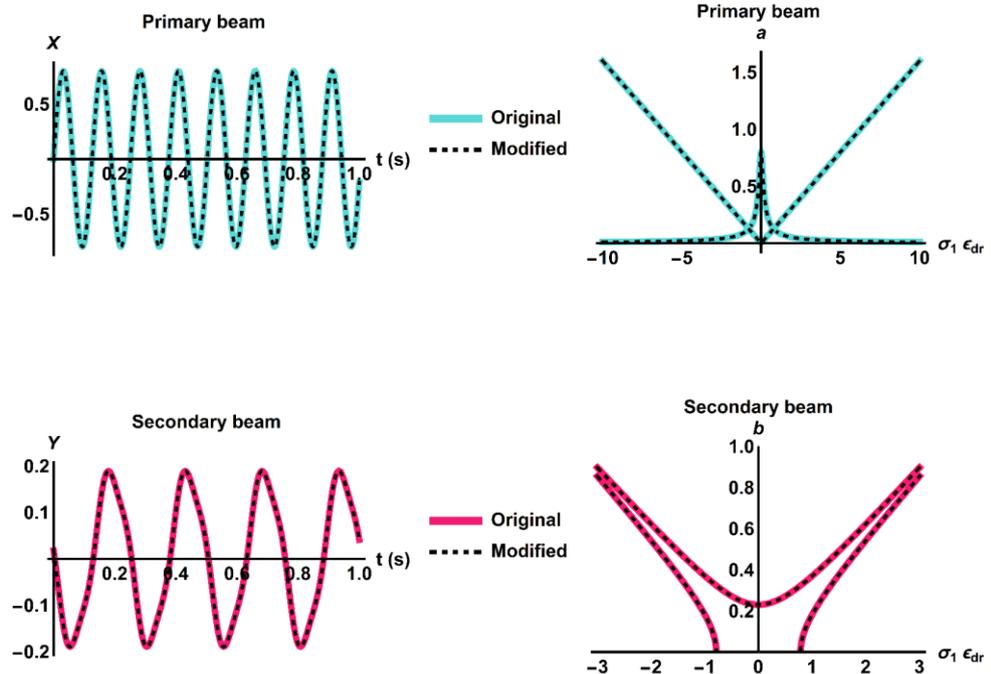


Figure 6-31 The *original* and *modified* responses after removing  $D_0^2 X_1 \epsilon_p$ .

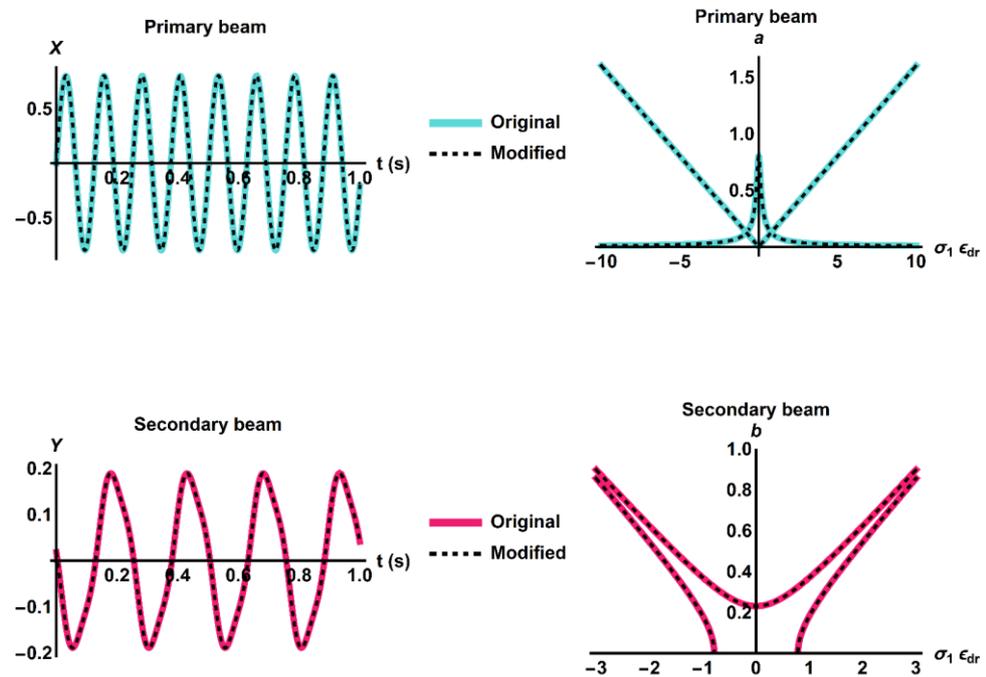


Figure 6-32 The *original* and *modified* responses after removing  $\epsilon_p \omega_1^2 X_1$ .

The analysis continues by plotting the SF graph for eqn (8) and this is shown in Figure 6-30. The maximum SF value number is almost equal to the threshold, therefore the  $\omega_2^2 Y_0$  and  $D_0^2 Y_0$  terms must be retained. Furthermore,  $D_0^2 X_0 \varepsilon_{cpls} Y_0$  and  $D_0 2_{dmps} Y_0 \varepsilon_{dmps} \zeta_2 \omega_2$  are classified as fundamental terms and they should stay in the equation. Consequently, the SF guideline suggests removing  $D_0^2 Y_1 \varepsilon_p$ ,  $\varepsilon_p \omega_2^2 Y_1$ , and  $D_0 D_1 2_{dr} Y_0 \varepsilon_{dr}$ .

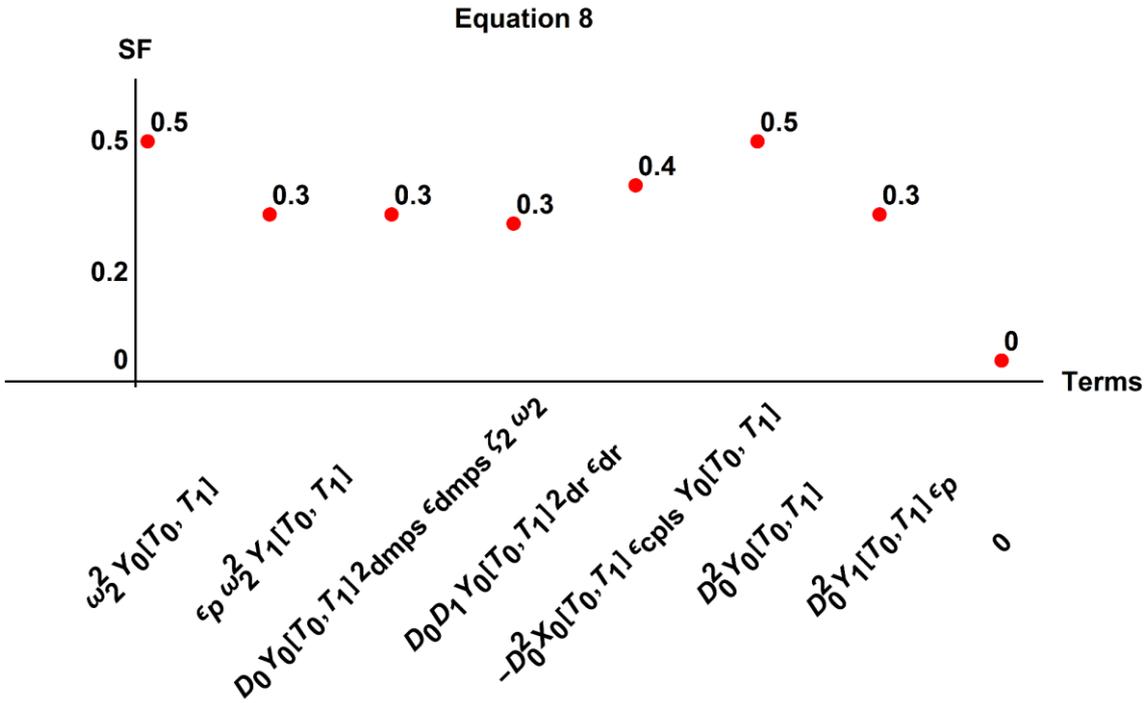


Figure 6-33 The Strength Factor configuration for eqn (8).

The *modified* response after removing  $D_0^2 Y_1 \varepsilon_p$  and  $\varepsilon_p \omega_2^2 Y_1$  are shown in Figure 6-34 and Figure 6-35. As it is shown removing these terms had no particular effect on the response of therefore they can be neglected. Although as it is shown in Figure 6-36, removing  $2_{dr} D_0 D_1 Y_0 \varepsilon_{dr}$  considerably affects the response for the secondary beam, and it should not be neglected.

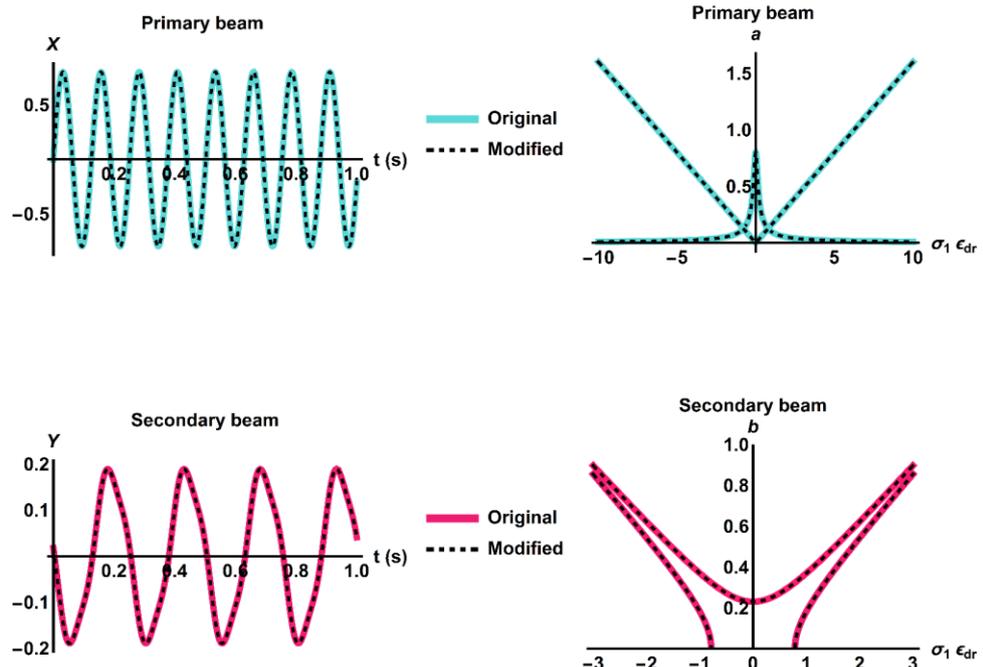


Figure 6-34 The *original* and *modified* responses after removing  $\epsilon_p \omega_2^2 Y_1$ .

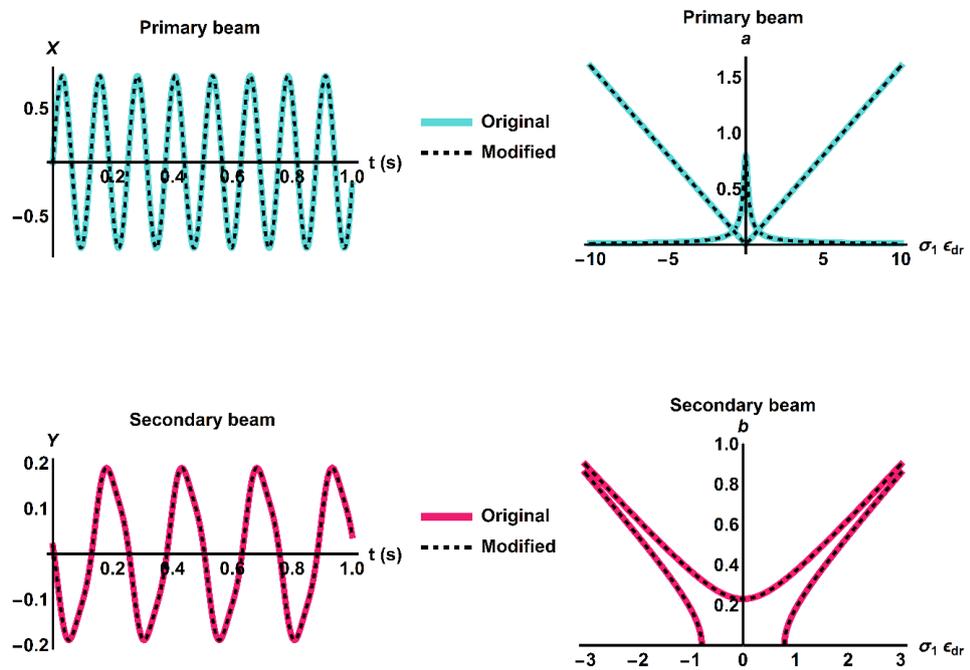


Figure 6-35 The *original* and *modified* responses after removing  $D_0^2 Y_1 \epsilon_p$ .

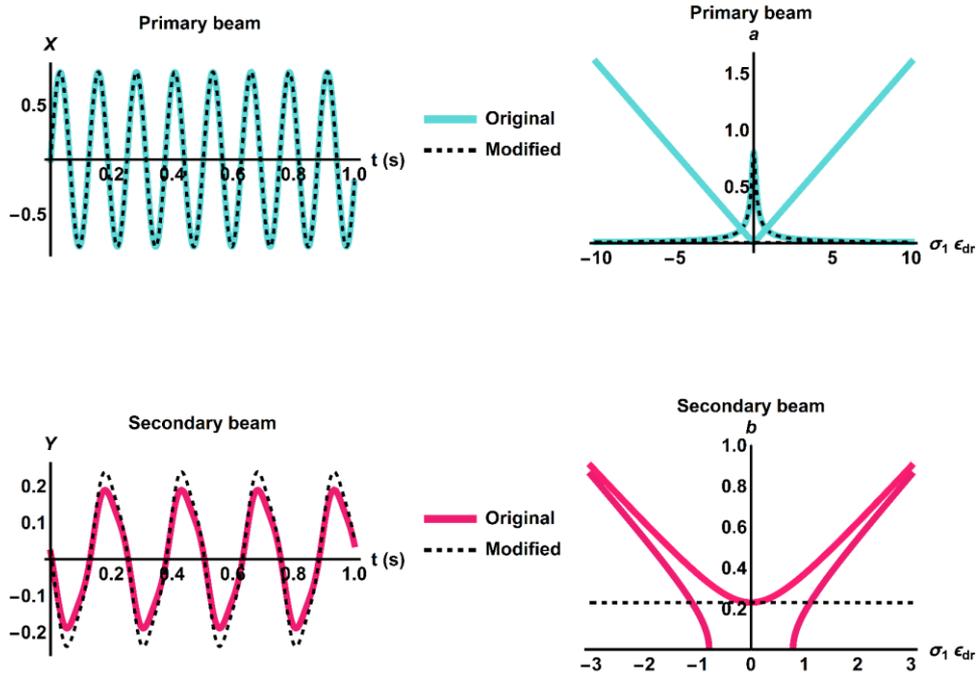


Figure 6-36 The *original* and *modified* responses after removing  $2_{dr}D_0D_1Y_0\epsilon_{dr}$ .

The SF graph for eqn (11) is given in Figure 6-37. The SF value for all the terms are less than the selected threshold, therefore terms with the terms with the maximum SF number are retained.

$D_0^2X_1$  and  $\omega_1^2X_1[T_0, T_1]$ ,  $\dot{X}[\tau] * \dot{X}[\tau]\gamma\epsilon_{nsf}$ ,  $\frac{\mu D_0^2 Y_0 \epsilon_{cplf} Y_0 [T_0, T_1]}{\epsilon_p}$ . It can be noted that the fundamental terms are also selected and retained in the equation structure; notably,  $\frac{D_0^2 \mu Y_0^2 \epsilon_{cplf}}{\epsilon_p}$ ,  $\frac{q \cos[T_0 \Omega] \epsilon_{ex}}{\epsilon_p}$ , and  $\frac{2_{dmpf} D_0 X_0 \epsilon_{dmpf} \zeta_1 \omega_1}{\epsilon_p}$ . Consequently  $\frac{2_{dr} D_0 D_1 X_0 \epsilon_{dr}}{\epsilon_p}$  is suggested as a negligible term.

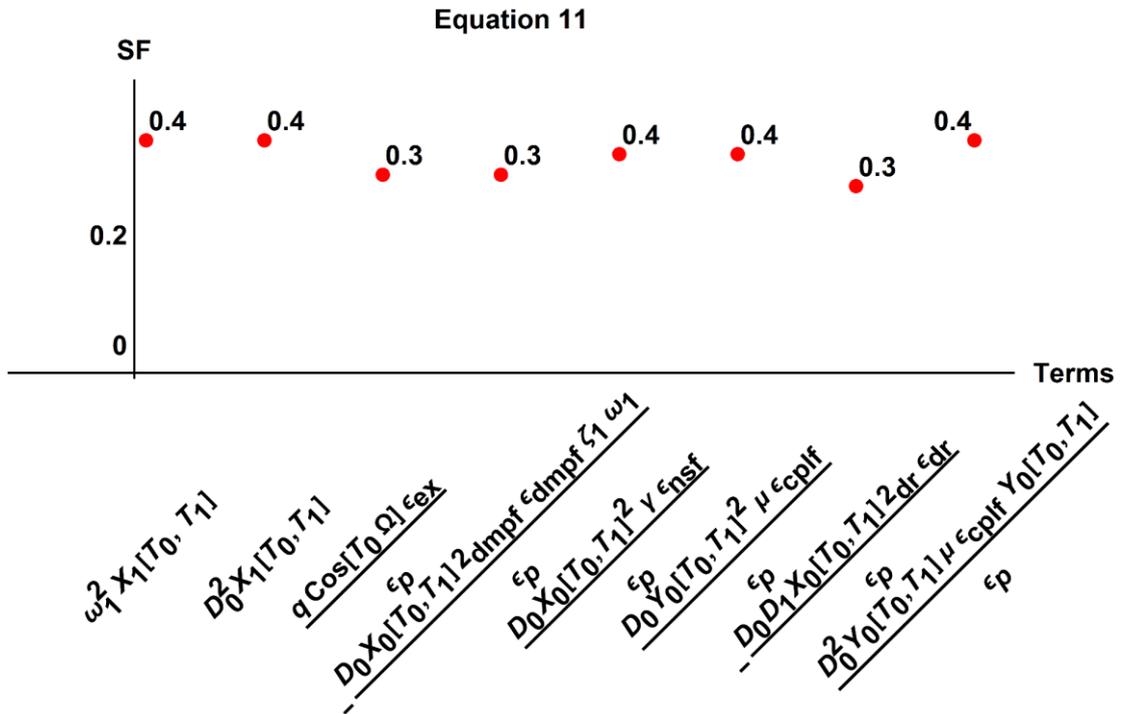


Figure 6-37 The Strength Factor graph for eqn (11).

The *modified* response after removing  $\frac{2 \text{dr} D_0 D_1 X_0 \epsilon \text{dr}}{\epsilon_p}$  from eqn (11) is calculated and compared to the *original* response, the results are shown in Figure 6-38. Removing this term has a significant effect on the response of the secondary beam, as the result this term should not be removed from this equation structure.

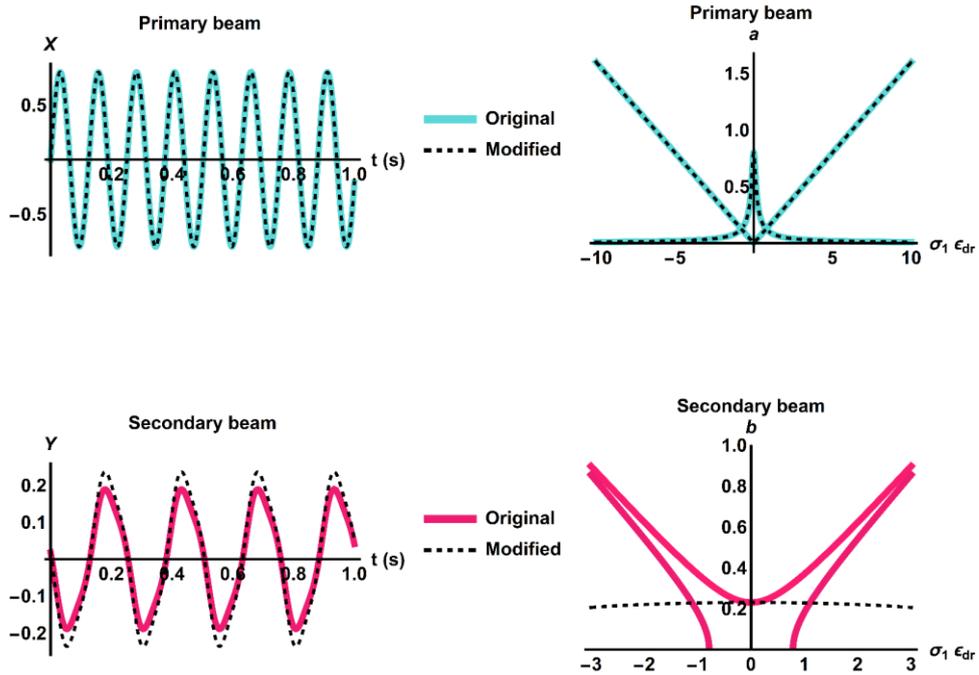


Figure 6-38 The *original* and *modified* responses after removing  $\frac{2_{dr}D_0D_1X_0\epsilon_{dr}}{\epsilon_p}$ .

The SF graph for eqn (12) is shown in Figure 6-39. The SF values for  $D_0^2Y_1$ ,  $\omega_2^2Y_1[T_0, T_1]$ , and  $\frac{D_0^2X_0\epsilon_{cps}Y_0[T_0, T_1]}{\epsilon_p}$  are higher than the rest of the terms and should be retained. Damping term  $\frac{D_0 2_{dmps}Y_0\epsilon_{dmps}\zeta_2\omega_2}{\epsilon_p}$  is categorised in the fundamental terms and also kept within the equation structure. Consequently, the *modified* response of the system without  $\frac{2_{dr}D_0D_1Y_0\epsilon_{dr}}{\epsilon_p}$  is considered.

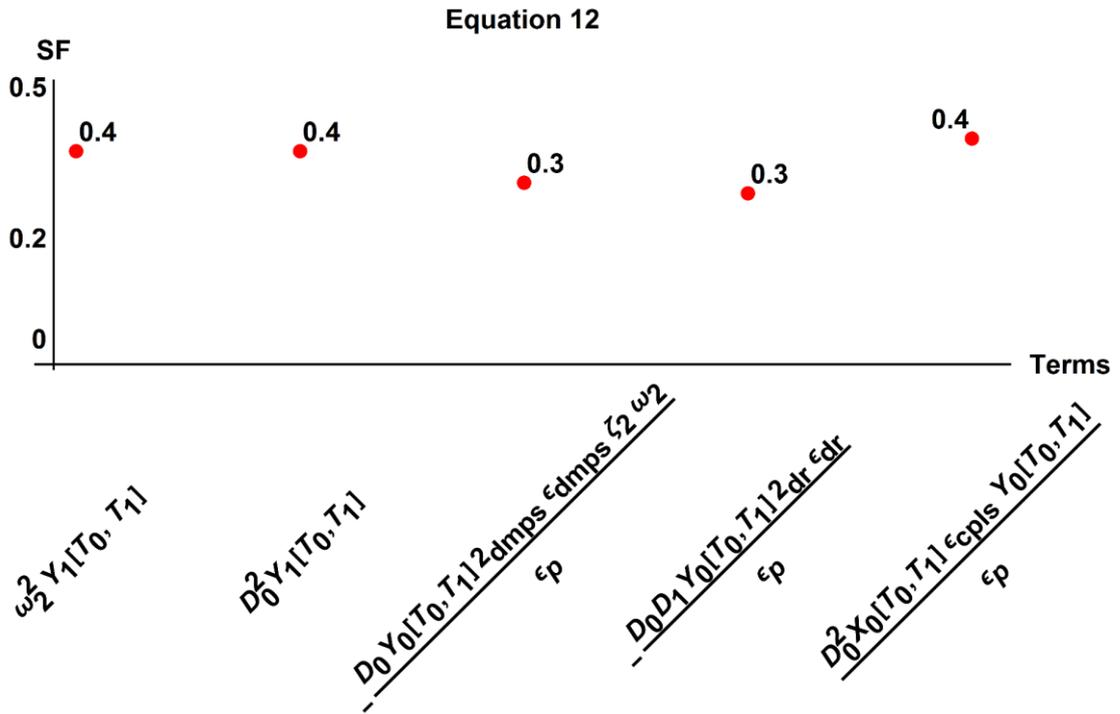


Figure 6-39 The Strength Factor configuration for eqn (12).

Figure 6-40 is a comparison between the the *original* response and the *modified* response, after removing  $\frac{2 d_r D_0 D_1 Y_0 \epsilon d_r}{\epsilon_p}$ . Removing this term has a huge effect on the response of the secondary beam, and this term should stay within the equation structure.

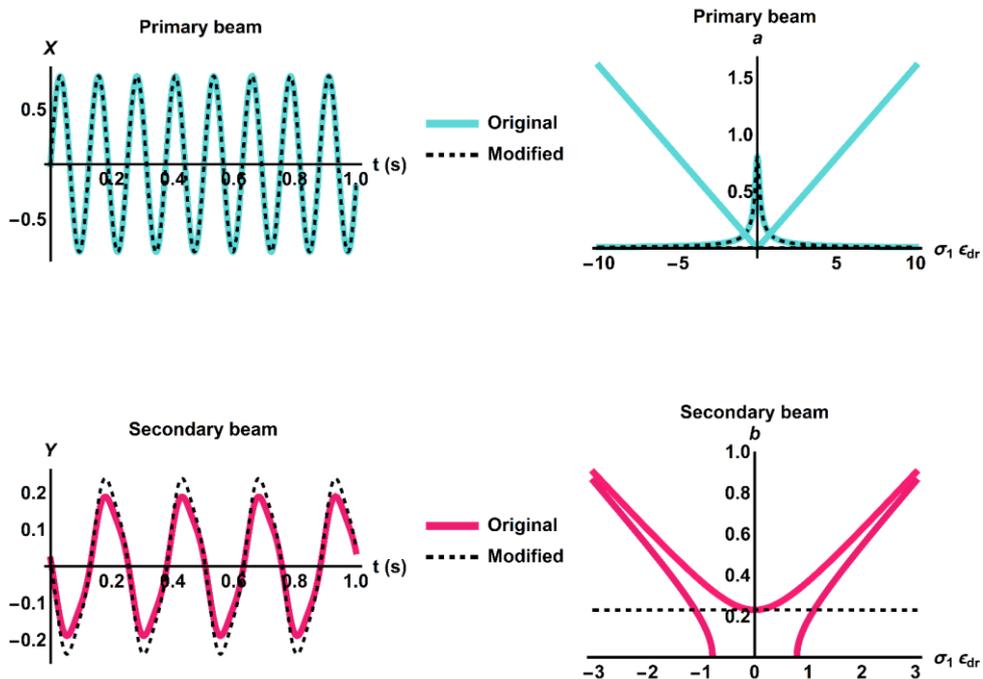


Figure 6-40 The *original* and *modified* responses after removing  $\frac{2drD_0D_1Y_0\epsilon dr}{\epsilon_p}$ .

Figure 6-41 shows the SF number graph for eqn (20). The SF values for  $\frac{e^{iT_0\Omega - iT_0\omega_1} q \epsilon_{ex}}{2\epsilon_x \epsilon_p}$  and

$\frac{2B^2 e^{-iT_0\omega_1 + 2iT_0\omega_2} \mu \epsilon_{cplf} \omega_2^2}{\epsilon_p}$  are higher than the threshold, and these terms are retained in eqn (20).

As eqn (20) is not a perturbation equation, fundamental terms can be considered negligible (the third rule in the SF guideline). Consequently, the SF guideline suggests removing  $\frac{iD_1A_2 dr \epsilon dr \omega_1}{\epsilon_p}$

and  $\frac{iA_2 \text{dmpf} \epsilon \text{dmpf} \zeta_1 \omega_1^2}{\epsilon_p}$  terms from this equation.

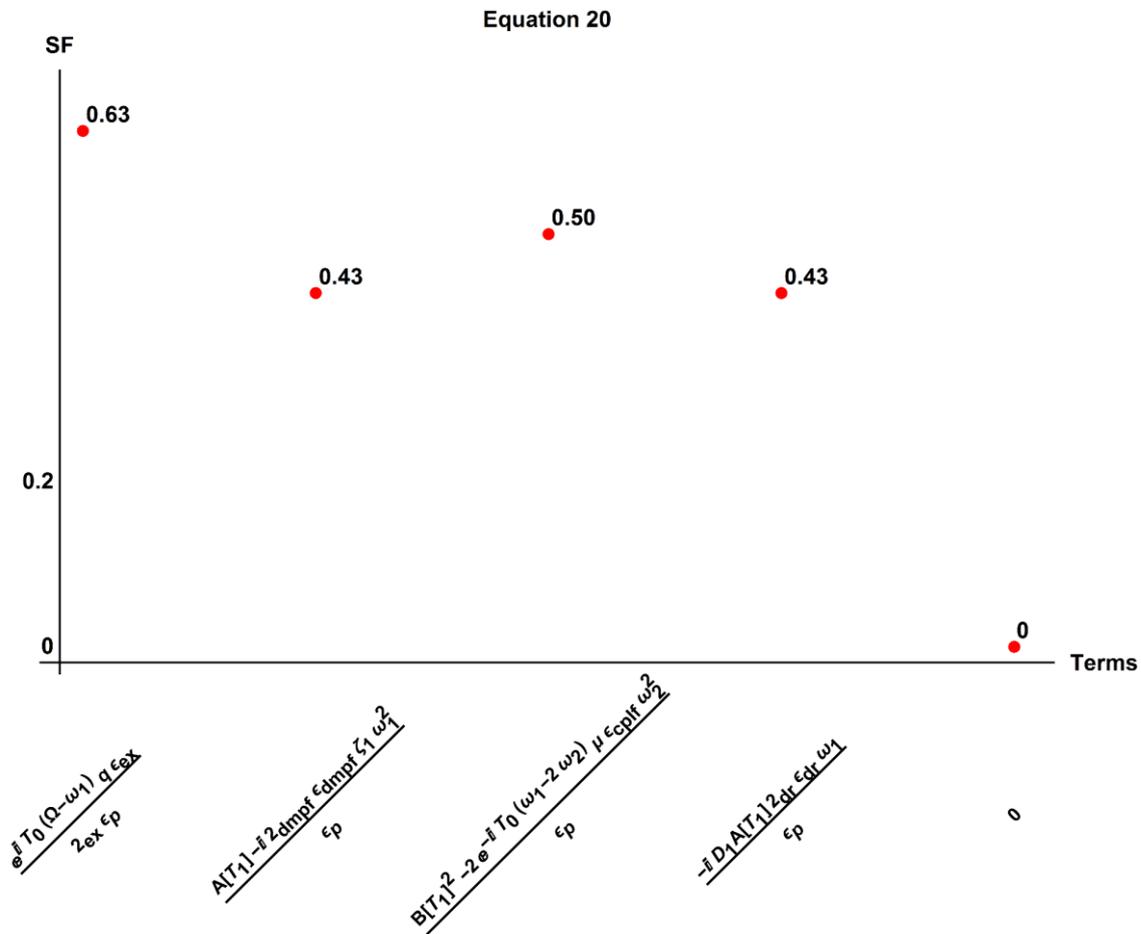


Figure 6-41 The Strength Factor configuration of eqn (20).

Figure 6-42 shows the *modified* response, after removing  $\frac{iA_2 \text{dmpf} \epsilon \text{dmpf} \zeta_1 \omega_1^2}{\epsilon_p}$  from eqn (20),

comparing to the *original* response. Both responses are identical and it is possible to remove

$\frac{iA_2 \text{dmpf} \epsilon \text{dmpf} \zeta_1 \omega_1^2}{\epsilon_p}$  from this equation.

The *modified* response after removing  $\frac{iD_1 A_2 \dot{\epsilon}_{dr} \epsilon_{dr} \omega_1}{\epsilon_p}$  is compared to the *original* response. As Figure 6-43 indicates, removing this term significantly affects the secondary beam response. As the results this term should stay in eqn (20) structure.

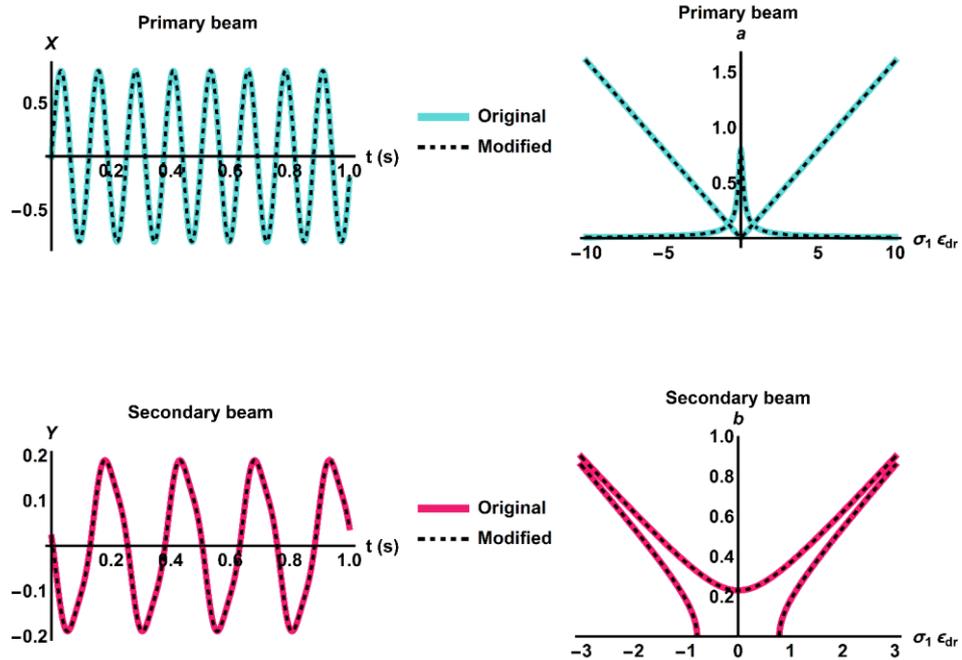


Figure 6-42 The *original* and *modified* responses after removing  $\frac{iA_2 \text{dmp} f \epsilon_{\text{dmp} f} \zeta_1 \omega_1^2}{\epsilon_p}$ .

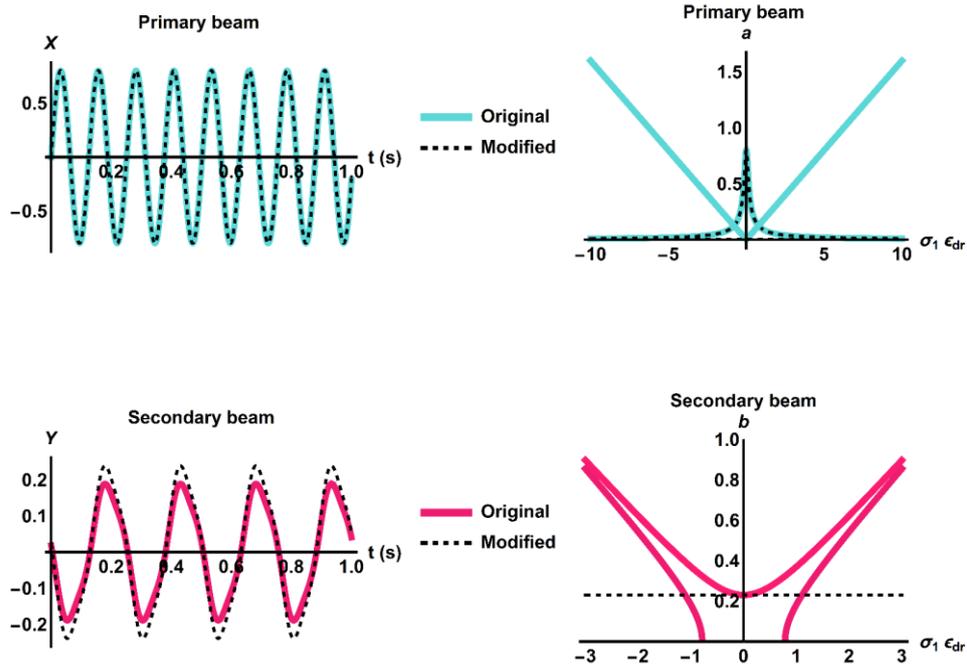


Figure 6-43 The *original* and *modified* responses after removing  $\frac{iD_1 A_2 d_r \epsilon_{dr} \omega_1}{\epsilon_p}$ .

Figure 6-44 shows the SF graph for eqn (22). The SF value of  $\frac{AB\bar{e}^{iT_0(\omega_1-2\omega_2)}\epsilon_{cpls}\omega_1^2}{\epsilon_p}$  is higher than the threshold, and therefore  $\frac{iB2dmps\epsilon_{dmps}\zeta_2\omega_2^2}{\epsilon_p}$  and  $\frac{iD_1 B_2 d_r \epsilon_{dr} \omega_2}{\epsilon_p}$  are selected as negligible terms in this equation.

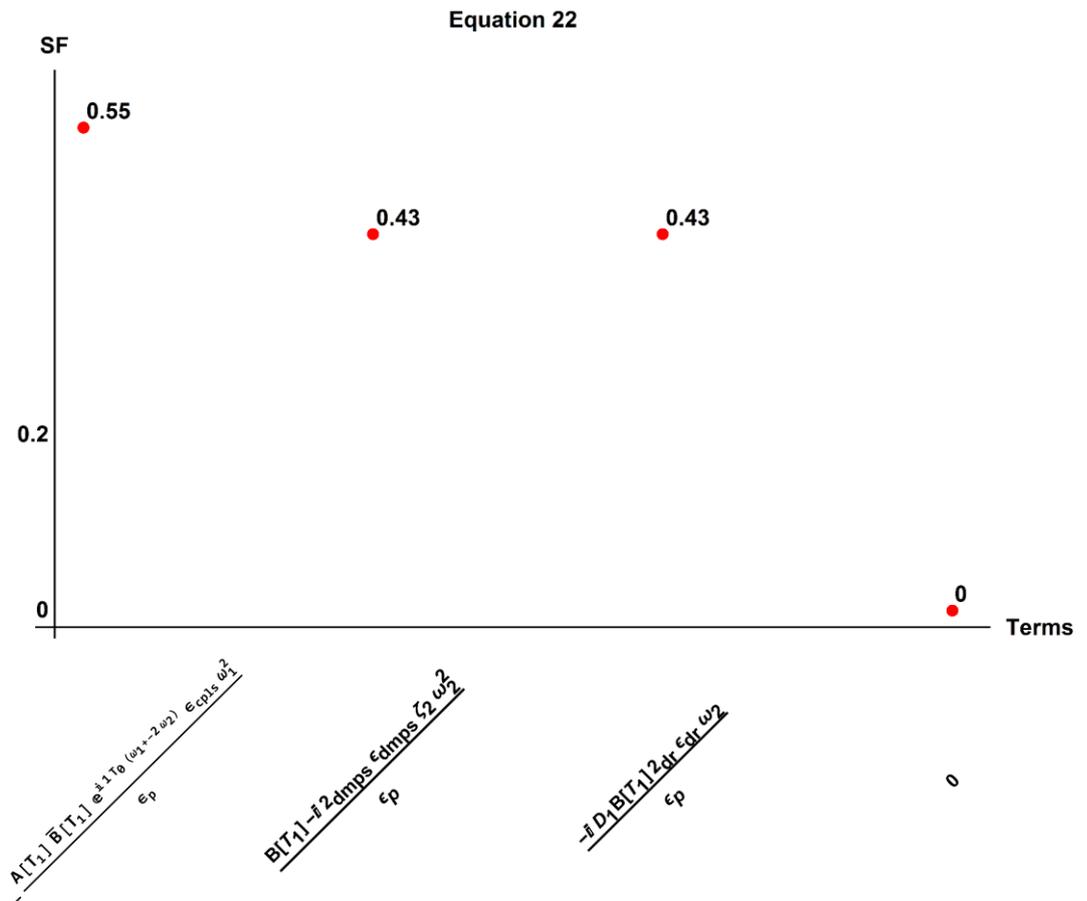


Figure 6-44 The Strength Factor configuration for eqn (22).

Figure 6-45 shows the *modified* response after removing  $\frac{iB2_{dmps}\epsilon_{dmps}\zeta_2\omega_2^2}{\epsilon_p}$  from eqn (22) compared to the *original* response. Both responses are identical and it is possible to remove this term from eqn (22) structure.

The *modified* response after removing  $\frac{iD_1B2_{dr}\epsilon_{dr}\omega_2}{\epsilon_p}$  is compared to the *original* response. The result is shown in Figure 6-46, removing this term affects the secondary beam response significantly. Consequently, this term should retain in eqn (22) structure.

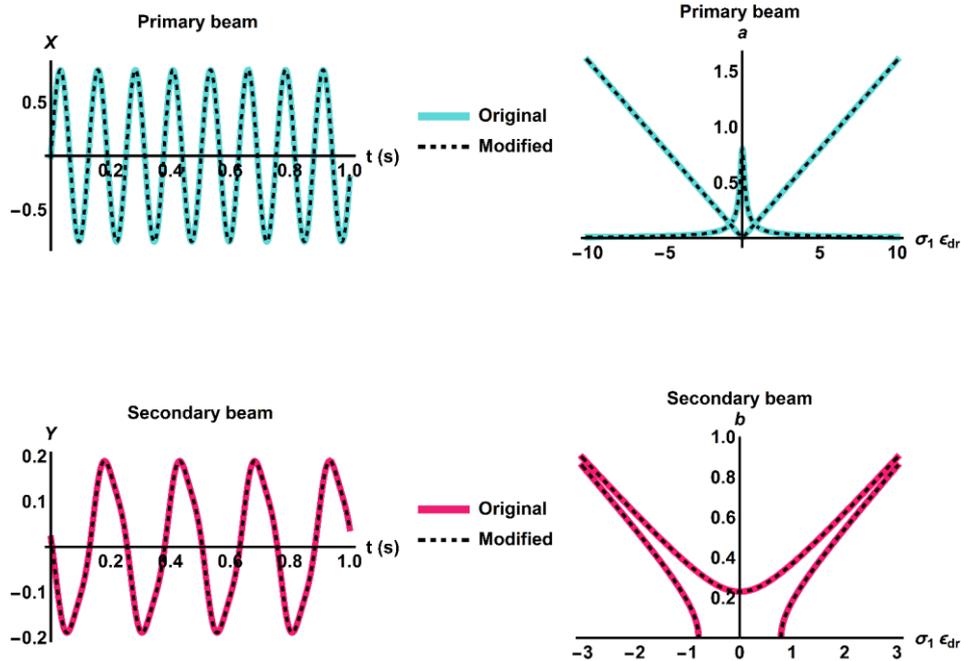


Figure 6-45 The *original* and *modified* responses after removing  $\frac{iB^2 \zeta_2 \omega_2^2}{\epsilon_p}$ .

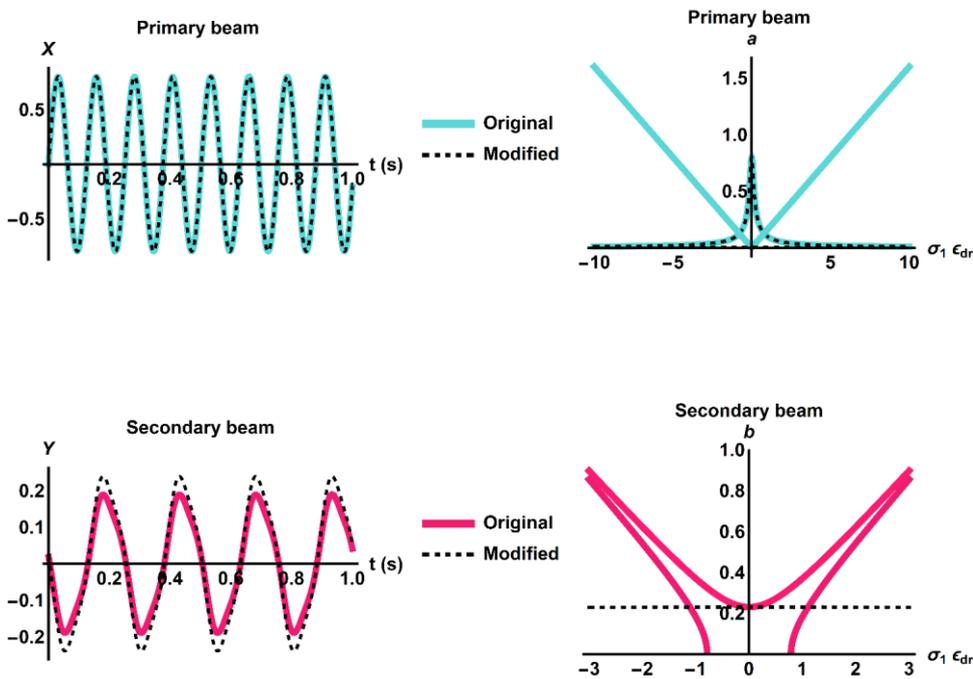


Figure 6-46 The *original* and *modified* responses after removing  $\frac{iD_1 B^2 \epsilon_{dr} \omega_2}{\epsilon_p}$ .

Figure 6-47 shows the SF graph for eqn (24). The SF values for both  $\omega_1^2 X_1$ ,  $-\frac{\bar{A}^2 e^{-2iT_0 \omega_1 \gamma \epsilon_{nsf}} \omega_1^2}{\epsilon_p}$ , and  $-\frac{A^2 e^{2iT_0 \omega_1 \gamma \epsilon_{nsf}} \omega_1^2}{\epsilon_p}$  are higher than the threshold and must retain in this equation. Also  $D_0^2 X_1$  is a fundamental term and should stay in the equation structure. Consequently, as Figure 6-48 shows  $\frac{2A\bar{A}\gamma \epsilon_{nsf} \omega_1^2}{\epsilon_p}$  can be considered negligible.

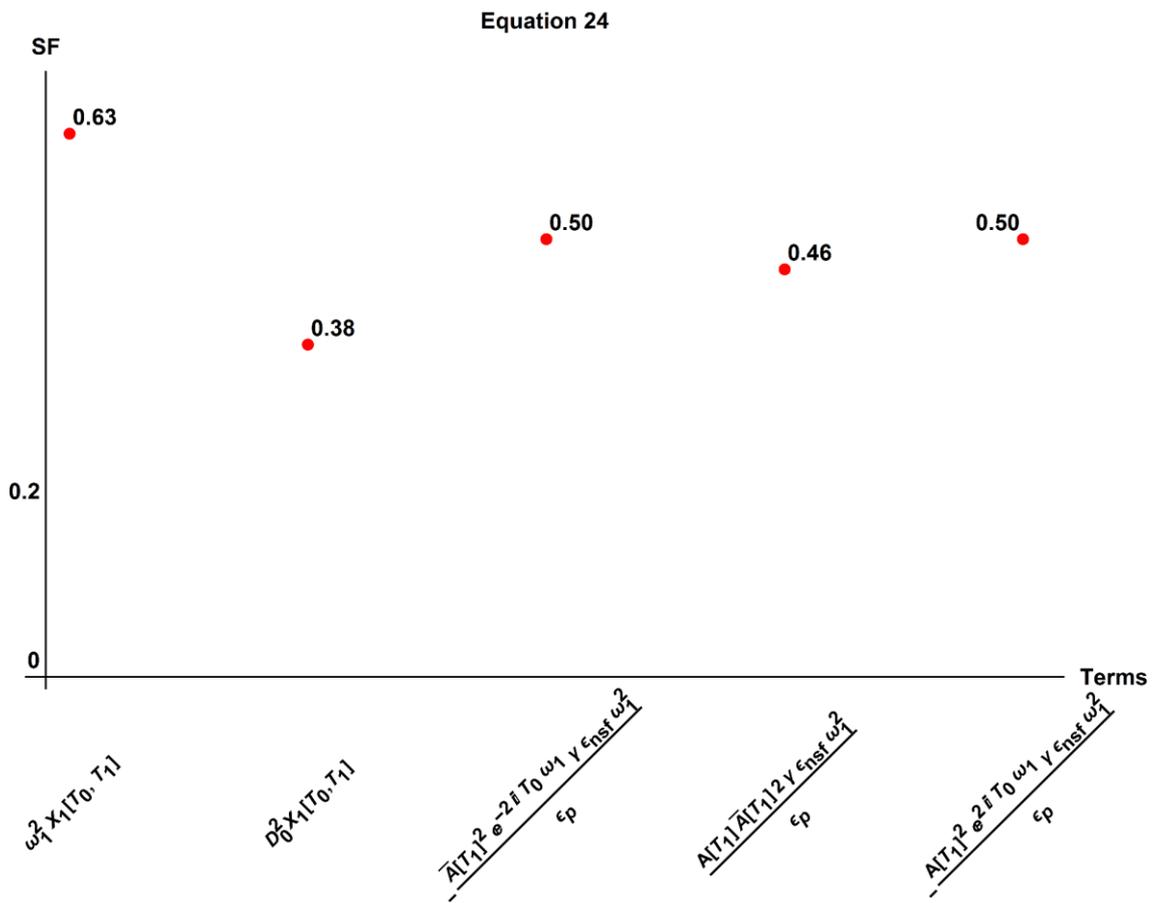


Figure 6-47 The Strength Factor configuration for eqn (24).

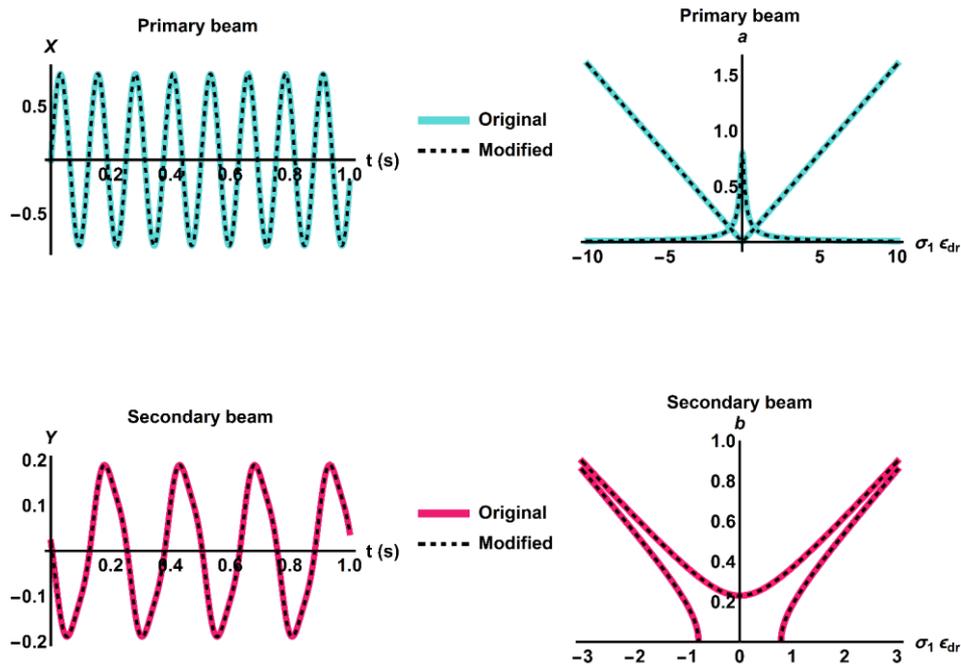


Figure 6-48 The *original* and *modified* responses after removing  $\frac{2A\bar{A}\gamma\epsilon_{nsf}\omega_1^2}{\epsilon_p}$ .

Table 6-6 summarises the results of the SF selections. Considering the current SF guideline, 15 terms were removed from 8 selected equations. The SF choice was valid for about 60% of these selections, and the response of the system after removing these terms remained unchanged. Adding a new policy to the guideline could decrease the invalid selections from 40% to 0, noting that all the invalid selected terms were seen to contain  $\epsilon_{dr}$ .

Table 6-6 Summarising the Strength Factor term selection for a case study.

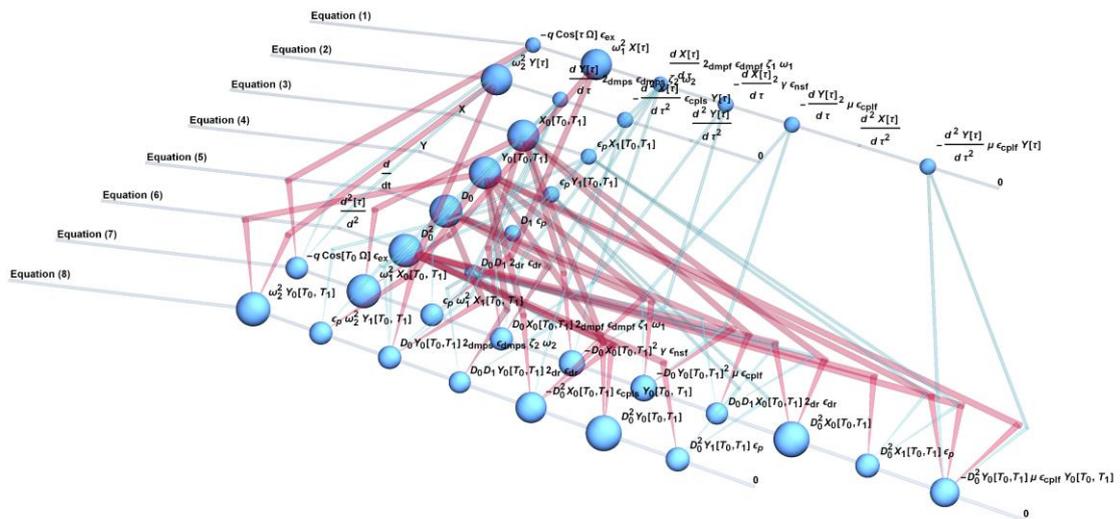
Equation number	Initial SF guideline			Revised SF guideline		
	Number of Selected terms	Number of valid selections	Number of invalid selections	Number of Selected terms	Number of valid selections	Number of invalid selections
1	1	1	1	1	1	0
7	4	4	3	3	3	0
8	3	3	2	2	2	0
11	1	1	0	1	1	0
12	1	1	0	0	0	0
20	2	2	1	1	1	0
22	2	2	1	1	1	0
24	1	1	1	3	3	0
	-	60%	40%	-	100%	0%

It is important to notice that these results are just for one case study and so the proposal for using the SF number to identify negligible terms requires further investigations before it can be considered as a generalisable rule-base.

### 6.6 A three-dimensional structural analogy for the solution procedure

The relationships between the solution method and the SEEM term-tracking method for various information aspects were investigated in this research. It is also potentially interesting to take this concept to another level and examine the relationship between the terms in an analysis regardless of the equation structure and the solution method hierarchy.

In the original version of the *Blueprint* visualisation graph, the solution hierarchy is illustrated by defining a special set of coordinates (x,y,z), for each equation within the visualisation pallet. For example, when considering Figure 6-49 the first nine equations for the problem of auto-parametric excitation of a coupled beam system can be seen. The solution procedure hierarchy, the equation structures, and term positions are clearly identifiable in this graph.



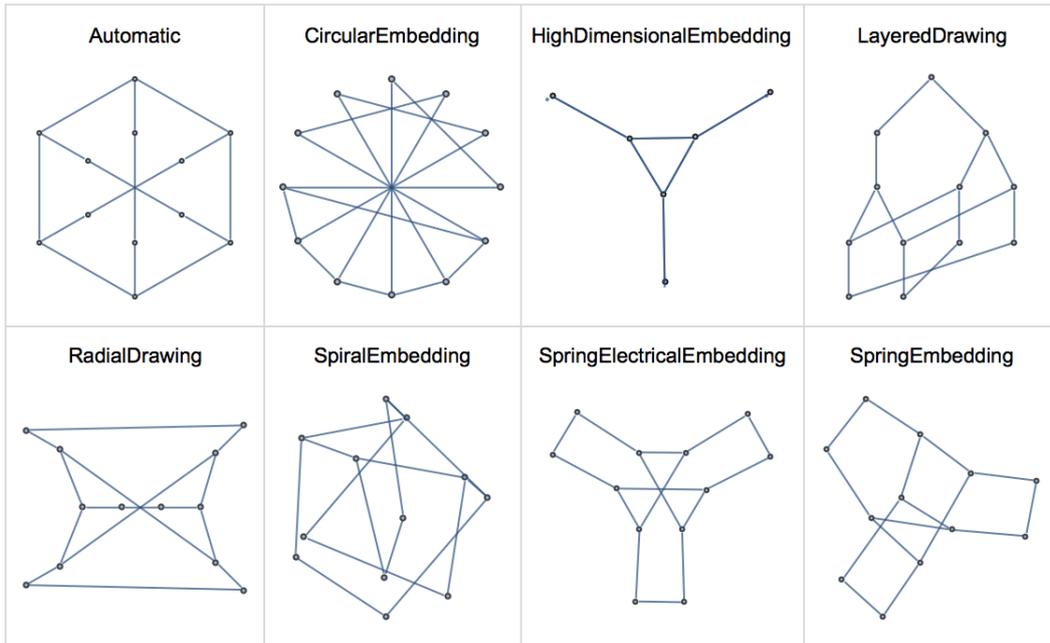


Figure 6-50 The 3D graph positioning algorithms in *Mathematica*

The result of removing the defined positioning and plotting based on the *HighDimensionalEmbedding* algorithm is illustrated in Figure 6-51. Interestingly in this sort of visualisation the key terms are quickly identifiable because a larger number of links is sourced to them. Furthermore, the SEEM encodings and the SF values can be clearly shown in this format.

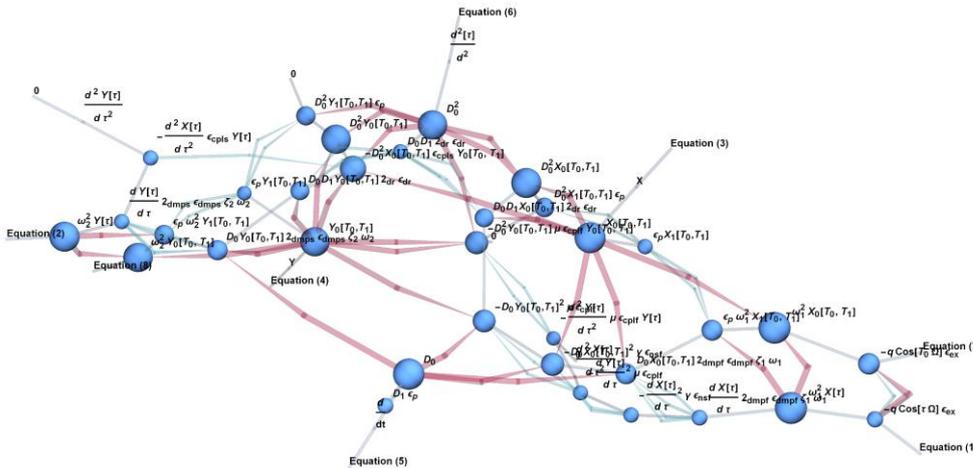


Figure 6-51 The modified Blueprint visualisation graph for the first nine equations in the autparametrically resonant coupled beam system.

It is theoretically possible to evaluate this form of plot for the complete solution procedure for any conceivable problem. Figure 6-52 shows the complete solution structure of the parametrically excited pendulum problem, which was discussed in chapters 3 and 4. Figure 6-53 shows the complete solution structure for the problem of the autoparametrically resonant coupled beam system, which is partially investigated in this chapter.

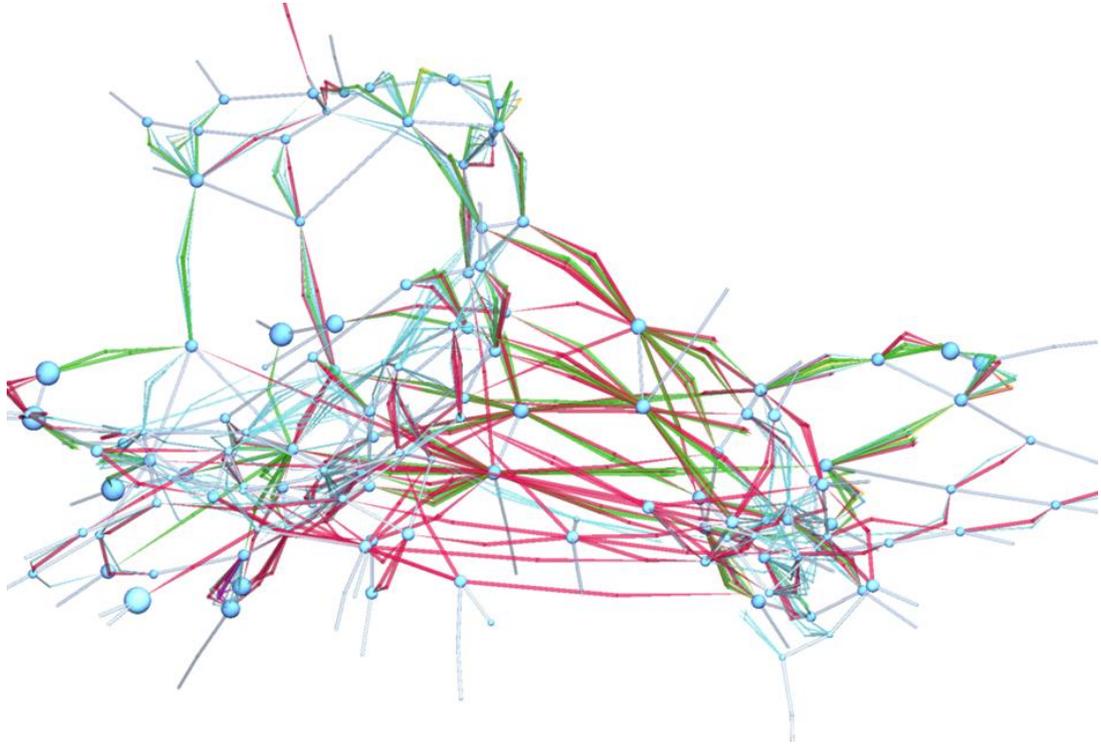


Figure 6-52 An alternative presentation of the Blueprint visualisation method for the solution of the parametrically excited pendulum problem.

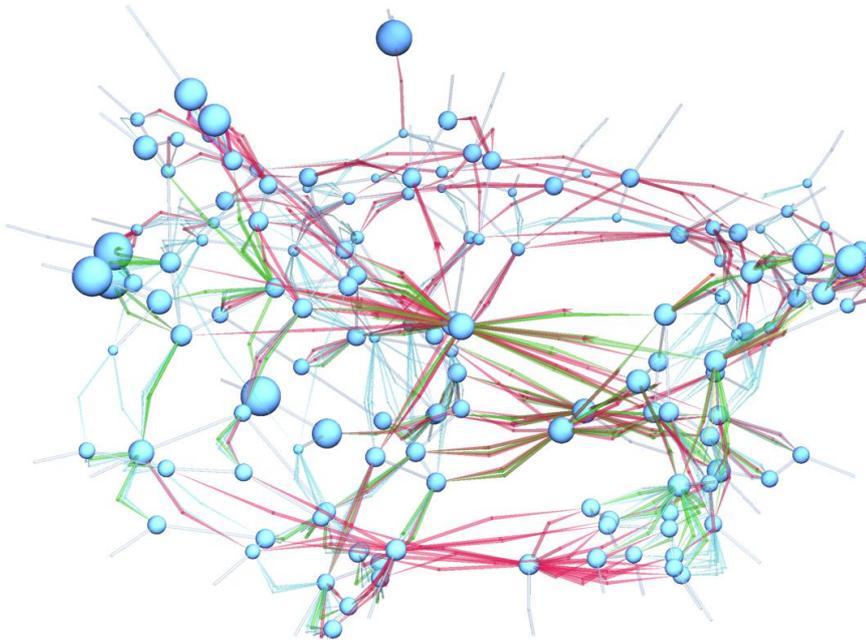


Figure 6-53 An alternative presentation of the Blueprint visualisation method for autoperometrically resonant coupled beam system.

This an alternative representation for analytical solution procedures and starts to suggest the possibility of manufacturing a solid 3D printed structure for each solution, as shown in Figure 6-54 and Figure 6-55. This offers a completely new perspective for experiencing the analytical solution procedures that have been studied, in the sense of generating an actual physical structure for a complex mathematical solution procedure.

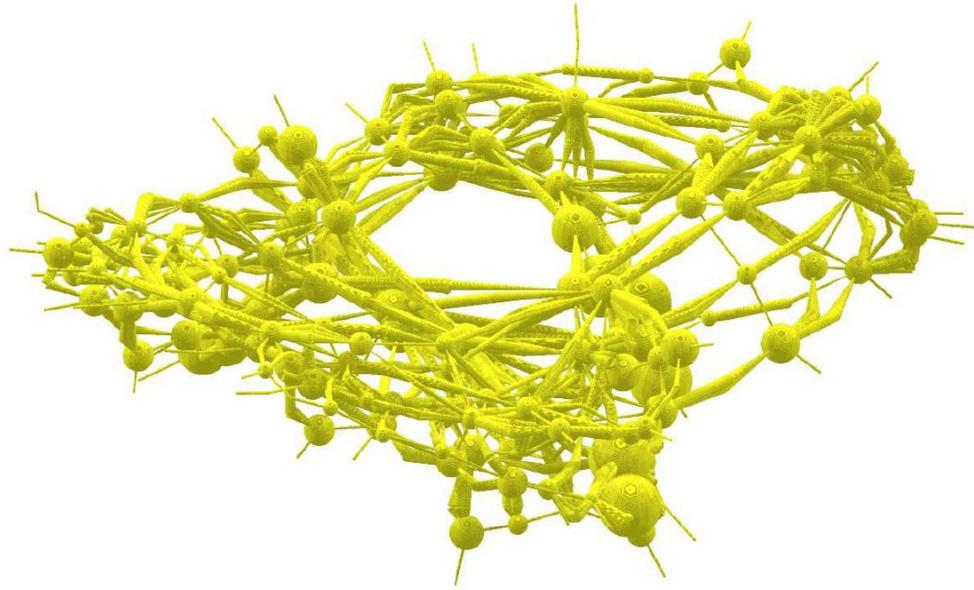


Figure 6-54 A conjectured 3D printed sample of a solution procedure for a parametrically excited pendulum problem.

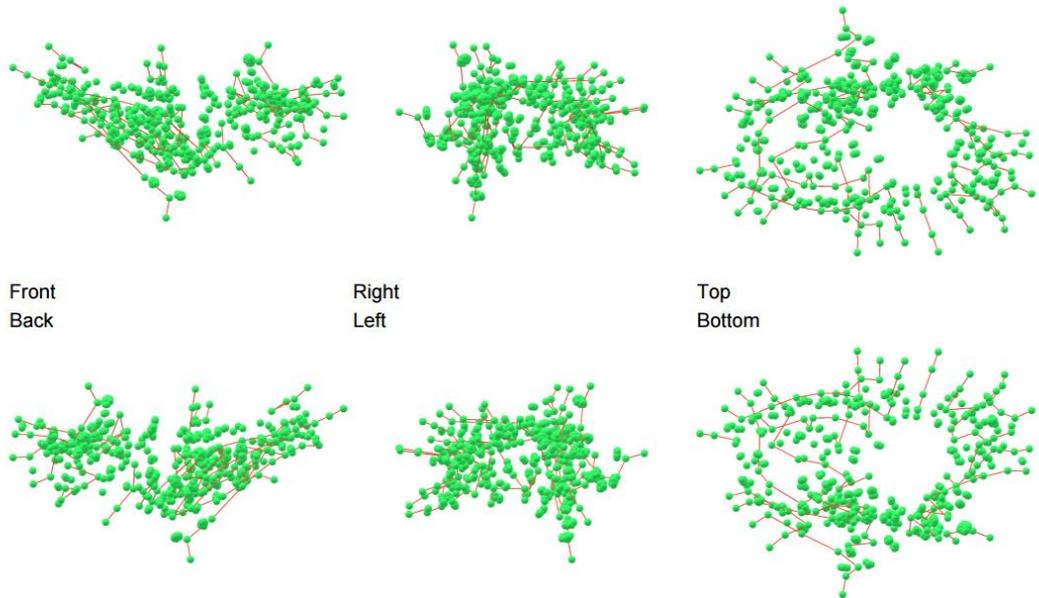


Figure 6-55 Different angle views of a conjectured 3D printed sample of a solution procedure for a parametrically excited pendulum problem.

An interesting observation can be made from investigation of the top and bottom views of the solution visualisations for both case studies.

The top view of the structure generated for the autparametrically resonant coupled beam system is shown in Figure 6-56. This Figure shows a reasonably symmetrical structure containing two cylindrical shapes that are connected around an axis of symmetry (highlighted in red). For convenience these cylindrical shapes are denoted by A and B.

The SEEM encoding information for each link can be visualised using the *Tooltip* function. Can be seen that most of the terms in A relate to the primary beam, whilst the majority of the terms in B are connected with the secondary beam. The coupling terms mostly appear near the axis of symmetry.

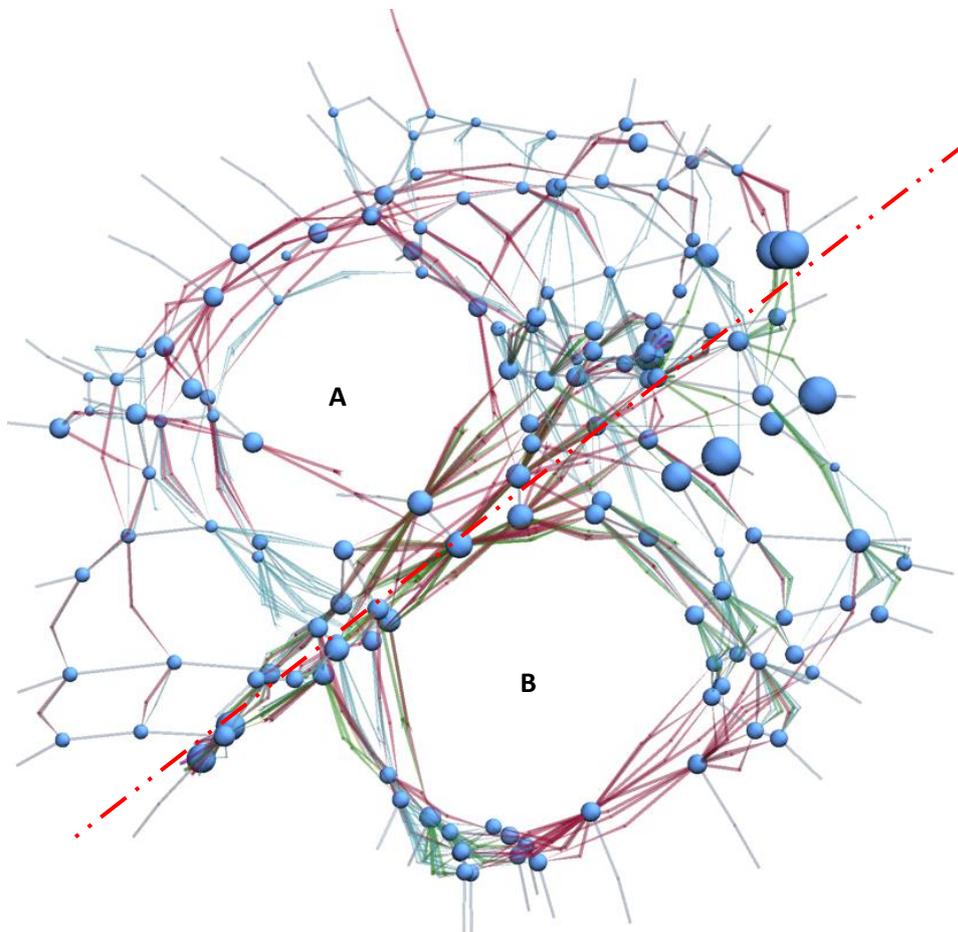


Figure 6-56 Plan view of the *Blueprint* visualisation for the autparametrically resonant coupled beam system, showing the physical sub-systems of the problem in the form of cylindrical regions A and B.

Considering the systematic nature of the perturbation method of multiple scales and the fact that almost the same computational process is repeated for each coordinate, in terms of defining the perturbation equations, defining the secular terms, the solvability conditions, and then deriving the modulation equations one can see where a regular repeated graphical structure could emerge from and that it could offer a basis for the computational sub-space associated with each generalised coordinate. Obviously in such cases the coupling terms would define the symmetrical intersection points of the overall solution structure.

The same observation for the parametrically excited pendulum can be observed. A cylindrical form can be identified in both the top and bottom views of Figure 6-55. There is only one generalised coordinate in this problem, therefore only one cylindrical shape can be identified.

## 6.7 Summary

Considerable developments for the SCD solvers have been accomplished in this chapter. These developments are summarised as follows:

- The Strength Factor (SF) concept has been introduced and implemented into the *Blueprint* visualisation method.
- A guideline for selecting potentially negligible terms in each equation is provided.
- To highlight the effectiveness of the developed Symbolic Computational Dynamic (SCD) solver, part of the problem of the autoparametrically excited coupled beam system was investigated.
- The automated SEEM encoding information, and the visualisation links for most of the terms have been investigated in detail.
- The concept of the SF guideline was applied to the autoparametrically coupled beam system problem.
- A 3D structure for the solution procedure based on the *Blueprint* method was introduced.
- In the 3D structure of each case study a cylindrical like shape per each generalised coordinate was observed.



# Chapter 7 Conclusions and Future Work

In this research a new link between the defining mathematical equations and the physics of problems in engineering dynamics has been created, principally by developing the concept and functionality of novel Symbolic Computational Dynamics (SCD) solvers. The Source and Evolution Encoding Method (SEEM) term-tracker has been discussed and computerised by means of advanced code written in the *Mathematica*<sup>TM</sup> programming language. The *Blueprint* visualisation method for the SCD solver has been developed in some considerable detail and computerised, again by means of code written in *Mathematica*<sup>TM</sup>. A metric for term evaluation in the form of the Strength Factor (SF) has been introduced. An adjunct to this is an alternative visualisation construct in the form of a three-dimensional printed solid structure which has also been introduced in this thesis. After discussing the conclusions and proposing some future work for this research area, this thesis ends with a closing note.

## 7.1 The Source and Evolution Encoding Method

The SEEM encoding method has been developed and discussed in some detail in this thesis. This term-tracking method extracts, saves, and pre-processes a considerable amount of information during the solution procedure. Several functions for applying and post-processing this information have also been developed, for visualisation. The application of this method has been demonstrated by applying the SEEM to the problem of the parametrically excited pendulum. The SEEM has been shown to identify valuable information at each stage of this analysis, and two uses for this information are identification of the sources of each term and in making a connection back to the physical conceptualisation of the problem.

## 7.2 The *Blueprint* visualisation method

The *Blueprint* visualisation method has been introduced and implemented into the SCD solver. This method is able to illustrate the process of development of the encoding information during the solution procedure. A flexible visualisation panel has been programmed, enabling the user to interact with the generated results. Then the application of this method has been demonstrated by discussing the generated graphs for the problem of the parametrically excited pendulum.

This idea has been implemented into the SCD solver and the generated graphs for partial analysis of the problem of a parametrically excited pendulum case study have been discussed.

In future it is suggested that adding features such as highlighting the path taken for a selected quantity in the solution procedure structure, and changing the texture of the links for colour-blind users are suggested as potentially important avenues for future development work.

## 7.3 A symbolic term evaluation method

The Strength Factor (SF) is a new metric based on the SEEM term-tracking information, and the solution procedure has been extended to use this to highlight the significance of each term in the equations. This feature has been implemented within the *Blueprint* visualisation method. A guideline for identifying less effective terms in each equation has subsequently been established. Part of the solution procedure for the problem of the autoparametrically resonant beam system, which has been found using the SCD solver developed in this research has been discussed. The SF guideline has been applied to all equations in this particular case study, resulting in a prediction that about 60% of the selected terms could be considered to be negligible. It has been shown in some follow-on numerical calculations that removing these terms does not significantly alter the quantitative or qualitative aspects of the response of this particular system. This finding is not yet sufficiently confirmed as a generic result from this research but it appears to be very promising.

This idea requires further development, in the form of a stringent sensitivity analysis for a defined points system, leading on to an updating of the SF guideline rules by investigating more case studies.

## 7.4 Alternative visualisation for the solution procedure

A proposal for a 3D printed version of the *Blueprint* visualisation method has also been introduced in this thesis. This has been developed by cancelling the predefined fixed coordinates

for each equation and the terms within the *Blueprint* visualisation graph and shows the relationships between the terms, and this is regardless of the solution procedure hierarchy. An interesting observation emerges from the case studies investigated whereby cylindrical structures can be identified in the visualisation, and furthermore that these structures appear to relate to the generalised coordinates of the input problem. These cylindrical sub-structures are only visible from looking perpendicularly onto the plan views of the visualisation, either from the top or the bottom.

In future it will be interesting to investigate different case studies for higher dimensional problems comprising more than two degrees of freedom (therefore defined physically by more than two generalised coordinates, and hence more than two coupled nonlinear differential equations of motion) in order to understand fully, and hence validate, this observation. Also structures emanating from different solution methods for the same suite of problems could usefully be compared to the same ends.

The application of this method can be extended to recent virtual reality technology, where it provides the user with a new form of interaction with the analytical solution procedure. As well as the professional users, this visualisation method can be used in the education sector.

## **7.5 Closing note**

As has been discussed in the motivation chapter of this thesis, a new perspective for looking at the information generated by approximate analytical solution procedures in engineering dynamics has been developed. Before a differential equation such as that shown in Figure 1-3, could have been considered to be just a differential equation. Development of the SCD approach has shown a considerable amount of implicit information hidden in such equation structures, and this information may be definable and applied to useful ends by the user. As Figure 7-1 illustrates, different forms of information can be extracted using the SCD solver technology developed in the research reported in this thesis.

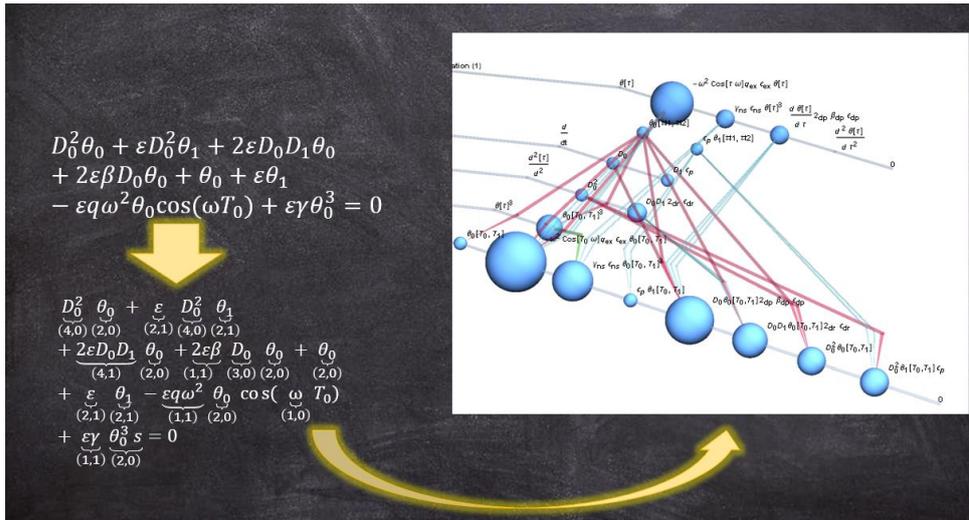


Figure 7-1 This is not just a differential equation.

# List of Publications

- [1] Motazedi N., Cartmell M. P, and Rongong J. A. 2017. *Extending the functionality of a symbolic computational dynamic solver by using a novel term-tracking method*. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science [*under review*]
- [2] Motazedi N., Cartmell M. P, and Rongong J. A. 2016. *Further developments of symbolic computational dynamics solvers through the introduction and implementation of a new technique for precise term-tracking and identification*. International Design Engineering Technical Conferences & Computers & Information in Engineering Conference. Charlotte, NC, USA: ASME.
- [3] Motazedi N., Cartmell M. P, and Rongong J. A. 2016. *The development on symbolic computational dynamics solvers*. 6th European Conference on Structural Control. Sheffield.



# References

1. Khanin, R. and M.P. Cartmell. *Parallelisation of perturbation analysis. in application of the multiple scales method to large scale problems in engineering.* 1999.
2. Khanin, R., M. Cartmell, and A. Gilbert, *A Computerized implementation of the multiple scales perturbation method using Mathematica.* Computers and Structures, 2000. **76**(5): p. 565-575.
3. Cartmell, M., D. Forehand, and R. Khanin, *Symbolic Computational Modelling and Solution of Problems in Applied Dynamics.* 2002.
4. Cartmell, M.P. and D.J. McKenzie, *A review of space tether research.* Progress in Aerospace Sciences, 2008. **44**(1): p. 1-21.
5. Cartmell, M.P., et al., *Multiple scales analyses of the dynamics of weakly nonlinear mechanical systems.* Applied Mechanics Reviews, 2003. **56**(5): p. 455-491.
6. Forehand, D.I.M., et al. *Analytical model of a three dimensional box structure undergoing combined translational and rotational motions.* 2000.
7. Forehand, D.I.M. and M.P. Cartmell, *The implementation of an automated method for solution term-tracking as a basis for symbolic computational dynamics.* Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2011. **225**(1): p. 40-49.
8. Forehand, D.I.M., M.P. Cartmell, and R. Khanin, *The theory and application of a classical Eulerian multibody code.* International Journal of Mechanical Engineering Education, 2005. **33**(2): p. 149-176.
9. Forehand, D.I.M., R. Khanin, and M.P. Cartmell, *A Lagrangian multibody code for deriving the symbolic state-space equations of motion for open-loop systems containing flexible beams.* Mathematics and Computers in Simulation, 2004. **67**(1-2): p. 85-98.
10. Khanin, R., *Extending the Method of Multiple Scales to Strongly Nonlinear Vibration Problems,* in *Modern Practice in Stress and Vibration Analysis: Proceedings of the 5th International Conference on Modern Practice in Stress and Vibration Analysis,* M.P. Cartmell, Editor. 2003: Glasgow, Scotland. p. 45-50.
11. Khanin, R. and M. Cartmell, *Parallelization of Perturbation Analysis: Application to Large-scale Engineering Problems.* Journal of Symbolic Computation, 2001. **31**(4): p. 461-473.
12. Khanin, R. and M. Cartmell, *Applying the Perturbation Method of Multiple-Scales.* Mathematica in Education and Research, 1999. **8**(2): p. 19-26.
13. Khanin, R., et al. *Parallel symbolic computational modelling of nonlinear dynamical problems using the method of multiple scales.* 1998.
14. St Exupery, A.d., *The little prince.* 1943: New York: Harcourt, Brace and World.
15. Thomsen, J.J., *Vibrations and stability, order and chaos.* 1997: McGraw-Hill Companies.

16. de Weck, O. and I.Y. Kim, *Finite element method*. Engineering Design and Rapid Prototyping, 2004.
17. MacNeal, R., *The NASTRAN Theoretical Manual (level 15.5): Msr-40*. 1972: MacNeal-Schwendler Corporation.
18. ADAMS, M.U., *Solver*. Driver, Mechanical Dynamics, 2000. **199711998**.
19. Simulations, C.L. *Vortex (software)*. 2001 [cited 2017; Available from: <https://www.cm-labs.com/en/vortex-studio/>].
20. Bourg, D.M. and B. Bywalec, *Physics for Game Developers: Science, math, and code for realistic effects*. 2013: " O'Reilly Media, Inc."
21. Coutinho, M.G., *Dynamic simulations of multibody systems*. 2013: Springer Science & Business Media.
22. Lanczos, C., *The variational principles of mechanics*. 2012: Courier Corporation.
23. Hutton, D., *Fundamentals of finite element analysis*. 2004: McGraw-Hill.
24. Brenner, S. and R. Scott, *The mathematical theory of finite element methods*. Vol. 15. 2007: Springer Science & Business Media.
25. Thomée, V., *Galerkin finite element methods for parabolic problems*. Vol. 1054. 1984: Springer.
26. Reddy, J.N., *An introduction to the finite element method*. Vol. 2. 1993: McGraw-Hill New York.
27. Zienkiewicz, O.C. and R.L. Taylor, *The finite element method: solid mechanics*. Vol. 2. 2000: Butterworth-heinemann.
28. Yu, T., *Counterintuitive behavior in a problem of elastic-plastic beam dynamics*. Journal of Applied Mechanics, 1985. **52**: p. 517.
29. Cook, R.D., *Finite Element Modeling for Stress Analysis*. 1995: John Wiley & Sons.
30. Smith, G.E., *The dangers of CAD*. Mechanical Engineering, 1986. **108**(2): p. 58-64.
31. O. de Weck, et al., *Engineering Design and Rapid Prototyping (IAP 2004)*. 2004, Massachusetts Institute of Technology Department of Aeronautical and Astronautical Engineering.
32. MacNeal, R., *Finite elements: Their design and performance*. 1993: CRC Press.
33. Nayfeh, A.H. and D.T. Mook, *Nonlinear Oscillations*, 1979.
34. Lakrad, F. and M. Belhaq, *Periodic solutions of strongly non-linear oscillators by the multiple scales method*. Journal of Sound and Vibration, 2002. **258**(4): p. 677-700.
35. Nayfeh, A., *Perturbation methods*. Wiley & Sons, 1973: p. 162.
36. Nayfeh, A.H. and D.T. Mook, *Nonlinear oscillations*. 1995: John Wiley & Sons.
37. Nayfeh, A.H., C.M. Chin, and J. Pratt, *Perturbation methods in nonlinear dynamics - applications to machining dynamics*. Journal of Manufacturing Science and Engineering, Transactions of the ASME, 1997. **119**(4 PART I): p. 485-493.
38. Cartmell, M.P. and D.I.M. Forehand, *On the assumptions and decisions required for reduced order modelling of engineering dynamical systems*. Proceedings of the 6th EUROMECH Nonlinear Dynamics Conference (ENOC), 2008.
39. Thomson, W.T., *Vibration theory and applications*. 1965: Prentice-Hall.
40. Macala, G.A. *SYMBOD: A COMPUTER PROGRAM FOR THE AUTOMATIC GENERATION OF SYMBOLIC EQUATIONS OF MOTION FOR SYSTEMS OF HINGE-CONNECTED RIGID BODIES*. in *AIAA 21st Aerospace Sciences Meeting*. 1983. New York, NY, USA: AIAA.
41. Schiehlen, W.O. *COMPUTER GENERATION OF EQUATIONS OF MOTION*. in *Computer Aided Analysis and Optimization of Mechanical System Dynamics (Proceedings of the NATO Advanced Study Institute)*. 1984. Berlin, West Ger: Springer-Verlag.
42. Schiehlen, W.O. *Computational dynamics of multibody systems*. 1990.

43. Lieh, J. and I.-u. Haque, *Symbolic closed-form modeling and linearization of multibody systems subject to control*. Journal of mechanisms, transmissions, and automation in design, 1991. **113**(2): p. 124-132.
44. King, K.N. and A.J. Offutt, *A fortran language system for mutation-based software testing*. Software: Practice and Experience, 1991. **21**(7): p. 685-718.
45. Chace, M.A. *METHODS AND EXPERIENCE IN COMPUTER AIDED DESIGN OF LARGE-DISPLACEMENT MECHANICAL SYSTEMS*. in *NATO ASI Series, Series F: Computer and Systems Sciences*. 1984.
46. Wehage, R.A. and E.J. Haug, *Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems*. Journal of Mechanical Design, 1982. **104**(1): p. 247-255.
47. Sheth, P.N. and J.J. Uicker Jr, *IMP (Integrated Mechanisms Program), a computer-aided design analysis system for mechanisms linkages*. ASME Journal of Engineering for Industry, 1972. **94**(2): p. 454-464.
48. Richard, M.J., R.J. Anderson, and G.C. Andrews, *The vector-network method for the modelling of mechanical systems*. Mathematics and Computers in Simulation, 1990. **31**(6): p. 565-581.
49. Wittenburg, J., *Graph-theoretical methods in multibody dynamics*. J. Contemp. Math., 1989. **97**: p. 459-468.
50. Shi, P. and J. McPhee, *Dynamics of Flexible Multibody Systems Using Virtual Work and Linear Graph Theory*. Multibody System Dynamics, 2000. **4**(4): p. 355-381.
51. Melzer, F., *Symbolic computations in flexible multibody systems*. Nonlinear Dynamics, 1996. **9**(1-2): p. 147-163.
52. Fiset, P., D.A. Johnson, and J.C. Samin, *A fully symbolic generation of the equations of motion of multibody systems containing flexible beams*. Computer Methods in Applied Mechanics and Engineering, 1997. **142**(1-2): p. 123-152.
53. Hu, Q., Y. Jia, and S. Xu, *A new computer-oriented approach with efficient variables for multibody dynamics with motion constraints*. Acta Astronautica, 2012. **81**: p. 380-389.
54. Jiang, W., et al., *Symbolic formulation of large-scale open-loop multibody systems for vibration analysis using absolute joint coordinates*. Journal of System Design and Dynamics, 2008. **2**(4): p. 1015-1026.
55. Rand, R.H. and D. Armbruster, *Perturbation Methods, Bifurcation Theory and Computer Algebra*, 1987.
56. Macsyma, I., *Macsyma Mathematics and System Reference Manual*. 1974, DTIC Document.
57. Guide, M.U., *Maplesoft, a division of Waterloo Maple Inc*. 2014.
58. Sanchez, N.E., *The method of multiple scales: Asymptotic solutions and normal forms for nonlinear oscillatory problems*. Journal of Symbolic Computation, 1996. **21**(2): p. 245-252.
59. Corless, R.M., et al., *Two perturbation calculations in fluid mechanics using large-expression management*. Journal of Symbolic Computation, 1997. **23**(4): p. 427-443.
60. Vakhidov, A.A. and N.N. Vasiliev, *A new approach for analytical computation of Hamiltonian of a satellite perturbed motion*. Journal of Symbolic Computation, 1997. **24**(6): p. 705-710.
61. Yu, P., *Computation of normal forms via a perturbation technique*. Journal of Sound and Vibration, 1998. **211**(1): p. 19-38.
62. Yu, P., *Symbolic computation of normal forms for resonant double Hopf bifurcations using a perturbation technique*. Journal of Sound and Vibration, 2001. **247**(4): p. 615-632.
63. Yu, P., *Analysis on double hopf bifurcation using computer algebra with the aid of multiple scales*. Nonlinear Dynamics, 2002. **27**(1): p. 19-53.

64. Wang, K., et al., *Derivation of the amplitude equation for reaction-diffusion systems via computer-aided multiple-scale expansion*. International Journal of Bifurcation and Chaos, 2014. **24**(7).
65. Franciosi, C. and S. Tomasiello, *The use of mathematica for the analysis of strongly nonlinear two-degree-of-freedom systems by means of the modified Lindstedt-Poincaré method*. Journal of Sound and Vibration, 1998. **211**(2): p. 145-156.
66. Pismen, L.M., B.Y. Rubinstein, and M.G. Velarde, *On automated derivation of amplitude equations in nonlinear bifurcation problems*. International Journal of Bifurcation and Chaos in Applied Sciences and Engineering, 1996. **6**(11): p. 2163-2167.
67. Pismen, L.M. and B.Y. Rubinstein, *Computer tools for bifurcation analysis: General approach with application to dynamical and distributed systems*. International Journal of Bifurcation and Chaos in Applied Sciences and Engineering, 1999. **9**(6): p. 983-1008.
68. Warminski, J., et al., *Approximate analytical solutions for primary chatter in the nonlinear metal cutting model*. Journal of Sound and Vibration, 2003. **259**(4): p. 917-933.
69. Cartmell, M.P. and D.I.M. Forehand, *Visualisation of Symbolic Computational Dynamics*, in *XXII ICTAM*. 2008: University of Adelaide, Australia.
70. Cartmell, M., *Introduction to linear, parametric and nonlinear vibrations*. 1990: Chapman and Hall London.
71. Wolfram, S., *The Mathematica Book*, 1999.
72. Roberts, J.W. and M.P. Cartmell, *Forced vibration of a beam system with autoparametric coupling effects*. Strain, 1984. **20**(3): p. 123-131.
73. Cartmell, M.P., *The equations of motion for a parametrically excited cantilever beam*. Journal of Sound and Vibration, 1990. **143**(3): p. 395-406.
74. Abou-Rayan, A.M., et al., *Nonlinear response of a parametrically excited buckled beam*. Nonlinear Dynamics, 1993. **4**(5): p. 499-525.
75. Anderson, T.J., B. Balachandran, and A.H. Nayfeh, *Nonlinear resonances in a flexible cantilever beam*. Journal of Vibration and Acoustics, Transactions of the ASME, 1994. **116**(4): p. 480-484.
76. Lee, W.K. and K.Y. Soh, *Nonlinear analysis of the forced response of a beam with three mode interaction*. Nonlinear Dynamics, 1994. **6**(1): p. 49-68.

# Appendices

Appendix: algorithm 1 Power function encoding where the index is a negative integer.

```
PowerFunction4(Line 20 SEEMcore)
1: if index == Integer and index = 1
2:   pos <- position base in the indexterm;
3:   if pos == {}
4:     out1 <- define a new encoding vector;
5:   else
6:     out1 <- update the encoding vector;
7:   end
8:   pos2 <- position base in the indexterm;
9:   if pos2 == {}
10:    out2 <- define a new encoding vector;
11:   else
12:    out2 <- update the encoding vector;
13:   end
14:   out1*out2
15: end
```

Appendix: algorithm 2 Power function encoding where the base is in an exponential form and the index is in multiplication from.

```
PowerFunction5(Line 20 SEEMcore)
1: if base == e and index == expression
2:   out <- 1
3:   Remove the time scales from the index;
4:   arg <- e ^ each item in the index;
5:   for each arg
6:     if there is a complex term in the arg
7:       im <- index
8:       if coefficient of the im == 1 or -1
9:         out1 <- 1
10:      else
11:        pos <- find position im in the indexterm
12:        if pos == {}
13:          out1 <- define a new encoding
14:        else
15:          out1 <- update the encodings
16:        end
17:      end
18:    end
19:    pos <- find the position of the base (e)
20:    if pos == {}
21:      ou2 <- define a new encoding
22:    else
23:      ou2 <- update the encodings
24:    end
25:    out <- out*out1*out2
26:  end
27:  out
28: end
```

Appendix: algorithm 3 Power function encoding where base is either a summation or multiplication.

```
PowerFunction6(Line 20 SEEMcore)
1: if base == expression or summation
2:   arg <- extraxt the term
3:   for each arg
4:     pos <- find the position of the arg in the indextrm
5:     if pos == {}
6:       out1 <- define a new encoding vector
7:     else
8:       out1 <- update the encoding vector
9:     end
10:    out <- out1*out
11:  end
12: end
```

Appendix: algorithm 4 Power function encoding where base is a derivative of the dependent variable.

```
PowerFunction7(Line 20 SEEMcore)
1: if base == Derivative of the dependent variables
2:   do noting
3: else
4:   pos <- find the position of the base
5:   if pos == {},
6:     out1 <- define a new encoding vector
7:   else
8:     out1 <- update the encoding vector
9:   end
10:  pos <- find the position of the index
11:  if pos == {},
12:    out2 <- define a new encoding vector
13:  else
14:    out2 <- update the encoding vector
15:  end
16:  out1*out2
17: end
```

#### Appendix: algorithm 5 Complex encoding

```

ComplexFunction (Line 23 SEEMcore)
1: if subterm == Complex
2:   im <- imaginary part of the subterm;
3:   if im == 1 or -1
4:     out1 <- 1
5:   else
6:     pos <- position im in the indextrm
7:     if pos == {},
8:       out1 <- define a new encoding vector
9:     else
10:      out1 <- update the encoding vector
11:    end
12:    pos <- find position the imaginary unit
13:    if pos == {},
14:      out2 <- define a new encoding vector
15:    else
16:      out2 <- update the encoding vector
17:    end
18:    update indextrm;
19:    out1*out2
20:  end
21: end

```

#### Appendix: algorithm 6 Integer encoding

```

NoneZeroIntegerFunction(Line 26 SEEMcore)
1: if subterm == Integer not equal to 0
2:   pos <- position subterm
3:   if pos == {}
4:     out <- define a new encoding vector
5:   else
6:     out <- update the encoding vector
7:   end
8: end

```

#### Appendix: algorithm 7 Rational encoding

```

RationalFunction(Line 26 SEEMcore)
1: if subterm == rational
2:   nom <- nominator
3:   den <- denominaor;
4:   pos1 <- find position nom in indextrm;
5:   if pos1 == {}
6:     out1 <- define a new encoding vector
7:   else
8:     out1 <- update the encoding vector
9:   end
10:  pos2 <- find position den in indextrm;
11:  if pos == {}
12:    out2 <- define a new encoding vector
13:  else
14:    out2 <- update the encoding vector
15:  end
16:  out1*out2
17: end

```

## Appendix: algorithm 8 Derivation encoding

```

DifferentiationFunction(Line 32 SEEMcore)
1: if subterm == Derivation of one of the dependent variables
2:   arg <- dependent variable after
3:   pos <- position arg
4:   if pos === {}
5:     out1 <- define a new encoding vector
6:   else
7:     out1 <- update the encoding vector
8:   end
9:   pos <- position derivation operator
10:  if pos === {}
11:    out2 <- define a new encoding vector
12:  else
13:    out2 <- update the encoding vector
14:  end
15:  out1*out2
16: end

```

## Appendix: algorithm 9 Partial derivative encoding

```

DifferentiationFunction(Line 32 SEEMcore)
1: if subterm == Derivation
2:   arg <- dependent variable after
3:   pos <- position arg
4:   if pos == {} or arg == pure function
5:     out1 <- define a new encoding vector
6:   else
7:     out1 <- update the encoding vector
8:   end
9:   pos <- position derivation operator
10:  if pos == {}
11:    out2 <- define a new encoding vector
12:  else
13:    out2 <- update the encoding vector
14:  end
15:  out1*out2
16: end

```

## Appendix: algorithm 10 Complex amplitude encoding

```

ComplexAmp(Line 32 SEEMcore)
1: if sunterm == complex amplitude or its condugate
2:   pos <- find the position of the subterm
3:   if pos === {}
4:     out1 <- define a new encoding vector
5:   else
6:     out1 <- update the encoding vector
7:   end
8:   out1
9: end

```

## Appendix: algorithm 11 Other functions encodings

```

ComlexAmp(Line 32 SEEMcore)
1: if sunterm == dependent variable or its derivatives
2:   out1 <- 1
3: else
4:   pos <- find the position of the subterm
5:   if pos == {}
6:     out1 <- define a new encoding vector
7:   else
8:     out1 <- update the encoding vector
9:   end
10:  out1
11: end

```

## Display function algorithms

Appendix: algorithm 12 Exponential quantities that are having a summation of symbol as the  
index

```

DisplayPowerFunction( line display )
1: if subterm == power function
2:   convert any string from to expression;
3:   if vase == e and index == summation
4:     ay <- change the index form to a+bI;
5:     im <- take the imaginary part of ay out;
6:     arg <- remove time related symbols from index;
7:     argOrig <- save the original from of the index;
8:     pos <- find position of any compound level term;
9:     strng <- drop the identity factor from the encoding vector;
10:    remove the used encoding vector from the intextrm;
11:    for each element in the index
12:      if element == summation or multiplication
13:        NewVar <- decompose the element;
14:        for each NewVar
15:          pos <- position NewVar in the indextrm;
16:          if pos == {}
17:            strngarg <- "";
18:          else
19:            strngarg <- shape the encoding vector
20:          end
21:          Replace the element in the argOrig with the tooltip
22:        end
23:      else
24:        for each element
25:          pos <- position element in the indextrm;
26:          if pos == {}
27:            strngarg <- "";
28:          else
29:            strngarg <- shape the encoding vector
30:          end
31:          Replace the element in the argOrig with the tooltip
32:        end
33:      end
34:    end
35:    out <- (ToolTip[E, strng])^(ToolTip[im, strng]*argOrig*ToolTip[I,
36: strng])
37:  end

```

## Appendix: algorithm 13 Power function with differentiations operator in the base

```

DisplayDerivationPerturbation
1: if base == derivation of any perturbations,
2:   var <- original from of the base
3:   arg <- input of the derivation;
4:   posarg <- find position arg in indextrm;
5:   pos <- position arg in the indextrm;
6:   if pos == {}
7:     strngarg <- "";
8:   else
9:     strngarg <- shape the encoding vector
10:  end
11:  input <- convert the appearance of arg to the user end format;
12:  pos <- position derivation function in the indextrm;
13:  if pos == {}
14:    strng <- "";
15:  else
16:    strng <- shape the encoding vector
17:  end
18:  diff <- convert the appearance of derivation operator to the user
19: end format;
20:  pos <- find the position of the index;
21:  if pos == {}
22:    strngpwr <- "";
23:  else
24:    strngpwr <- shape the encoding vector
25:  end
26:  out <- ToolTip[diff, strng]* (ToolTip[input, strngarg]^ToolTip[index,
27: strngpwr])
28: end

```

Appendix: algorithm 14 Power function where the base is either a summation or multiplication  
and the index is an integer

```

Powerfunction2
1: if base == expression or submission and index == integer
2:   if index == -1
3:     out <- simply from of the base;
4:     item <- elements of base in a list format;
5:     for each item
6:       if item == I
7:         vec <- position compound from encodings;
8:         strng <- convert the encoding vector;
9:         out <- out replace I with ToolTip[I, strng];
10:      else
11:        if item == Sting
12:          varr <- expression form of the string;
13:          vec <- find the position of varr in indextrm;
14:          strng <- convert the encoding vector;
15:          out <- out replace varr with ToolTip[varr, strng];
16:        else
17:          vec <- find the position of element in indextrm;
18:          strng <- convert the encoding vector;
19:          out <- out replace varr with ToolTip[element, strng];
20:        end
21:      end
22:    end
23:    1/ out
24:  else
25:    out <- simply from of the base;
26:    item <- elemnets of base in a list format;
27:    for each item
28:      if item == i
29:        vec <- position compound from encodings;
30:        strng <- convert the encoding vector;
31:        out <- out replace I with ToolTip[I, strng];
32:      else
33:        if item == Sting
34:          varr <- expression form of the string;
35:          vec <- find the position of varr in indextrm;
36:          strng <- convert the encoding vector;
37:          out <- out replace varr with ToolTip[varr, strng];
38:        else
39:          vec <- find the position of element in indextrm;
40:          strng <- convert the encoding vector;
41:          out <-out replace varr with ToolTip[elemnet, strng];
42:        end
43:      end
44:    end
45:    out
46:  end
47: end

```

## Appendix: algorithm 15 Power function where the index is a negative integer

```
Powerfunction3
1: if base == expression or summation
2:   pos <- find the position of power index;
3:   if pos == {}
4:     strng <- "";
5:   else
6:     strng <- develop encoding vector from indextrm;
7:   end
8:   pwr <- Tooltip[ power, strng];
9:   arg <- list from of the base;
10:  head <- function type of the base;
11:  for each arg
12:    pos <- find the position of arg index;
13:    strng <- develop encoding vector from indextrm;
14:    replace each arg with Tooltip[arg,strng];
15:  end
16:  apply the head to the encoded arg;
17:  arg^pwr
18: end
```

## Appendix: algorithm 16 Any other power function format

```

Powerfunction4 (line XXXX )
1: if subterm == any other power function
2:   out1 <- base^list form of power (removing the time elements);
3:   out2 <- original form of the subterm;
4:   if base == summation,
5:     for each element in the base
6:       pos <- find the position of power index;
7:       if pos == {}
8:         strng <- "";
9:       else
10:        strng <- develop encoding vector from indextrm;
11:      end
12:      out2 <- replace element with Tooltip[element, string];
13:    end
14:  else
15:    if base == exponential and there is a complex element in the index
16:      im <- imaginary parts of the index in a list form
17:      outim <- 1;
18:      if multiplication of the list == 1 or -1
19:        pos <- find E form encoding vector in the indextrm;
20:        if pos == {}
21:          strng <- "";
22:        else
23:          strng <- develop encoding vector from indextrm;
24:        end
25:        out <- Tooltip[E, strng]^{ Tooltip[I, strng]}
26:      else
27:        for each element in im
28:          pos <- imaginary parts of the element in a list form;
29:          if pos == {}
30:            strngim <- "";
31:          else
32:            strngim <- develop encoding vector from indextrm;
33:          end
34:          outim <- outim* Tooltip[element, strngim];
35:        end
36:        pos <- find position exponential function in the indextrm;
37:        if pos == {}
38:          strng <- "";
39:        else
40:          strng <- develop encoding vector from indextrm;
41:        end
42:        out <- Tooltip[E, strng]^{outim* Tooltip[I, strng]}
43:      end
44:    else
45:      if base == string
46:        out <- 1;
47:        var <- expression from the base;
48:        if var == Integer or Rational or Complex;
49:          update out2 with the expression from;
50:          for each var
51:            pos <- find position var function in the indextrm;
52:            if pos == {}
53:              strng <- "";
54:            else
55:              strng <- develop encoding vector from indextrm;
56:              remove pos from indextrm

```

```
57:             end
58:             out <- out * out*Tooltip[var*, strng]
59:         end
60:     end
61: end
62: else
63:     if index == -1
64:         input <- change the base to the enduser format;
65:         pos <- find position base in the indextrm;
66:         if pos == {}
67:             strng <- "";
68:         else
69:             strng <- develop encoding vector from indextrm;
70:         end
71:         out <- Tooltip[input, strng]^-1
72:     else
73:         input <- change the base to the end user format;
74:         pos <- find position base in the indextrm;
75:         if pos == {}
76:             strng <- "";
77:         else
78:             strng <- develop encoding vector from indextrm;
79:         end
80:         pospr <- find position base in the indextrm;
81:         if pospr == {}
82:             strngpr <- "";
83:         else
84:             strngpr <- develop encoding vector from indextrm;
85:         end
86:         out <- Tooltip[input, strng]^Tooltip[power, strngpr]
87:     end
88: end
89: end
90: end
```

### B. Multiple Scales Applied to a Parametrically Excited Beam

The equation of motion for a cantilever model with an end-mass is taken from Cartmell [70] and it is presented in equation (B.1). In this equation,  $X$  represents the non-dimensionalized coordinate which indicates lateral displacement of the end-mass,  $\omega$  specifies the natural frequency,  $\zeta$  defines the damping coefficient, and finally  $\Omega$  signifies the external excitation frequency.

$$\ddot{X} + 2\varepsilon\zeta\omega\dot{X} + \omega^2X - \varepsilon \cos(\Omega t)X = 0 \quad (\text{B.1})$$

To apply the multiple scales method to the equation of motion of the beam,  $X$  is perturbed up to the first order  $\varepsilon$ , truncating the series after that. Equation (B.2) is used to obtain the first and second derivatives of  $X$ . The expansion and derivatives are substituted into the original equation (B.1). The zeroth and the first order perturbations are presented by equations (B.3) and (B.4), respectively.

$$X = X_0 + \varepsilon X_1 \quad (\text{B.2})$$

$$D_0^2 X_0 + \omega^2 X_0 = 0 \quad (\text{B.3})$$

$$D_0^2 X_1 + \omega^2 X_1 = -2D_0 D_1 X_0 - 2\zeta\omega D_0 X_0 + X_0 \cos(\Omega T_0) \quad (\text{B.4})$$

The solution for the homogeneous zeroth-order perturbation equation (B.3) is obtained by inspection and is shown in equation (B.5), noting that the bars denote complex conjugates.  $A[T_1]$  is an arbitrary function of slow time scale  $T_1$ , which facilitates the diminishing of secular terms in the solution of the first order perturbation equation.

$$X_0 = (A[T_1]e^{i\omega T_0} + \bar{A}[T_1]e^{-i\omega T_0}) \quad (\text{B.5})$$

The solution of the zeroth-order perturbation equation is then substituted into the first order perturbation equation and the exponential form of the excitation function ( $\cos(\Omega T_0) = \frac{1}{2}[e^{i\Omega T_0} + e^{-i\Omega T_0}]$ ) is used (B.6), noting that  $CC$  is used as an abbreviation for Complex Conjugates.

$$D_0^2 X_1 + \omega^2 X_1 = e^{i\omega T_0} \left[ \frac{A[T_1]}{2} e^{i\Omega T_0} + \frac{\bar{A}[T_1]}{2} e^{iT_0(\Omega-2\omega)} - i2\omega D_1 A[T_1] - i2\zeta\omega^2 A[T_1] \right] + CC \quad (\text{B.6})$$

The uniformity of the expansion must be valid during the solution procedure, therefore *secular terms* should vanish from the first order perturbation equation. The secular terms are mostly generated by resonant excitation terms and, if retained, would cause a disproportionate increase in the magnitude of the solution of the first order perturbation equation ( $X_1$ ) when compared to that of the zeroth-order ( $X_0$ ). Resonant terms are oscillating at the natural frequency of the homogeneous system. To avoid secular terms in the general solution, resonant exponent terms (in this equation containing  $e^{i\omega T_0}$ ) are taken out and set to zero (B.7).

$$\frac{A[T_1]}{2} e^{i\Omega T_0} + \frac{\bar{A}[T_1]}{2} e^{iT_0(\Omega-2\omega)} - i2\omega D_1 A[T_1] - i2\zeta\omega^2 A[T_1] = 0 \quad (\text{B.7})$$

Considering equation (B.7), the last two terms are clearly resonant terms and must vanish from the particular solution. However, the first two terms might be considered to be resonant but this is depending on the value of the external excitation frequency ( $\Omega$ ). If the external excitation frequency is set to  $2\omega$ , then the first term ( $\frac{A[T_1]}{2} e^{i2\omega T_0}$ ) will not generate a secular term (non-resonant). Finally the secular terms are identified and set to zero (B.8).

$$\frac{\bar{A}[T_1]}{2} e^{iT_0(\Omega-2\omega)} - i2\omega D_1 A[T_1] - i2\zeta\omega^2 A[T_1] = 0 \quad (\text{B.8})$$

The frequency which makes the second term of equation (B.7) secular, defines an important condition known as *principal parametric resonance* (B.9). To investigate the *near resonant* situation, a small latitude is allowed around the resonant point (B.10) and  $\rho$  and  $\varepsilon\rho$  are known as the *detuning* and *detuning parameter*, respectively.

$$\Omega = 2\omega \quad (\text{B.9})$$

$$\Omega = 2\omega + \varepsilon\rho \quad (\text{B.10})$$

The secular terms for the case when the excitation frequency is near to satisfying the case of principal parametric resonance is provided in equation (B.11).

$$\frac{\bar{A}[T_1]}{2} e^{iT_0(\varepsilon\rho)} - i2\omega D_1 A[T_1] - i2\zeta\omega^2 A[T_1] = 0 \quad (\text{B.11})$$

Equation (B.12) shows  $A[T_1]$  and  $\bar{A}[T_1]$  based on the steady state amplitude  $a[T_1]$  and phase  $\alpha[T_1]$ , which are parts of the solution to zeroth-order  $\varepsilon$  ( $X_0 = a \cos(\omega T_0 + \alpha)$ ). It is possible to replace  $A[T_1]$ ,  $\bar{A}[T_1]$  and  $D_1 A[T_1]$ , in equation (B.14), resulting in equation (B.14).

$$A[T_1] = \frac{a e^{i\alpha}}{2} \text{ and } \bar{A}[T_1] = \frac{a e^{-i\alpha}}{2} \quad (\text{B.12})$$

$$D_1 A[T_1] = \frac{1}{2} a' e^{i\alpha} + i \frac{1}{2} a \alpha' e^{-i\alpha} \quad (\text{B.13})$$

$$\frac{a}{4} e^{i(\varepsilon \rho T_0 - 2\alpha)} - i a' \omega - a \alpha' \omega - i \zeta \omega^2 a = 0 \quad (\text{B.14})$$

Subsequently, the real and imaginary parts of equation (B.14) are separated and shown in equation (B.15) and (B.16), noting that  $\varepsilon \rho T_0 = \rho T_1$ .

$$\text{Re:} \quad \cos(\rho T_1 - 2\alpha) + a \alpha' \omega = 0 \quad (\text{B.15})$$

$$\text{Im:} \quad \sin(\rho T_1 - 2\alpha) - i a' \omega - i \zeta \omega^2 a = 0 \quad (\text{B.16})$$

To investigate the amplitude  $a$ , two approaches could be considered [70]. The first one is to solve the system of equations (B.15) and (B.16) by numerical integration, whilst the other is to use equations (B.15) and (B.16) to derive an analytical expression for the detuning parameter ( $\varepsilon \rho$ ), which is accomplished in this report. As shown in equations (B.15) and (B.16), all parameters are functions of slow time ( $T_1$ ) rather than the fast time scale ( $T_0$ ). Therefore, it is acceptable to assume that the amplitude  $a$  is almost static at the slow time scale (B.17). An autonomous system ( $\psi$ ) is then defined to avoid the explicit presence of any time scales (B.18).

$$a' \simeq 0 \quad (\text{B.17})$$

$$\psi = \rho T_1 - 2\alpha \quad (\text{B.18})$$

Then the solvability equations (B.19) and (B.20) are derived by substituting (B.17) and (B.18) into (B.15) and (B.16).

$$\cos(\psi) + a \alpha' \omega = 0 \quad (\text{B.19})$$

$$\sin(\psi) - i \zeta \omega^2 a = 0 \quad (\text{B.20})$$

The next step is to determine the value for the slow varying phase angle ( $\alpha'$ ). If it is considered to be zero, the contribution of equation (B.15) will be lost. Therefore, a helpful analytical substitution based on an assumption of constant phase in the continuous system is considered (B.21), and the slowly varying phase angle is re-expressed in terms of the detuning, which itself generally has a small magnitude.

$$\psi' = 0; \rho - 2\alpha' = 0; \alpha' = \frac{\rho}{2} \quad (\text{B.21})$$

It is possible to substitute equation (B.21) into the solvability equations and derive the equation based on the detuning ( $\rho$ ), equations (B.22) to (B.24)

$$\frac{a^2}{16} \cos^2(\psi) + \frac{a^2}{4} \rho^2 \omega^2 = 0 \quad (\text{B.22})$$

$$\frac{a^2}{16} \sin^2(\psi) - a^2 \zeta^2 \omega^4 = 0 \quad (\text{B.23})$$

$$1 - 16 \zeta^2 \omega^4 + 4\rho^2 \omega^2 = 0 \quad (\text{B.24})$$

Both the un-damped system ( $\zeta = 0$ ) and the damped system ( $\zeta \neq 0$ ) are investigated. In the un-damped case, the second term of equation (B.24) vanishes and the detuning value can be determined (B.25). Then the detuning is replaced by the nearly resonant excitation frequency (B.10), with the resulting equation (B.27).

$$\rho = \pm \frac{1}{2\omega} \quad (\text{B.25})$$

$$\Omega = 2\omega \pm \frac{\varepsilon}{2\omega} \quad (\text{B.26})$$

The same approach as above is taken for the damped case. The detuning and the detuning parameter are shown in equations (B.27) and (B.28), respectively. Furthermore, the value of  $\varepsilon$  for this case is shown in equation (B.29),  $B_1 \sqrt{\frac{m_0}{I_0}}$  is a function of material properties and geometry [70].

$$\rho = \pm \sqrt{\frac{1}{4\omega^2} - 4\zeta^2 \omega^2} \quad (\text{B.27})$$

$$\Omega = 2\omega \pm \sqrt{\frac{\varepsilon^2}{4\omega^2} - 4\zeta^2 \omega^2} \quad (\text{B.28})$$

$$\varepsilon = W_0 \Omega^2 B_1 \sqrt{\frac{m_0}{I_0}} \quad (\text{B.29})$$

## Experimental investigation

A spring steel beam with an end-mass was designed and built in order to compare the theoretical model with the experimental case [73-76]. Principal parametric resonance of the beam was experimentally confirmed as well as points on the stability chart which express the overall transition curve. Then the experimental data are compared to the theoretical solution (B.28).

A rectangular spring steel beam, with a cross-sectional area of  $7.00 \times 10^{-3} m^2$  and thickness of  $1.22 \times 10^{-3} m$  was manufactured. In order to have the ability to tune the system, two stainless steel rectangular masses with a total mass of  $9.1 \times 10^{-2} kg$  were attached to one end of the beam. The other end was fixed on to the base of a vertically oriented electromagnetic shaker (Figure B. 1)

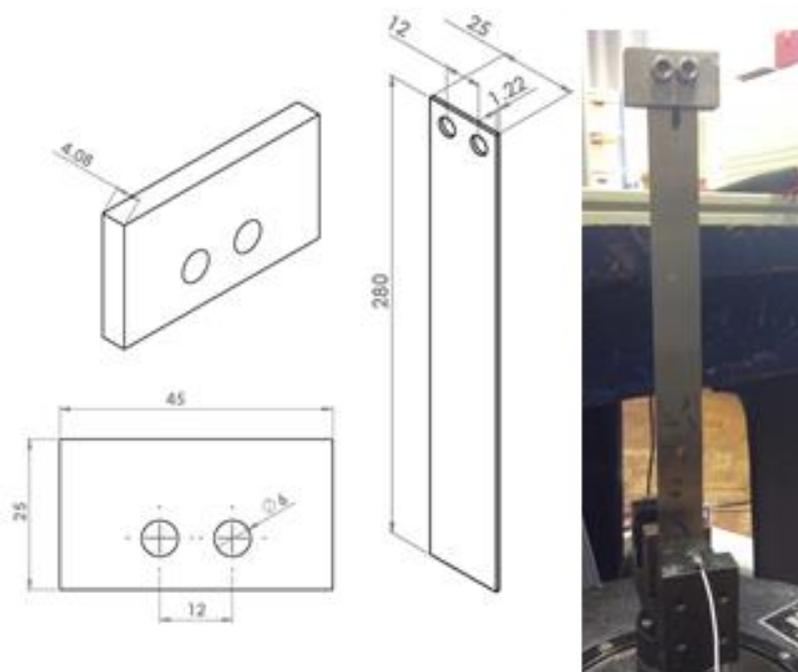


Figure B. 1 Spring steel beam and the end mass geometry configurations (mm).

Initially the region of principal parametric resonance of the test beam was determined. As shown in equation (B.9) this frequency is equal to twice the natural frequency of the system. The natural frequency was determined by performing a free vibration test which gave a value of  $6.012 Hz$ . Therefore, the principal parametric resonance was found to occur at approximately  $12.024 Hz$ .

The second part of the experiment was to determine points defining the transition curve. The shaker excitation frequency was set to the value of the principal parametric resonance, with the minimum excitation amplitude. Then a slight perturbation was applied to the tip of the beam which led to a very large displacement in the perpendicular direction to the excitation axis. The excitation acceleration and frequency of the excitation were measured by an accelerometer that was placed at the base of the beam supported on the shaker. This process was continued for frequencies around principal parametric resonance. By that means, after setting the excitation frequency, the amplitude of excitation was increased gradually until the nonplanar motion was observed and recorded.

To facilitate the experiment procedure and data acquisition process, a LabVIEW programme was written. This programme can take the principal parametric resonance value as an input and increase the excitation amplitude gradually for a predefined time period, until the user senses the nonplanar motion of the beam and stops the experiment manually in the programme. Then the accelerometer output, for the last detected time step, and with the value of excitation frequency is extracted. This programme could be developed in the future by adding a feedback loop, and the horizontal displacement of the beam could be measured by another sensor to detect the nonplanar motion and stop the programme automatically.

To decrease the experimental error, principally the human error, the experiment was repeated five times. In the post processing step, the frequency of excitation and the corresponding amplitude for each point were determined and plotted against each other.

The experimental stability curve is shown in Figure B.2 by discrete the triangular points. It can be noted that the experimental data could not cover all the theoretical range. This could be addressed by increasing the capacity of the electrodynamic shaker. The shaker is not capable of generating the required excitation amplitude for excitation frequencies of less than 9 Hz and more than 20 Hz.

The theoretical results were evaluated by using equations B.28 and B.29 and plotted against the experimental data. Considering Figure B. 2, the theoretical and experimental results were almost in the same range at the threshold area (this is magnified in the top right-hand side). Although as the excitation frequency moves from the principal parametric resonant point, more of a difference between the theoretical and experimental results could be observed. These might be due to the nonlinear stiffness, or some stability effects.

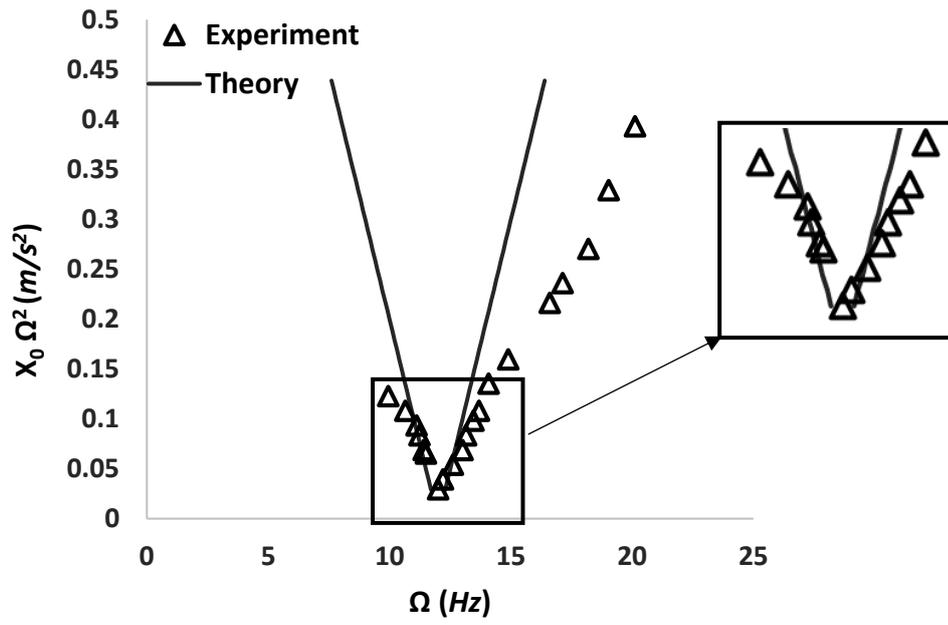


Figure B. 2 Comparison of the experimental and theoretical results for the stability region.

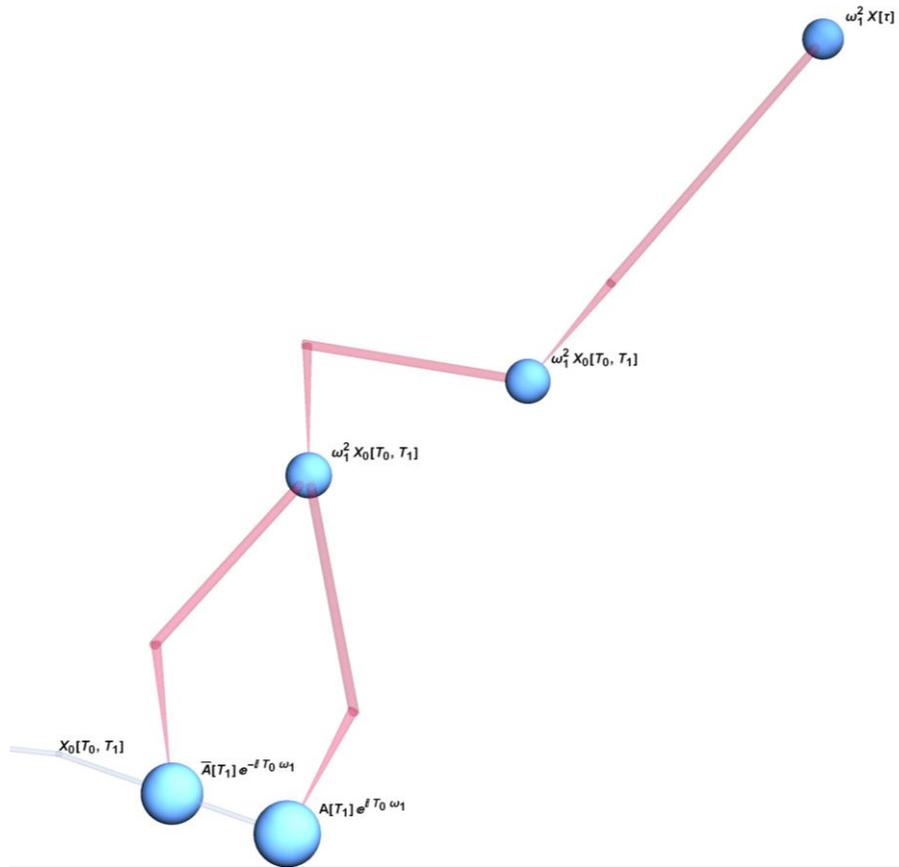
### C. The blueprint method applied to the problem of autoparametrically excited beam system

The zeroth-order perturbation equation solution is given in eqn (13). This is a part of the overall solution for the primary beam, and  $A$  is an arbitrary complex amplitude which is a function of the slow time scale.

$$X_0 = Ae^{iT_0\omega_1} + e^{-iT_0\omega_1}\bar{A} \quad \text{eqn (13)}$$

Figure 6-23 shows the general visualisation graph for this equation and it is possible to track the natural frequency ( $\omega_1$ ) to the stiffness term ( $\omega^2 X[t]$ ) in the equation of motion of the primary beam. Interestingly, this can quickly show the linear solution for the system as all the first perturbation order links are hidden in this graph.

In this case study, it was decided to connect the source of the arbitrary complex amplitude  $A$  to the zeroth-order solution equation number, not to the perturbation expansion. Conceptually, from the mathematical reasoning inherent to the method, this amplitude can be related to the  $X_0$  in the perturbation expansion, eqn (3). However, as this symbol does not appear directly in that equation this might cause confusion in the later stages. This is another example of the flexibility that is available and also necessary in the application of SCD solvers.



$$X_0 [T_0, T_1] =$$

$$\begin{aligned}
 & \begin{pmatrix} A [T_1] \\ (13, \theta) \end{pmatrix} e^{i T_0 \omega_1} \begin{pmatrix} (13, \theta) \\ (1, \theta, 13) \end{pmatrix} + \\
 & \begin{pmatrix} \bar{A} [T_1] \\ (13, \theta) \end{pmatrix} e^{-i T_0 \omega_1} \begin{pmatrix} (13, \theta) \\ (1, \theta, 13) \end{pmatrix}
 \end{aligned}$$

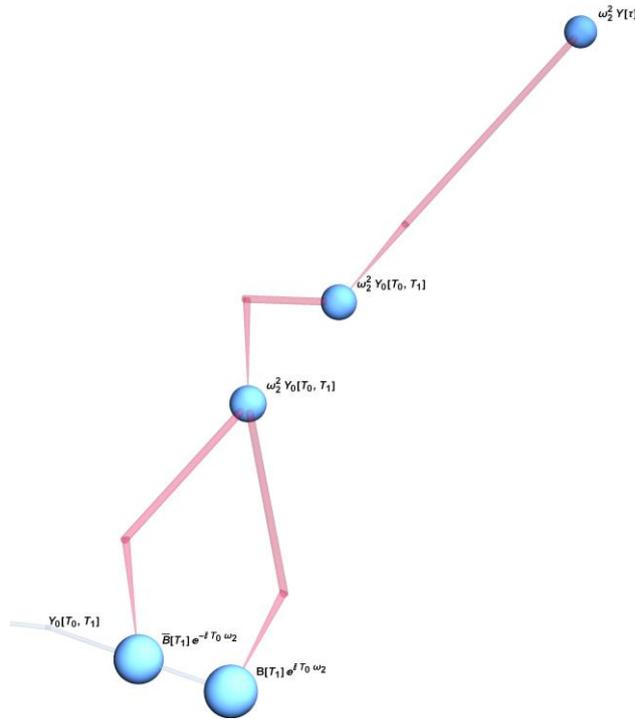
Figure 6-23 Overall visualisation graph of eqn (13).

The zeroth-order perturbation solution for the secondary beam is given in eqn (14); where,  $B$  is the arbitrary complex amplitude and is a function of slow time scale  $T_1$ .

$$Y_0 [T_0, T_1] = B e^{i T_0 \omega_2} + e^{-i T_0 \omega_2} \bar{B} \tag{eqn (14)}$$

The details of the Blueprint visualisation method and the SEEM encodings are shown in Figure 6 24 and it is possible to track the natural frequency to the stiffness term in the equation of motion for the primary beam. All the links are zeroth perturbation order and are shown in pink, and the SF numerical values are roughly the same.

The source of the arbitrary complex amplitudes it was set to eqn (14). It is also correct to connect these amplitudes into the perturbation expansion for the secondary beam in eqn (4) as well.



$$Y_0 [T_0, T_1] ==$$

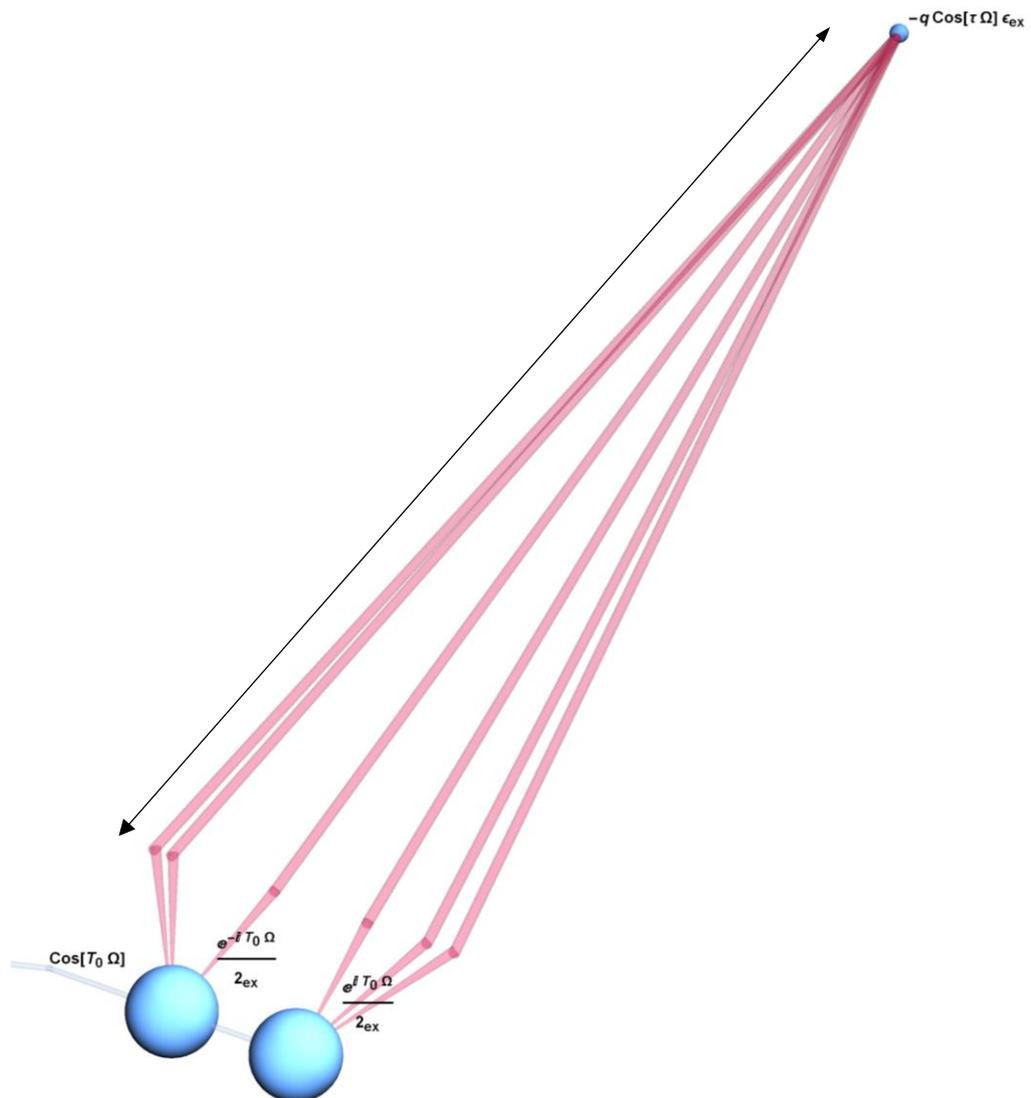
$$\begin{aligned}
 & \frac{B [T_1]}{(14, \theta)} e^{i T_0 \omega_2} + \frac{\bar{B} [T_1]}{(14, \theta)} e^{-i T_0 \omega_2} \\
 & \frac{\omega_2^2 Y_0 [T_0, T_1]}{(14, \theta)} + \frac{\omega_2^2 Y [T_1]}{(2, \theta, 14)}
 \end{aligned}$$

Figure 6-24 A detailed investigation of the encoding information of eqn (14).

The polar form of the external excitation is given in eqn (15). In order to identify the source of number 2, it was decided to add the *ex* subscript to this number.  $2_{ex}$  is introduced because of the trigonometrical identity that appears here and numerically has the same value as any general number 2.

$$\cos [T_0 \Omega] = \frac{e^{-iT_0 \Omega}}{2_{ex}} + \frac{e^{iT_0 \Omega}}{2_{ex}} \tag{eqn (15)}$$

Figure 6-25 describes the details of the encoding information for this equation. As expected all the quantities in this equation are linked to the external excitation term in the equation of motion for the primary beam. This is a strong term that is directly related to the physical act of excitation and all the links are shown in pink. Furthermore this equation is expressed in explicit form so the third digit of the SEEM is added here. The SF representatives for both terms are equal as they have the same source and contribution in this analysis. The large distance highlighted with an arrow in this plot shows that this equation is independent from the other mathematical procedure that were carried out before this step. In other words this equation can be defined in any stage of the analysis, and not necessarily after determining the solutions for the perturbation equations.



$$\cos [T_0 \Omega] = \frac{e^{i T_0 \Omega} (1, \theta, 15)}{2_{ex} (1, \theta, 15)} + \frac{e^{-i T_0 \Omega} (1, \theta, 15)}{2_{ex} (1, \theta, 15)}$$

Figure 6-25 A detailed investigation of the encoding information of eqn (15).

The multiple scales analysis continues by substituting the polar form of the external excitation and the solutions of the zeroth-order perturbation equations for both the primary and secondary beams, into the first order perturbation equation for the primary beam (11), resulting in:

$$D_0^2 X_1 + \omega_1^2 X_1 = \frac{e^{-iT_0\Omega} q \varepsilon_{ex}}{2_{ex} \varepsilon_p} + \frac{e^{iT_0\Omega} q \varepsilon_{ex}}{2_{ex} \varepsilon_p} - \frac{i D_1 A e^{iT_0\omega_1} 2_{dr} \varepsilon_{dr} \omega_1}{\varepsilon_p} + \frac{2 A \bar{A} \gamma \varepsilon_{nsf} \omega_1^2}{\varepsilon_p} - \frac{\bar{A}^2 e^{-2iT_0\omega_1} \gamma \varepsilon_{nsf} \omega_1^2}{\varepsilon_p} - \frac{A^2 e^{2iT_0\omega_1} \gamma \varepsilon_{nsf} \omega_1^2}{\varepsilon_p} + \frac{i \bar{A} e^{-iT_0\omega_1} 2_{dmpf} \varepsilon_{dmpf} \zeta_1 \omega_1^2}{\varepsilon_p} - \frac{i A e^{iT_0\omega_1} 2_{dmpf} \varepsilon_{dmpf} \zeta_1 \omega_1^2}{\varepsilon_p} - \frac{2 \bar{B}^2 e^{-2iT_0\omega_2} \mu \varepsilon_{cplf} \omega_2^2}{\varepsilon_p} - \frac{2 B^2 e^{2iT_0\omega_2} \mu \varepsilon_{cplf} \omega_2^2}{\varepsilon_p} + \frac{i D_1 \bar{A} e^{-iT_0\omega_1} 2_{dr} \varepsilon_{dr} \omega_1}{\varepsilon_p} \tag{eqn 16}$$

The general overview of the visualisation links for eqn (16) is given in Figure 6-26. The fundamental quantities are visible at the upstream of this figure.

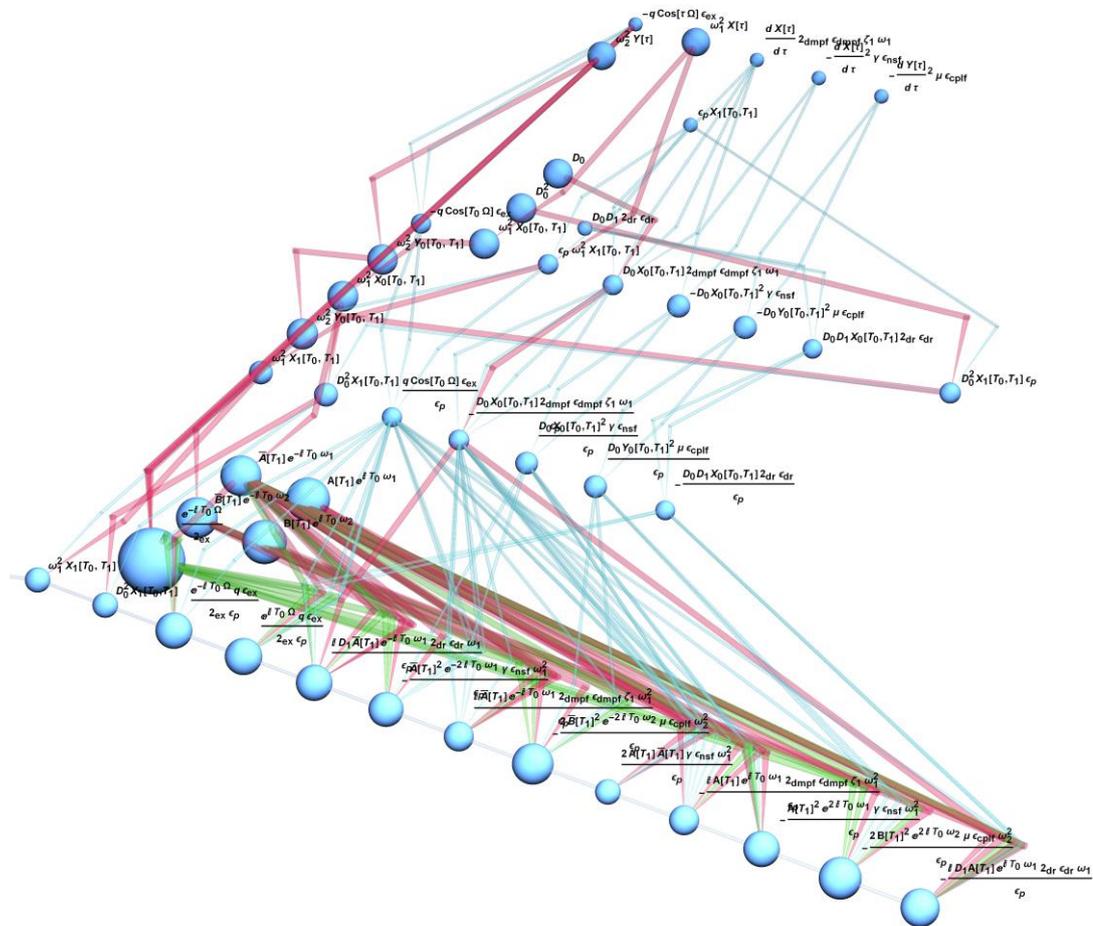
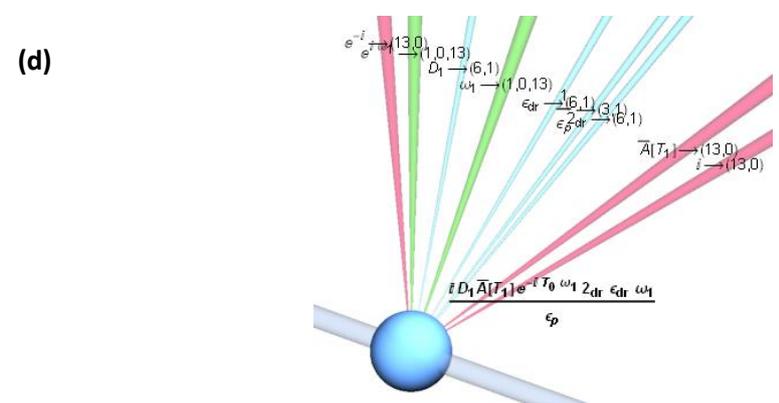
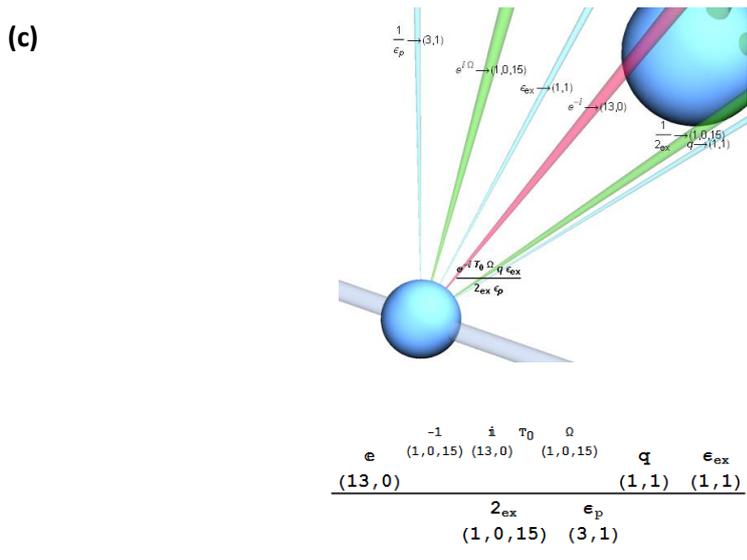
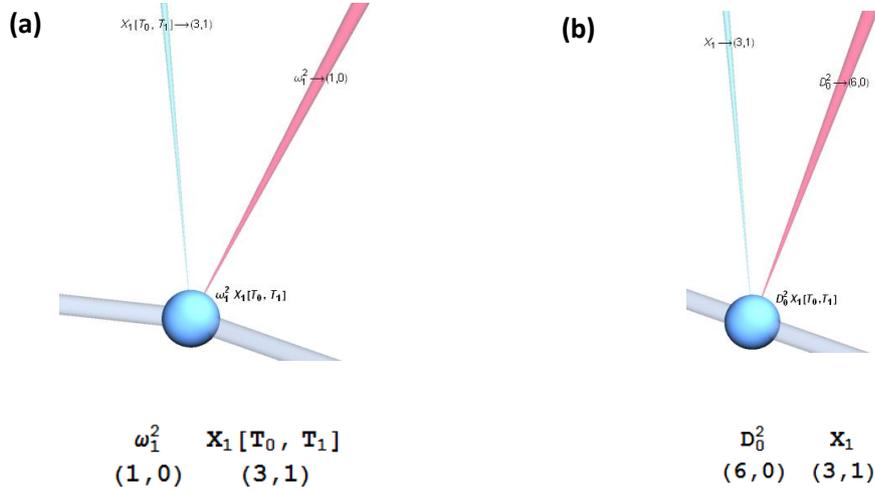


Figure 6-26 Overall visualisation graph for eqn (16).

More details about the SEEM information and the *Blueprint* visualisation method are given in Figure 6-27. The stiffness term is given in part (a) of this Figure, and has two constitutive quantities. The inertia term is shown in Figure 6-27-b, this term structured from equations 3 and 6. Part of the external excitation is shown in Figure 6-27-c, and the links showing the number  $2_{ex}$  are shown in green, whilst the links for  $2_{dr}$ , and  $2_{dmpf}$  are in pink. Considering the equation numbers of the sources for this term is mostly formed by means of the physical modelling, and  $\epsilon_p$  is the only quantity that is introduced by the perturbation method.

Figure 6-27-d shows a perturbational correction term which is structured from two green, three pink and four blue visualisation links. The natural frequency of the primary system is the only quantity that has an explicit physical definition, the rest are related to the mathematical process.

Considering the nonlinear stiffness term in Figure 6-27-e, there are two green, three red, and four blue links in this Figure. The quantities are sourced to equations 1, 3 and 13. The encodings clearly show the source of each quantity and there are two green, three pink and four blue links in this graph.



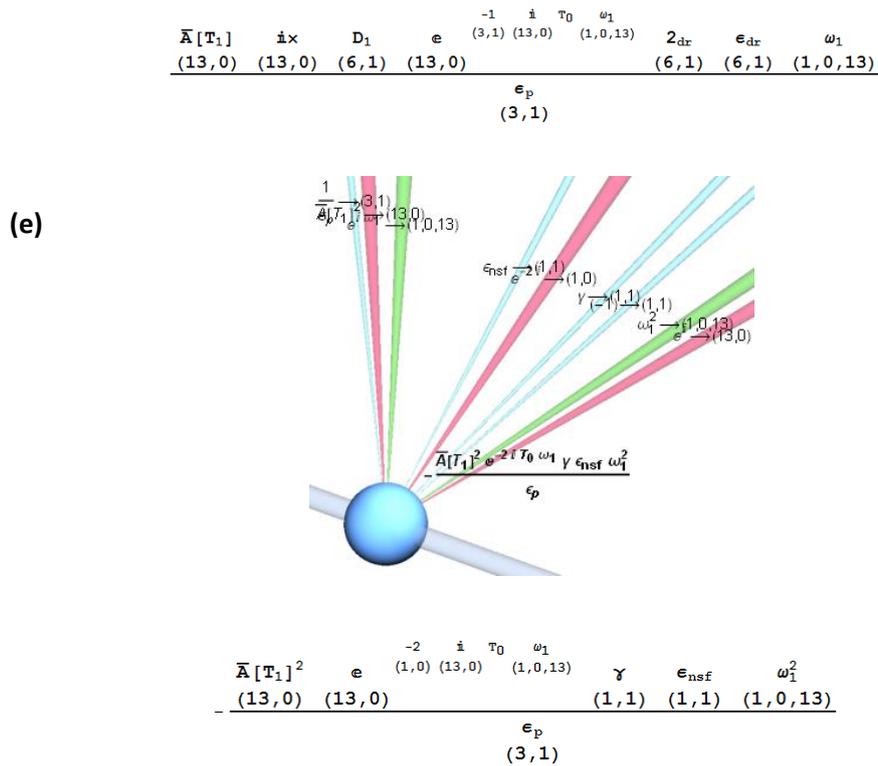


Figure 6-27 A detailed investigation of the encoding information for eqn (16); where, (a) is stiffness, (b) is inertia, (c) is external excitation, (d) is a perturbational correction term, (e) is nonlinear stiffness, (f) is a damping term and (g) is a coupling term.

Implementing the polar form of the external excitation, and the solutions of the zeroth-order perturbation equations for the both primary and secondary beams, into the first order perturbation equation for the secondary beam, results in:

$$\begin{aligned}
 D_0^2 Y_1 + \omega_2^2 Y_1 = & - \frac{\bar{A}\bar{B}e^{-iT_0\omega_1 - iT_0\omega_2} \epsilon_{cpls} \omega_1^2}{\epsilon_p} - \frac{A\bar{B}e^{iT_0\omega_1 - iT_0\omega_2} \epsilon_{cpls} \omega_1^2}{\epsilon_p} \\
 & - \frac{B\bar{A}e^{-iT_0\omega_1 + iT_0\omega_2} \epsilon_{cpls} \omega_1^2}{\epsilon_p} - \frac{A\bar{B}e^{iT_0\omega_1 + iT_0\omega_2} \epsilon_{cpls} \omega_1^2}{\epsilon_p} \\
 & - \frac{iD_1 e^{iT_0\omega_2} B 2_{dr} \epsilon_{dr} \omega_2}{\epsilon_p} + \frac{i e^{-iT_0\omega_2} D_1 \bar{B} 2_{dr} \epsilon_{dr} \omega_2}{\epsilon_p} \\
 & + \frac{i\bar{B} e^{-iT_0\omega_2} 2_{dmps} \epsilon_{dmps} \zeta_2 \omega_2^2}{\epsilon_p} - \frac{iB e^{iT_0\omega_2} 2_{dmps} \epsilon_{dmps} \zeta_2 \omega_2^2}{\epsilon_p}
 \end{aligned} \tag{eqn (17)}$$

The general overview of the visualisation links for eqn (17) is given in Figure 6-28. The fundamental quantities are visible at the upstream of this figure.

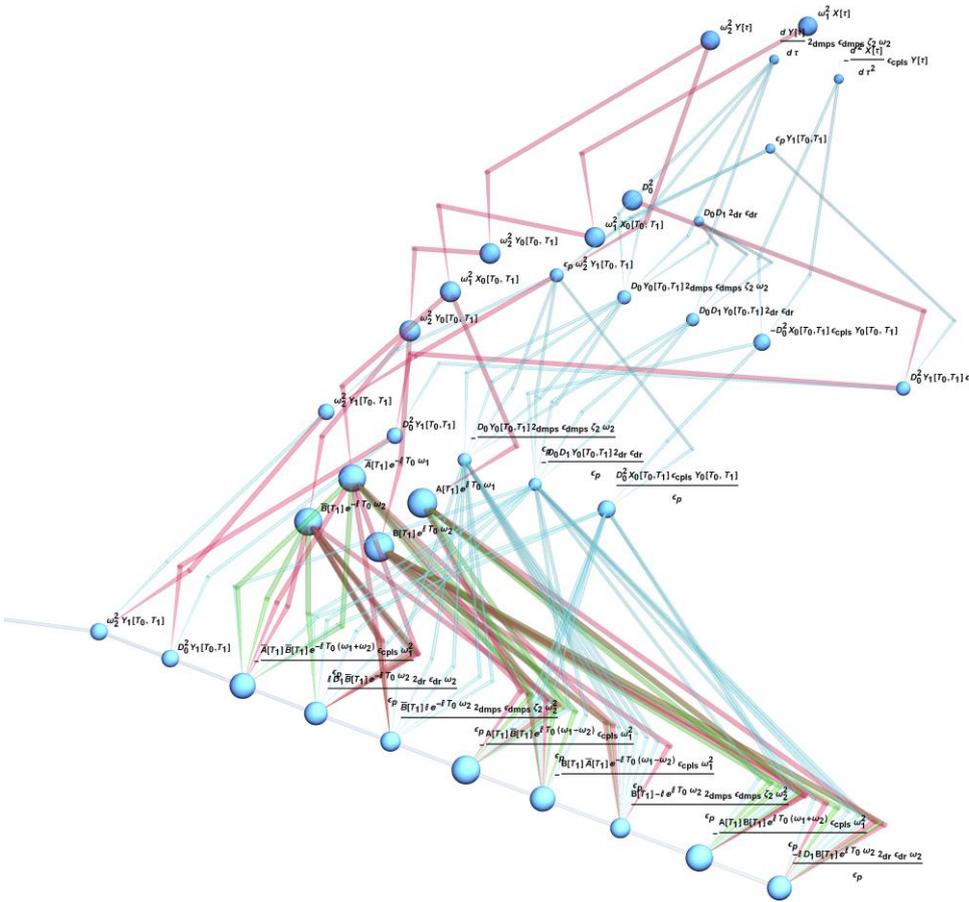


Figure 6-28 Overall visualisation graph for eqn (17).

Details of the SEEM information and visualisation links are shown in Figure 6-29. The stiffness and inertia terms for the secondary beam are given in Figure 6-29-a and b. The coupling term is shown in Figure 6-29-c, noting that this is a powerful term as it contains five dense visualisation links. It is possible to identify the influence of both the primary and secondary beams in this term. The contribution of both sub-systems' natural frequencies  $\omega_1$  and  $\omega_2$  can be identified in a single term. The compound level of the SEEM can also be identified in this term.

Furthermore a perturbational correction term is shown in Figure 6-29-d and this term is introduced because of the choice of the solution procedure. The system parameters can be identified by the SEEM links colours; the zeroth perturbation order solution links are in pink, the multiple scales related quantities are in blue, and the natural frequencies are in green.

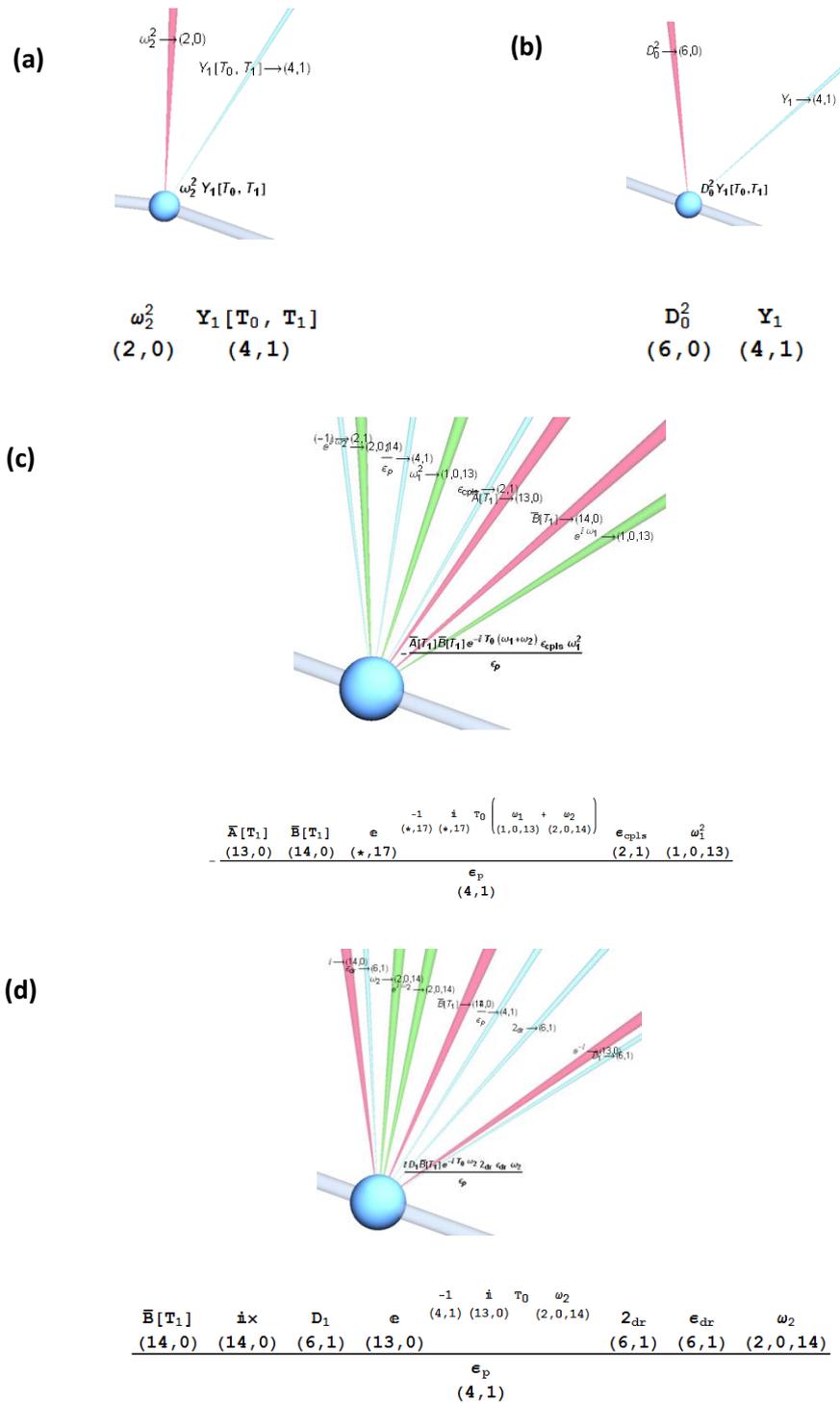


Figure 6-29 A detailed investigation of the encoding information for eqn (17); where, (a) is stiffness, (b) is inertia,(c) is a coupling term, and (d) is a perturbational correction term.

The first resonance condition is introduced in eqn (18). In this equation  $\sigma_1$  is the detuning parameter. As it is discussed in Chapter 3, in order to guarantee a valid solution procedure the small parameter within the detuning must be numerically and symbolically the same as the

perturbation parameter. The overall visualisation graph for this equation is provided in Figure 6-30.

$$\Omega - \omega_1 = \varepsilon_p \sigma_1 \tag{eqn (18)}$$

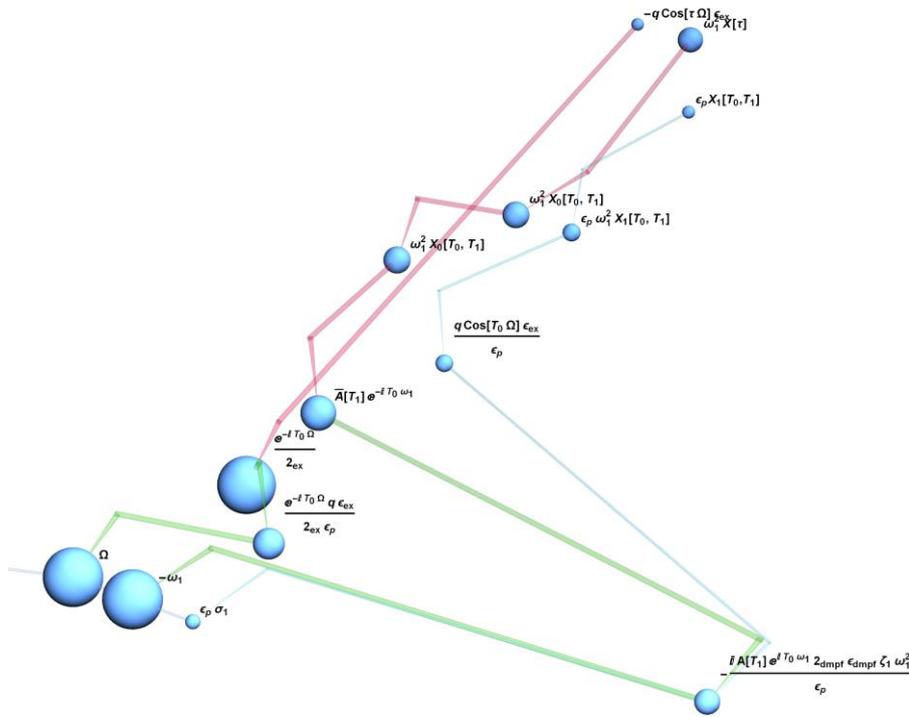
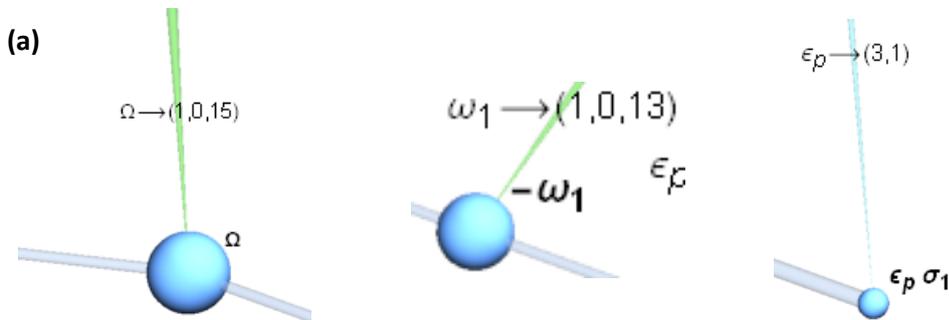


Figure 6-30 Overall visualisation graph for eqn (18).

Detailed information about the SEEM information and the *Blueprint* visualisation links for all the terms is provided in Figure 6-31. Both the external excitation frequency (a) and the linear natural frequency (b) for the primary beam have a green visualisation link. As expected the small detuning term has a blue (first perturbation order) visualisation link.



$$\begin{matrix} \Omega & - & \omega_1 & \epsilon_p & \sigma_1 \\ (1, 0, 15) & & (1, 0, 13) & (3, 1) & (18, 1) \end{matrix}$$

Figure 6-31 A detailed investigation of the encoding information for eqn (18); where, (a) is the external excitation frequency, (b) is the natural frequency for the primary beam, (c) is the detuning parameter.

The second resonance condition is given in eqn (19) where  $\sigma_2$  is the detuning parameter for this equation. The PPR index for number 2 is introduced by the user so as to track this number in the other stages of the analysis.

$$\omega_1 - 2\omega_2 = -2_{PPR} \epsilon_p \sigma_2 \tag{eqn (19)}$$

Figure 6-32 shows the overall *Blueprint* visualisation graph of the second resonance condition. In this figure the sources of two natural frequencies are linked to the equations of motion and highlighted in light blue. Afterwards the colour of the links changes to green as they appeared as a part of the solution of the zeroth-order perturbation equation, this region is highlighted in orange. There are two transitional areas in this Figure, which are shown by arrows, for which no particular modification has been introduced to these quantities. The SF representative of the  $\omega_1$  is larger than  $-2\omega_2$ .  $-2$  is introduced during the solution procedure and does not have an explicit physical meaning. Finally, the detuning parameter has the smallest SF sphere.

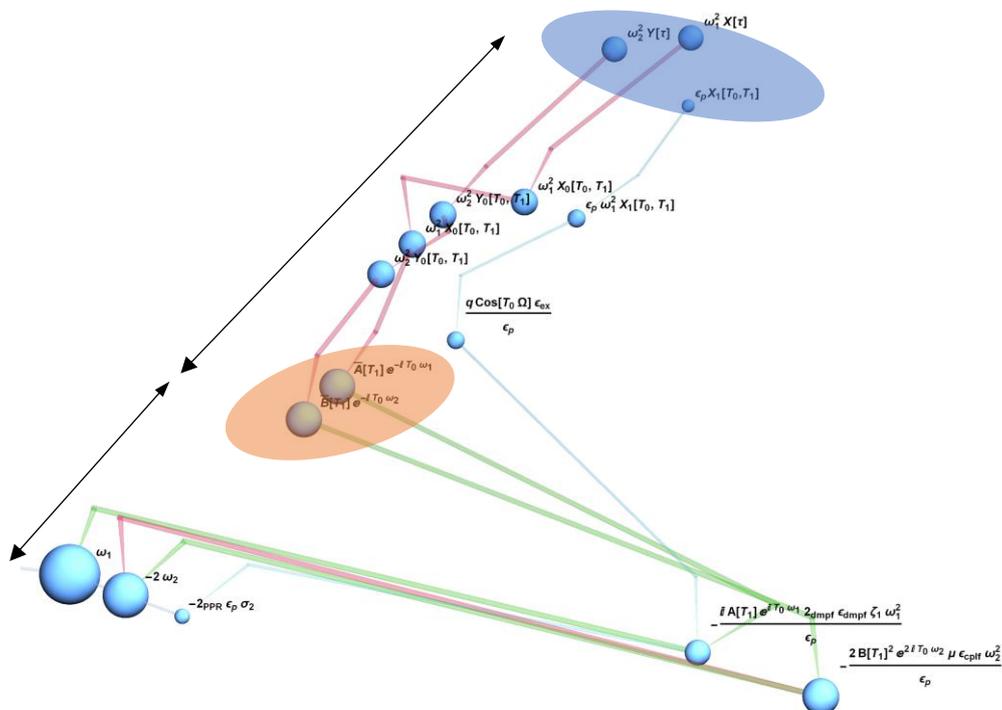


Figure 6-32 Overall visualisation graph for eqn (19), where, the equations of motion are highlighted in blue, the solutions of the zeroth-order perturbation equations are highlighted in orange, and the transitional area is shown with arrows.

The details of the SEEM encoding information and the visualisation for each term are given in Figure 6-33. The natural frequency of the primary beam is shown in part (a) of this Figure and the highlighted stages of the analysis in Figure 6-32 are also visible within the encoding vector. The terms related to the natural frequency of the secondary beam and the detuning parameter are provided in parts (b) and (c), respectively.

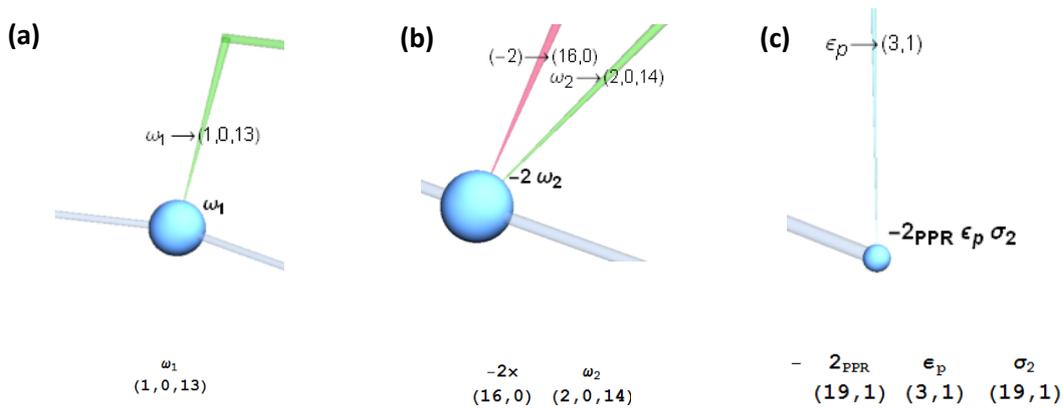


Figure 6-33 A detailed investigation of the encoding information for eqn (18); where, (a) is the natural frequency for the primary beam, (b) is the natural frequency for the secondary beam, (c) is the detuning parameter.

The secular terms from eqn (16) are identified and selected and then set to zero. The modulation equation for the primary beam is given by the following:

$$\frac{e^{iT_0\Omega - iT_0\omega_1} q \epsilon_{ex}}{2\epsilon_x \epsilon_p} - \frac{iD_1 A 2_{dr} \epsilon_{dr} \omega_1}{\epsilon_p} - \frac{iA 2_{dmpf} \epsilon_{dmpf} \zeta_1 \omega_1^2}{\epsilon_p} - \frac{2B^2 e^{-iT_0\omega_1 + 2iT_0\omega_2} \mu \epsilon_{cplf} \omega_2^2}{\epsilon_p} = 0 \quad \text{eqn (20)}$$

The overall *Blueprint* visualisation for this equation graph is provided below:

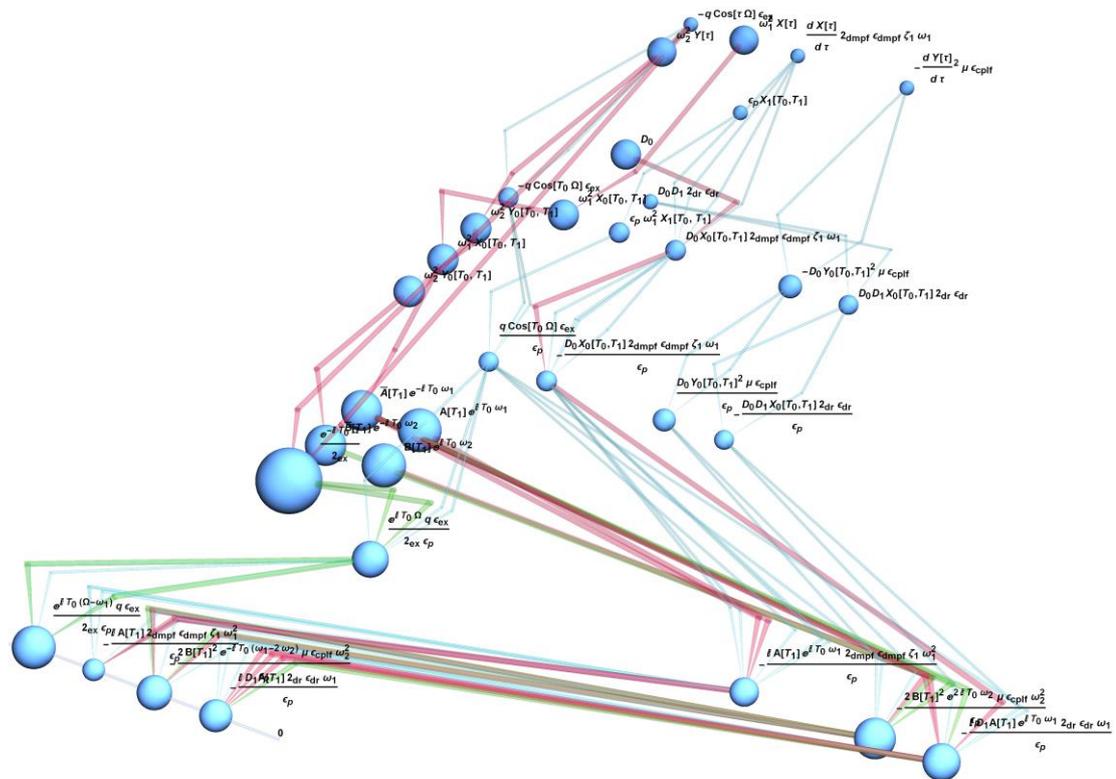


Figure 6-34 Overall visualisation graph for eqn (20).

The details of the SEEM encoding information and the visualisation links for selected terms are given in Figure 6-34. An interaction between the external excitation and the natural frequency of the primary beam is shown in Figure 6-34 -a, noting that there are two green and three blue links in this term.

There is no visualisation link provided for  $\omega_1$ , as the method conventionally requires that the terms in the modulation equations are multiplied by structures such as  $e^{-\omega_1 T_0}$  in order to get them into more manageable forms for later analysis. This  $\omega_1$  does not have the same source as that from the equation of motion, however the numerical value of these two instances of this quantity must obviously be the same.

Figure 6-35-b represents the damping related term. Both the automatically generated SEEM information and the visualisation links provide detailed information about the source and the history of each quantity. There is no influence from the secondary system in this term.

(b)

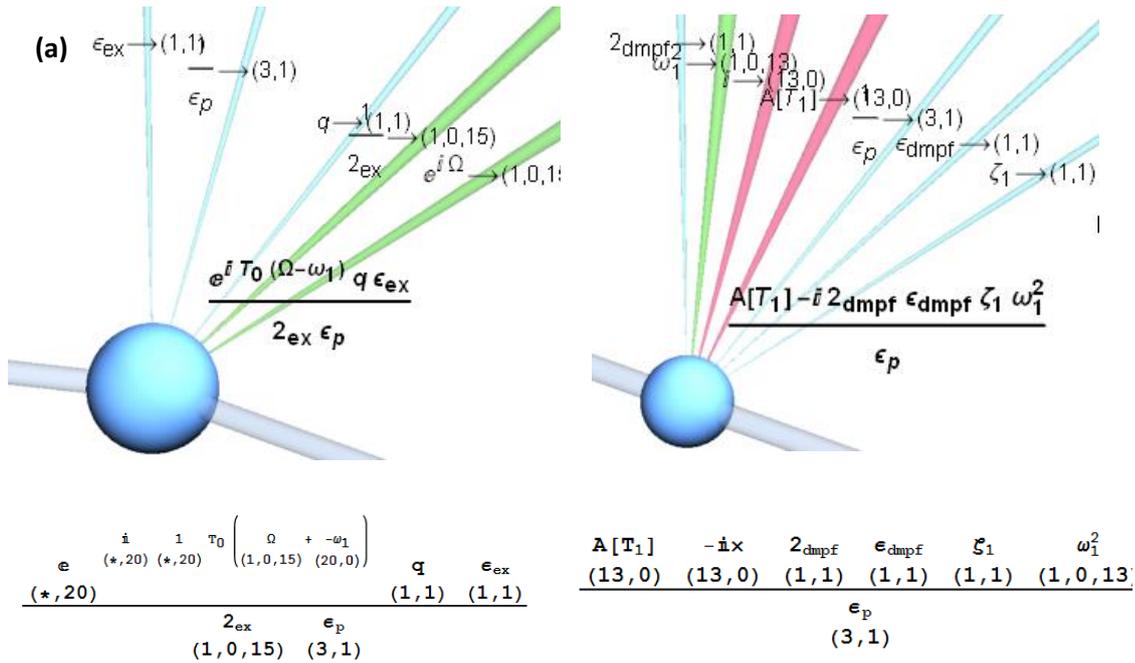


Figure 6-35 A detailed investigation of the encoding information for eqn (20), where, (a) is the excitation related term, and (b) is a damping related term

The secular terms from eqn (16) are identified and selected and then set to zero and the modulation equation for the primary beam is subsequently given as follows:

$$\frac{e^{-iT_0\Omega+iT_0\omega_1} q \epsilon_{ex}}{2\epsilon_{ex}\epsilon_p} + \frac{i\bar{A}2_{dmpf}\epsilon_{dmpf}\zeta_1\omega_1^2}{\epsilon_p} - \frac{2\bar{B}^2 e^{iT_0\omega_1-2iT_0\omega_2} \mu\epsilon_{cplf}\omega_2^2}{\epsilon_p} + \frac{iD_1\bar{A}2_{dr}\epsilon_{dr}\omega_1}{\epsilon_p} = 0 \tag{eqn (21)}$$

An overview of the second modulation equation is shown in Figure 6-36. The fundamental terms are highlighted in blue. Mathematical interactions between the fundamental equations and the solutions method can be identified in between.

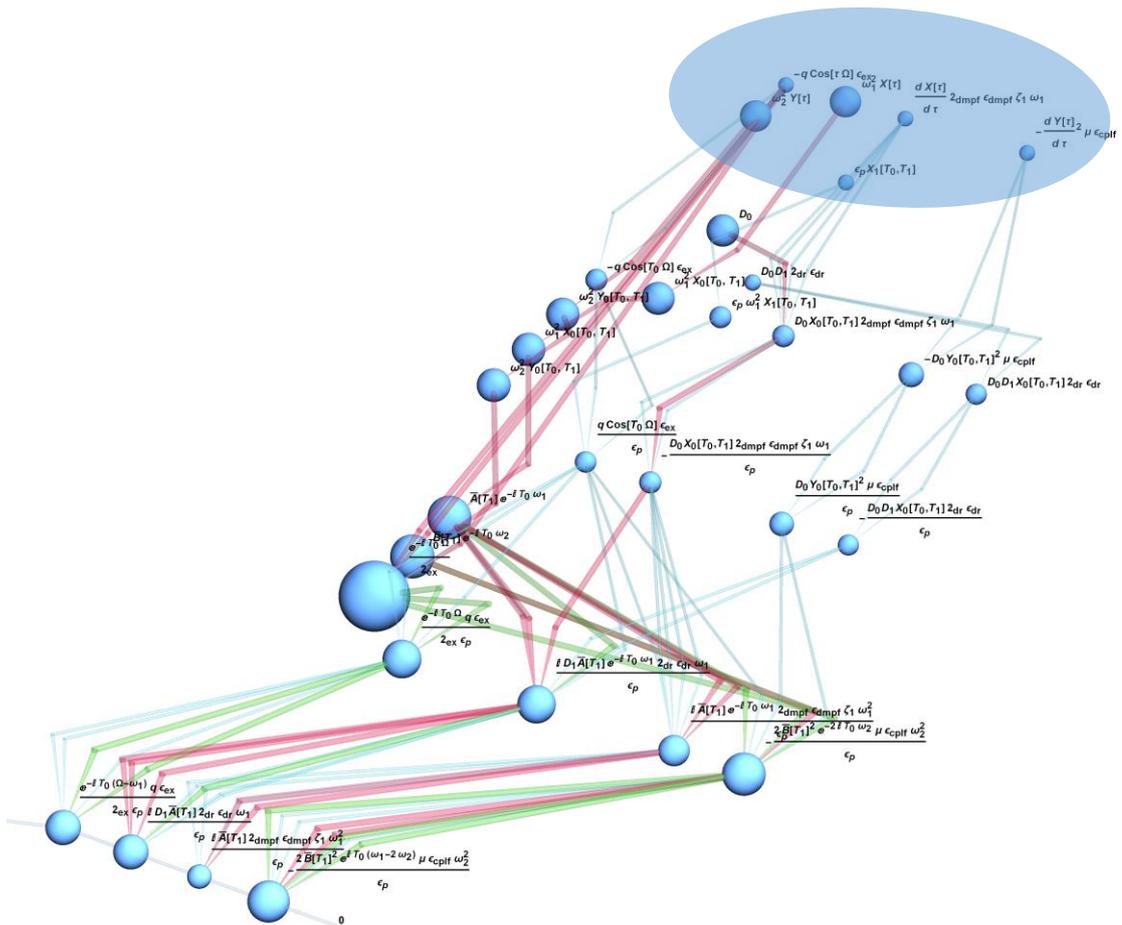
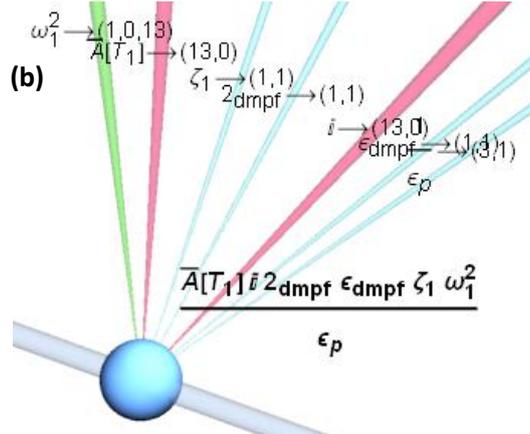
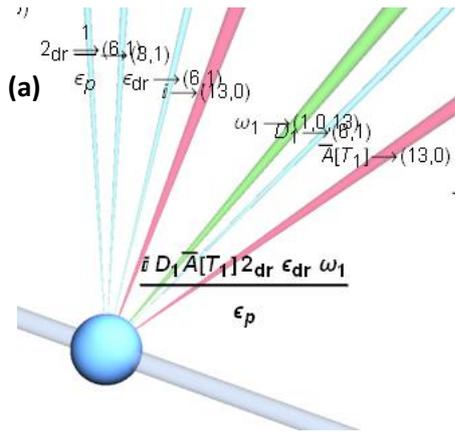


Figure 6-36 Overall visualisation graph for eqn (21).

Figure 6-37 shows the detailed investigation for some of the selected terms in this equation. Figure 6-37-a is a perturbational correction term that appears due to the multiple scales method expansion. There is no influence from the secondary beam in this equation. The damping related term is shown in Figure 6-37-b and the structure of this term is mainly based on the fundamental equations (1) and (3). There is no influence from the secondary beam in this equation.

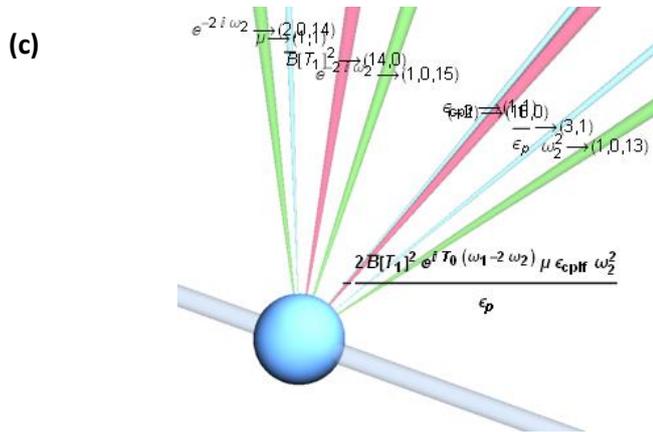
In addition, the coupling term is shown in Figure 6-37-c. The number of dense links is larger than for the slim first perturbation order links. In this term there are links to both the primary and the secondary beams equations of motion. No connecting link for  $\omega_2$  is defined, as both sides of this equation was multiplied by  $e^{-\omega_2 T_0}$ . For this instance  $\omega_2$  does not have the same encoding source as that in the equation of motion.

Finally, Figure 6-37-d shows the interaction between the external excitation frequency and the first natural frequency of the primary beam. The links are either in dense green or weak blue.

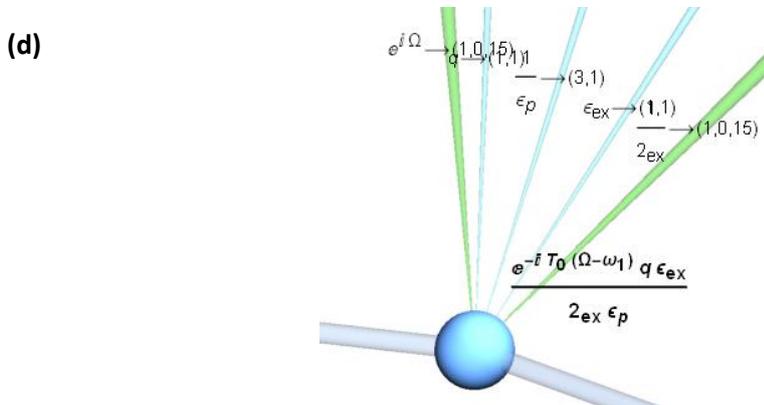


$\bar{A}[T_1]$	$i \times$	$D_1$	$2_{dr}$	$\epsilon_{dr}$	$\omega_1$
(13,0)	(13,0)	(6,1)	(6,1)	(6,1)	(1,0,13)
$\epsilon_p$					
(3,1)					

$\bar{A}[T_1]$	$i \times$	$2_{dmpf}$	$\epsilon_{dmpf}$	$\zeta_1$	$\omega_1^2$
(13,0)	(13,0)	(1,1)	(1,1)	(1,1)	(1,0,13)
$\epsilon_p$					
(3,1)					



$-2 \times$	$B[T_1]^2$	$e^{-1}$	$i$	$T_0$	$\left( \begin{matrix} \omega_1 & +2 \\ (*,20) & (2,0,14) \end{matrix} \right)$	$\mu$	$\epsilon_{cpif}$	$\omega_2^2$
(16,0)	(14,0)	(* ,20)				(1,1)	(1,1)	(1,0,13)
$\epsilon_p$								
(3,1)								



$$\begin{array}{c}
 \begin{array}{c}
 \mathbf{e} \\
 (*, 21)
 \end{array}
 \begin{array}{c}
 -1 \\
 (*, 21)
 \end{array}
 \begin{array}{c}
 i \\
 (*, 21)
 \end{array}
 T_0 \left( \begin{array}{c}
 \Omega + -\omega_1 \\
 (1, 0, 15) \quad (21, 0)
 \end{array} \right)
 \begin{array}{c}
 \mathbf{q} \\
 (1, 1)
 \end{array}
 \begin{array}{c}
 \mathbf{e}_{ex} \\
 (1, 1)
 \end{array} \\
 \hline
 \begin{array}{c}
 2_{ex} \\
 (1, 0, 15)
 \end{array}
 \begin{array}{c}
 \mathbf{e}_p \\
 (3, 1)
 \end{array}
 \end{array}$$

Figure 6-37 A detailed investigation of the encoding information for eqn (21), where, (a) is a perturbational correction term, (b) is a damping related term, (c) is a coupling term, and (d) is an external excitation related term

The secular terms from eqn (17) are identified and selected and then set to zero. The modulation equation for the secondary beam is given as follows:

$$\frac{A\bar{B}e^{iT_0(\omega_1-2\omega_2)}\epsilon_{cpls}\omega_1^2}{\epsilon_p} - \frac{iD_1B2_{dr}\epsilon_{dr}\omega_2}{\epsilon_p} - \frac{iB2_{dmps}\epsilon_{dmps}\zeta_2\omega_2^2}{\epsilon_p} = 0 \tag{eqn (22)}$$

An overview of the visualisation graph for the second modulation equation is given in Figure 6-38. The fundamental terms are located at the upstream of this Figure and are highlighted in blue. The solutions of the zeroth-order perturbation equations, for both beams, are highlighted in green. A transitional area, where there is no significant mathematical procedure, is highlighted in orange and shown with an arrow.

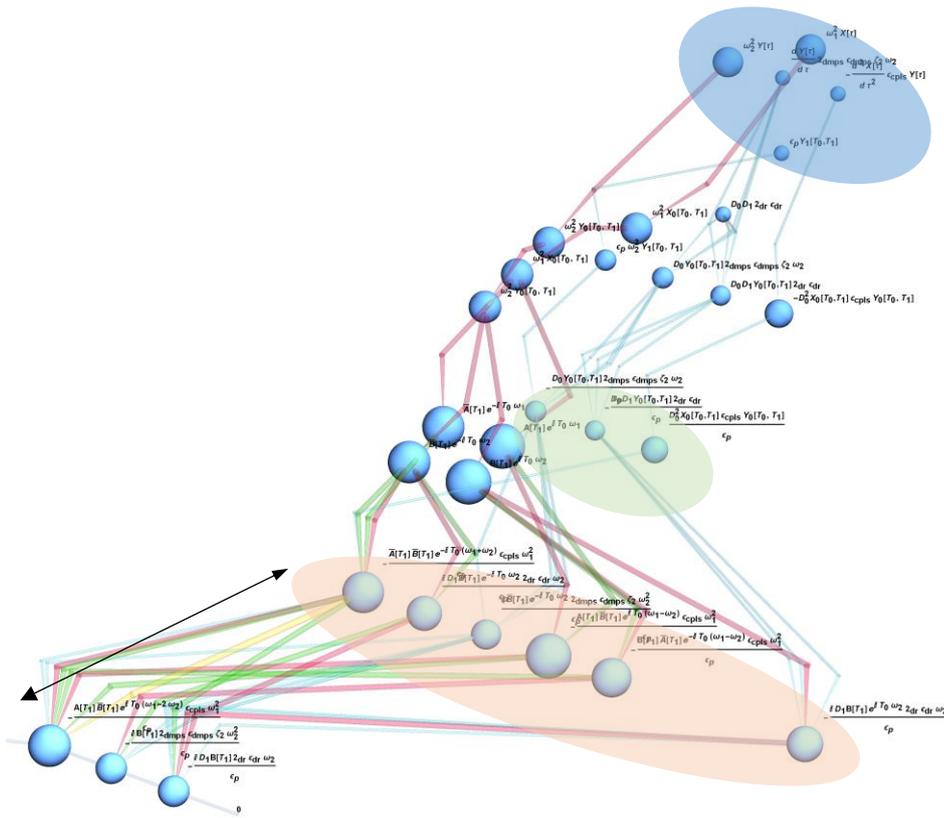


Figure 6-38 Overall visualisation graph for eqn (22), where; fundamental quantities are highlighted in blue, the solution of the zeroth-order perturbation equation is in green, and the transition area is highlighted in orange.

A detailed investigation of the encoding information for eqn (22) is shown in Figure 6-39. The coupling term is shown in Figure 6-39-a and the yellow visualisation link related to the compound level of the SEEM. As both sides of the equation are multiplied by  $e^{-\omega_2 iT_0}$ , the first digit of the SEEM encoding of  $-2$  in the index of the exponential function is addressed to eqn (22).

The damping term for the secondary beam is given in Figure 6-39-b. This term is structured from equations 2, 4, and 14. Finally Figure 6-39-c shows a perturbational correction term which is mainly sourced to equation (6). This term is completely related to the multiple scales method so its structure would potentially alter through modifying the solution procedure.

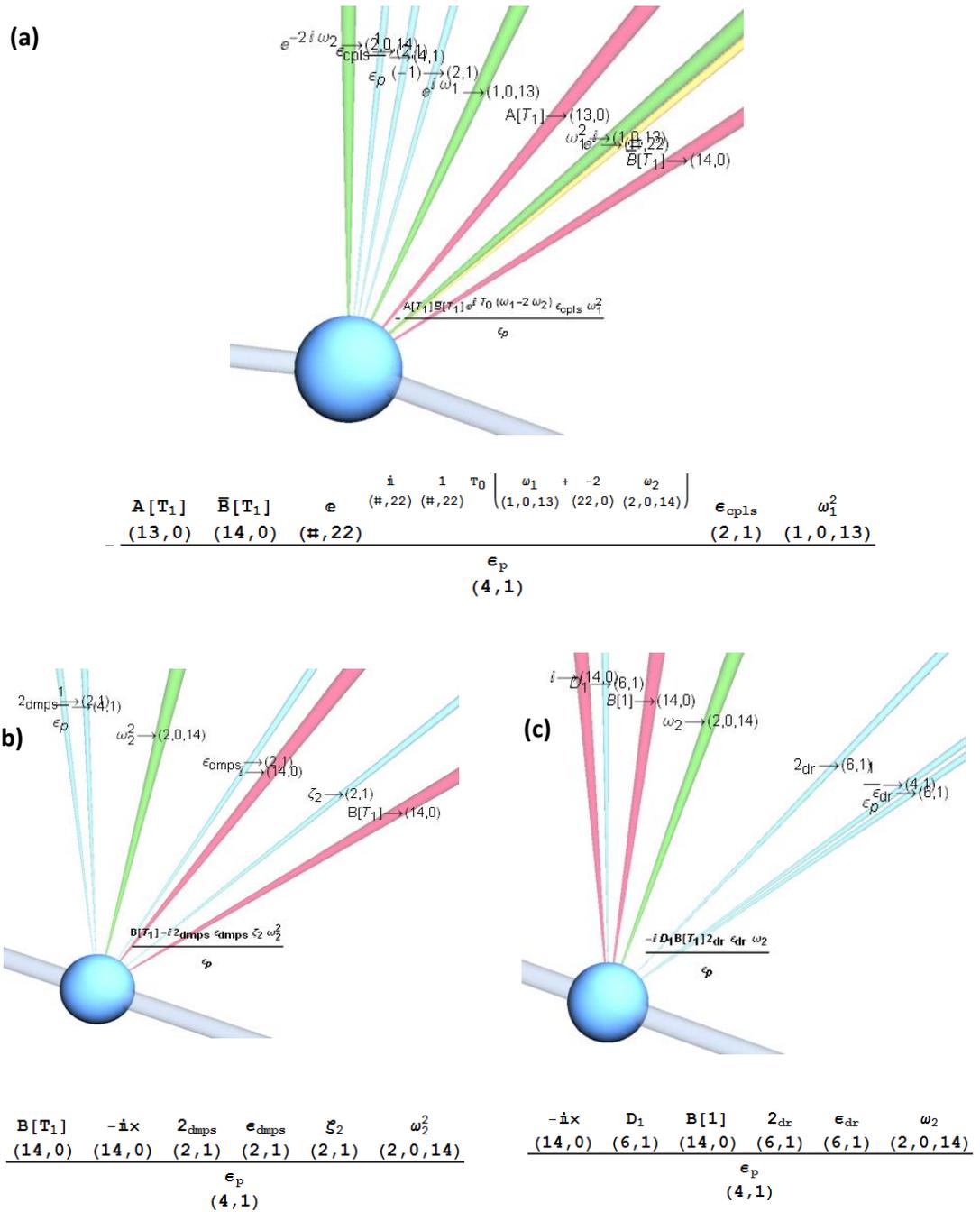


Figure 6-39 A detailed investigation of the encoding information for eqn (22); where, (a) is a coupling term, (b) is a damping term, and (c) is a perturbational correction term

The secular terms from eqn (17) are identified, selected and then set to zero. The modulation equation for the primary beam is given as follows:

$$-\frac{B\bar{A}e^{-iT_0(\omega_1-2\omega_2)}\epsilon_{cpls}\omega_1^2}{\epsilon_p} + \frac{iD_1\bar{B}2_{dr}\epsilon_{dr}\omega_2}{\epsilon_p} + \frac{i\bar{B}2_{dmps}\epsilon_{dmps}\zeta_2\omega_2^2}{\epsilon_p} = 0 \quad \text{eqn (23)}$$

The general visualisation plot for eqn (23) is shown in Figure 6-40. The fundamental terms are highlighted in blue, the solution for the zeroth-order perturbation equations are highlighted in green, and the transitional stage is shown by an arrow. The SF representative of the coupling term is larger than the other terms, whilst the damping term has the smallest SF value.

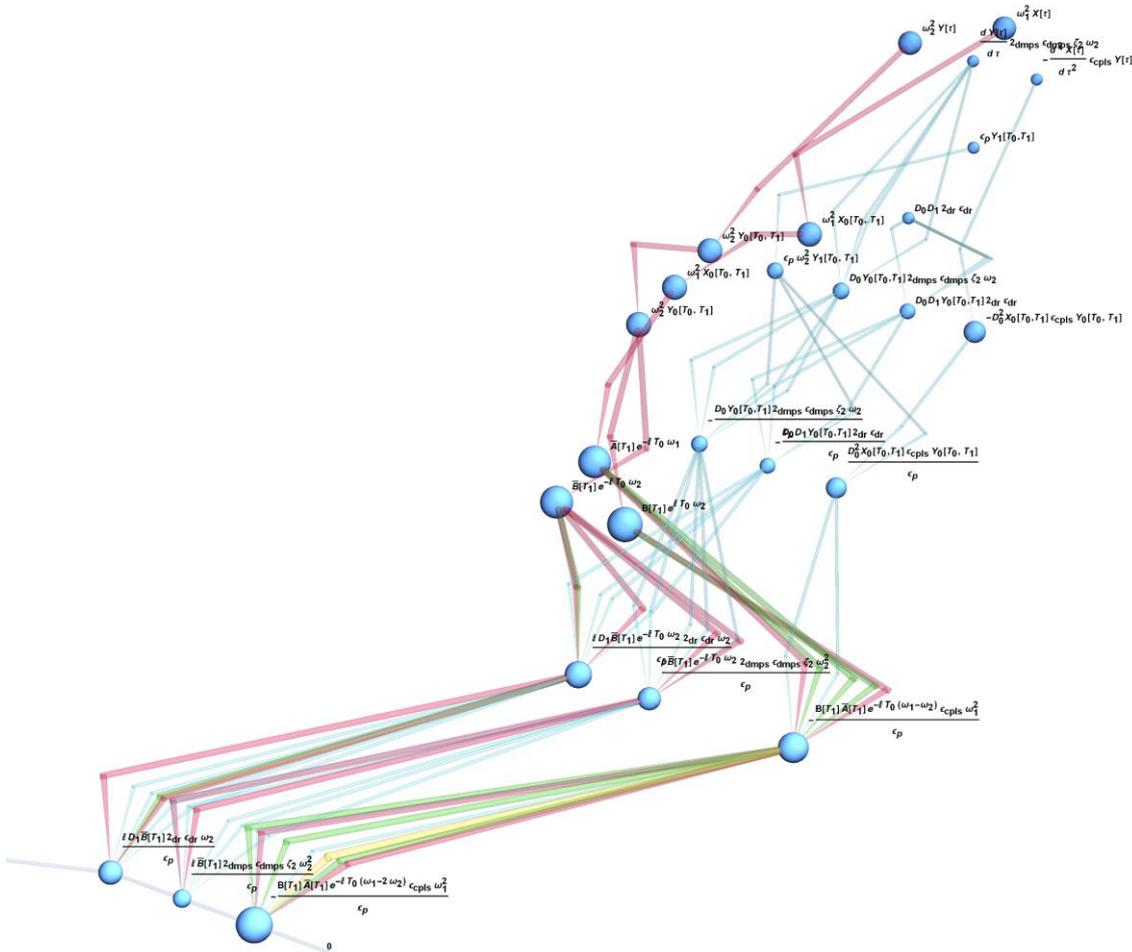


Figure 6-40 Overall visualisation graph for eqn (23), where; fundamental quantities are highlighted in blue, the solution of the zeroth-order perturbation equation is in green, and the transition area is shown with an arrow.

A detailed analysis for each term is given in Figure 6-41. The perturbational correction term is shown in Figure 6-41-a and there are three zeroth perturbation order quantities in this term. These quantities are linked to the dependent variable for the out-of-plane response and the first linear natural frequency of the secondary beam. Changing the solution method type would affect the structure of this term, and hence the appearance of equation (23). Finally the damping term, with three zeroth perturbation order quantities is shown in Figure 6-41-b.

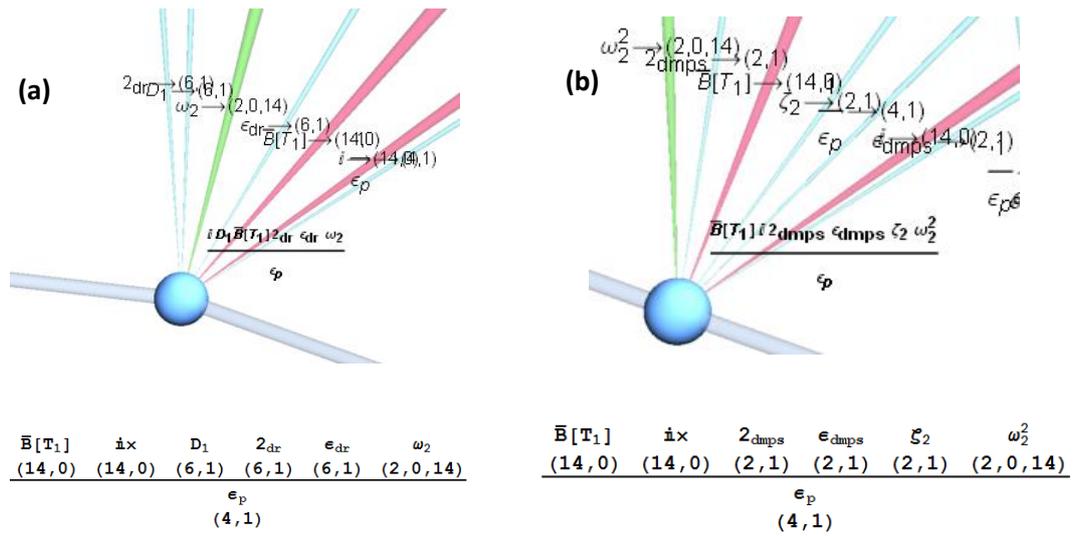


Figure 6-41 A detailed investigation of the encoding information for eqn (23); where, (a) is a perturbational correction term, and (b) is a damping term

Equation (24) is the first order perturbation equation, for the primary beam, after removing the secular terms.

$$D_0^2 X_1 + \omega_1^2 X_1 = \frac{2A\bar{A}\gamma\epsilon_{nsf}\omega_1^2}{\epsilon_p} - \frac{\bar{A}^2 e^{-2iT_0\omega_1}\gamma\epsilon_{nsf}\omega_1^2}{\epsilon_p} - \frac{A^2 e^{2iT_0\omega_1}\gamma\epsilon_{nsf}\omega_1^2}{\epsilon_p} \quad \text{eqn (24)}$$

The overall visualisation graph for this equation is provided below:

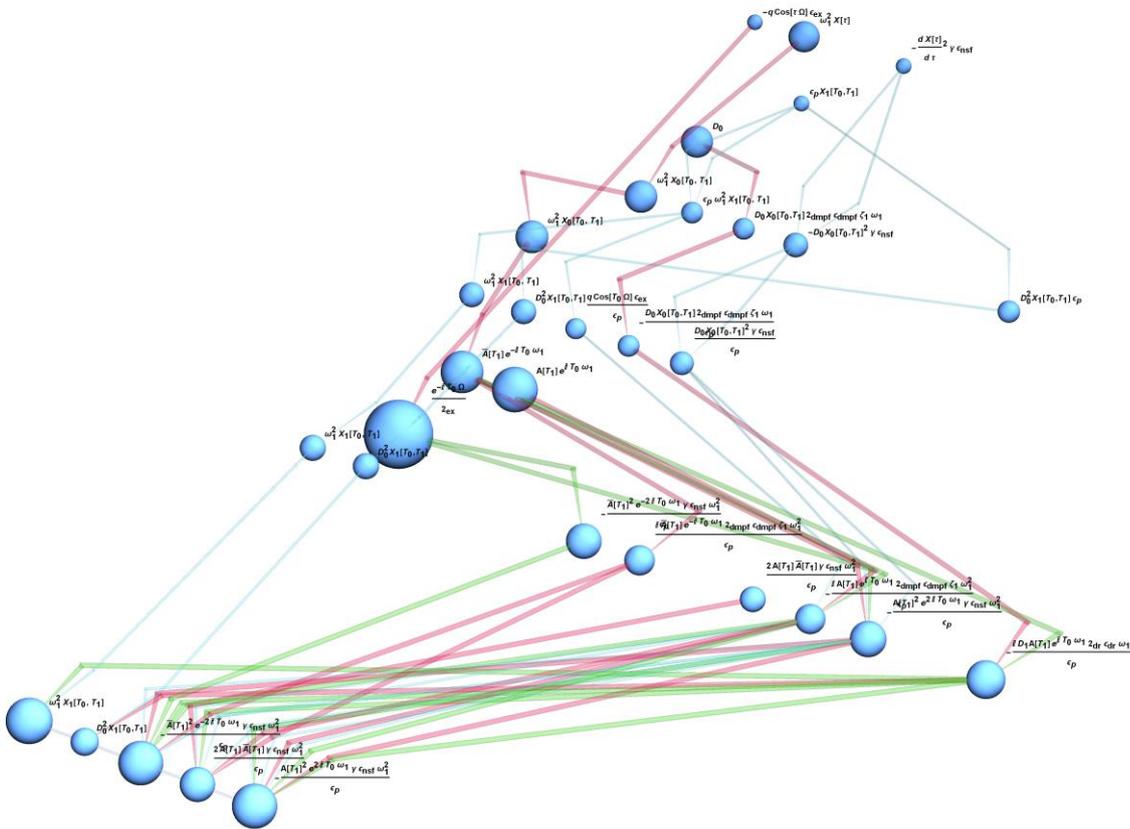
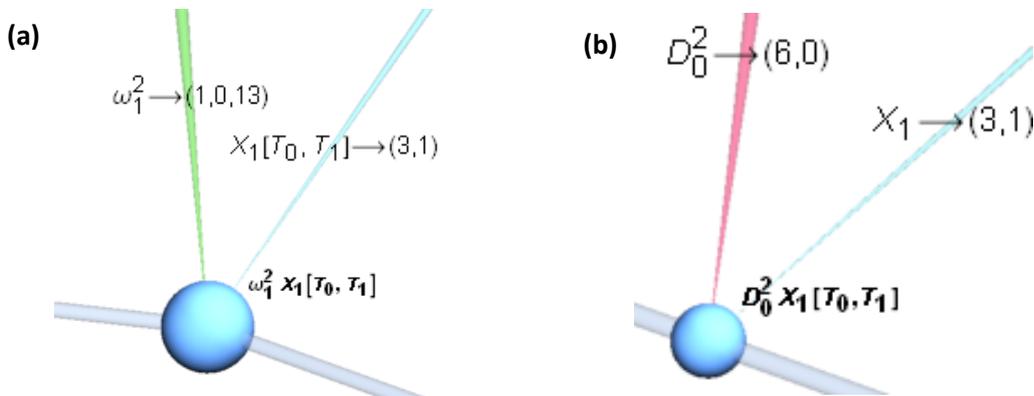


Figure 6-42 Overall visualisation graph for eqn (24).

The details of the SEEM visualisation method are given in Figure 6-43. The stiffness and inertia terms are shown in parts (a) and (b), respectively. Furthermore the nonlinear stiffness related terms are given in parts (c), (d), and (e) of this Figure. There is no contribution from the primary beam in these terms.



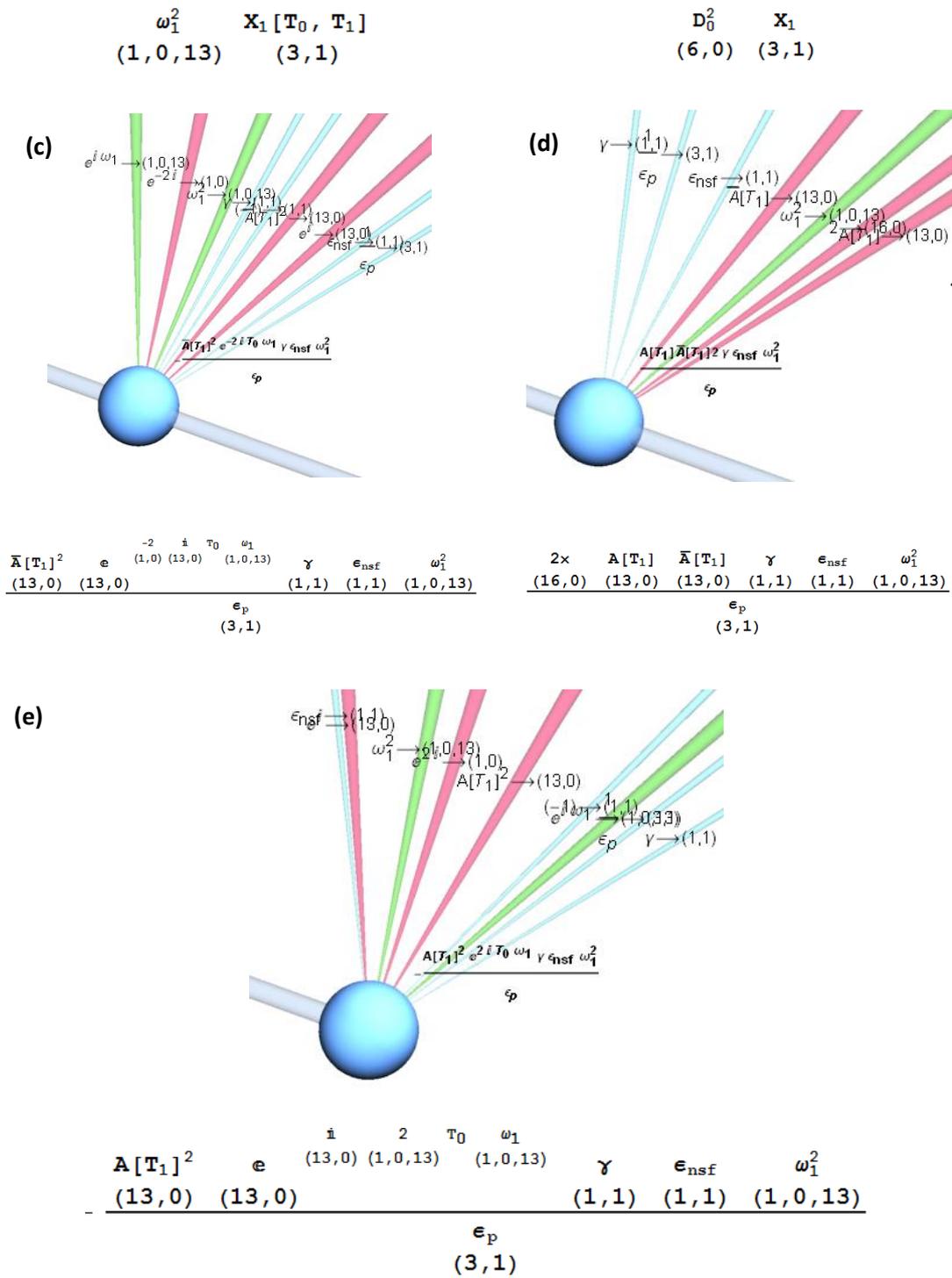


Figure 6-43 A detailed investigation of the encoding information for eqn (24); where, (a) is linear stiffness term, (b) is the inertia term, and the nonlinear stiffness related terms are given in parts (c), (d), and (e).

Eqn (25) is the first order perturbation equation for the secondary beam after removal of the secular terms.

$$D_0^2 Y_1 + \omega_2^2 Y_1 = -\frac{\bar{A}\bar{B}e^{-iT_0\omega_1 - iT_0\omega_2} \varepsilon_{cpls} \omega_1^2}{\varepsilon_p} - \frac{ABe^{iT_0\omega_1 + iT_0\omega_2} \varepsilon_{cpls} \omega_1^2}{\varepsilon_p} \quad \text{eqn (25)}$$

Figure 6-44 shows the overall representation of the sources and links structuring eqn (25). There is a large amount of links on the upstream part of the Figure, which are representing the essential mathematical procedures affecting the structure of this equation. The fundamental quantities are highlighted in blue, the solution of the zeroth-order perturbation equations are highlighted in green, and the transitional area is shown by an arrow.

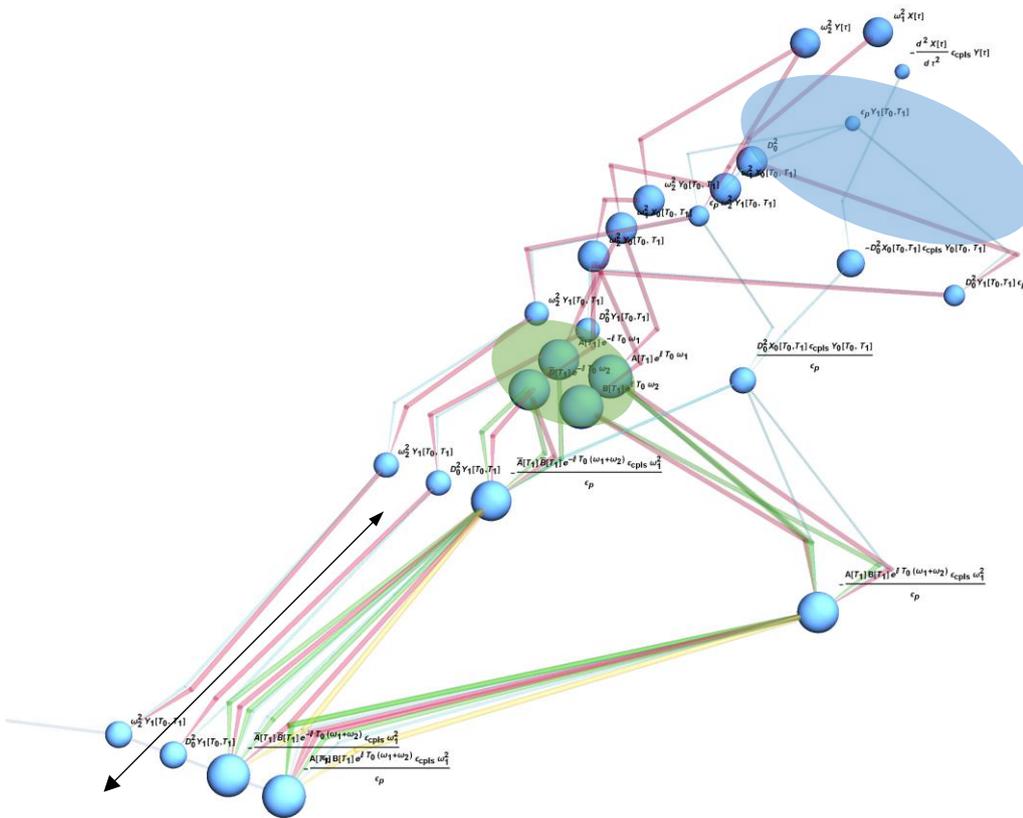


Figure 6-44 Overall visualisation graph for eqn (25), where; fundamental quantities are highlighted in blue, the solution of the zeroth-order perturbation equation is in green, and the transition area is shown with an arrow.

As the details of the SEEM encodings for these terms have already been discussed in Figure 6-27, no further detailed analysis is given for this equation. The multiple scales analysis continues by solving eqn (25), resulting in:



possible to notice that the third level of the SEEM is considered for all the quantities in this equation.

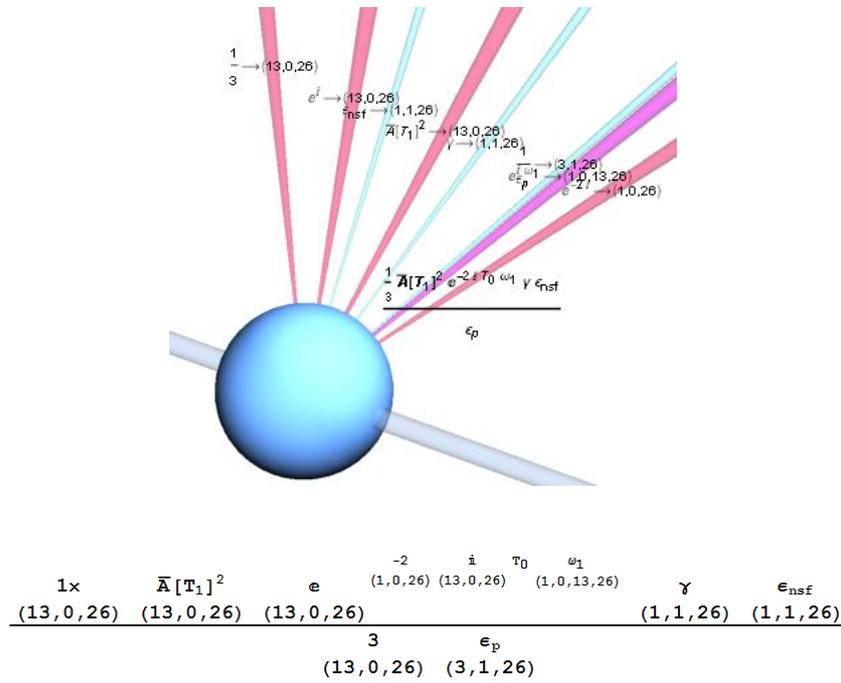


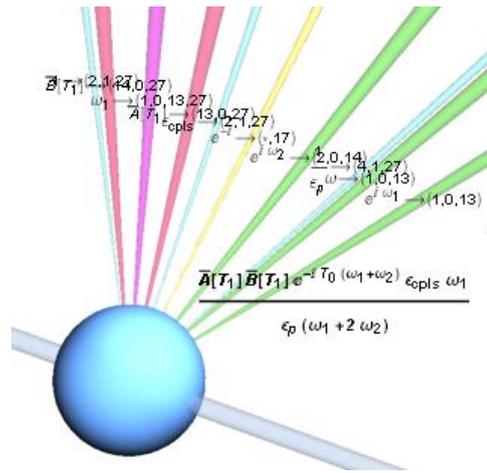
Figure 6-46 Details of the SEEM encodings information for a term in eqn (26).

The same approach for the secondary beam has been carried out, and the solution of the first order perturbation equation is given as follows:

$$Y_1[T_0, T_1] = \frac{ABe^{iT_0(\omega_1+\omega_2)}\epsilon_{cpls}\omega_1}{\epsilon_p(\omega_1 + 2\omega_2)} + \frac{e^{-iT_0(\omega_1+\omega_2)}\bar{A}\bar{B}\epsilon_{cpls}\omega_1}{\epsilon_p(\omega_1 + 2\omega_2)} \quad \text{eqn (27)}$$

The overall visualisation graph for this equation is shown in Figure 6-47. The explicit involvements from the primary beam can be identified in this graph. The fundamental terms are highlighted in blue. The coupling term and the first order perturbation terms are clearly visible in the upstream part of this Figure. The solutions of the zeroth-order perturbation equation for both the primary and secondary beams are also visible in this graph, and are highlighted in green. There is long transitional area for which no mathematical operations related to this equation are undertaken, and which is highlighted once again by an arrow. As expected again, the SF representation for both the terms are approximately the same.





$$\frac{\bar{A}[T_1] \quad \bar{B}[T_1] \quad e^{-i\tau_0(\omega_1+\omega_2)} \quad \epsilon_{opls} \quad \omega_1}{(13,0,27) \quad (14,0,27) \quad (*,17) \quad (2,1,27) \quad (1,0,13)}$$

$$\epsilon_p \left( \begin{matrix} \omega_1 & + & 2 & \omega_2 \\ (1,0,13) & & (2,1,27) & (2,0,14) \end{matrix} \right)$$

Figure 6-48 Detail of the SEEM encodings information for a term in eqn (27).

The polar form of the complex amplitude of  $A$  is given in eqn (28); where,  $\alpha$  is the phase angle and  $a$  is the amplitude of response of the primary beam. The subscript  $A$  for the numeral 2 is used by choice of the user. It must be noted that  $2_A$  obviously has the same numerical value as the general number 2. Additionally, the polar form of the complex amplitude of  $B$  is given in eqn (29); where,  $\beta$  is the phase angle and  $b$  is the amplitude of the response for the secondary beam. As in the above,  $B$  is used as the subscript for number 2 for identification purposes, and here too the numerical value must be identical to that of the general number 2.

$$A = \frac{e^{i\alpha[T_1]}a[T_1]}{2_A} \tag{eqn (28)}$$

$$B = \frac{e^{i\beta[T_1]}b[T_1]}{2_B} \tag{eqn (29)}$$

The encoding information for these equations is given in Figure 6-49 where the first level of the SEEM is applied.

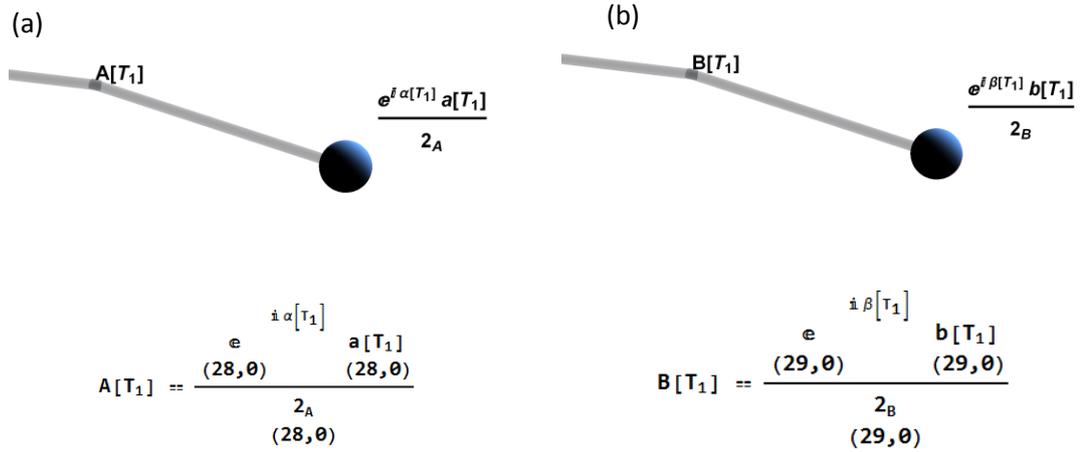


Figure 6-49 The first level of the encoding method is defined for eqn (28) and (29).

The perturbation expansion parameter can be explicitly defined by relating the time scales, as given by eqn (30).

$$\varepsilon_p = \frac{T_1}{T_0} \quad \text{eqn (30)}$$

The resonance conditions, eqn (18) and (19), and the complex amplitudes, eqn (28) and (29), are substituted into the modulation equation for the primary beam (20), resulting in:

$$\begin{aligned} & \frac{q \cos[T_1 \sigma_1 - \alpha[T_1]] \varepsilon_{ex}}{2_{ex} \varepsilon_p} + \frac{i q \sin[T_1 \sigma_1 - \alpha[T_1]] \varepsilon_{ex}}{2_{ex} \varepsilon_p} - \frac{i a[T_1] 2_{dmpf} \varepsilon_{dmpf} \zeta_1 \omega_1^2}{2_A \varepsilon_p} \\ & - \frac{2 \mu b[T_1]^2 \cos[2T_1 \sigma_2 - \alpha[T_1] + 2\beta[T_1]] \varepsilon_{cplf} \omega_2^2}{2_B^2 \varepsilon_p} \\ & - \frac{2 i \mu b[T_1]^2 \sin[2T_1 \sigma_2 - \alpha[T_1] + 2\beta[T_1]] \varepsilon_{cplf} \omega_2^2}{2_B^2 \varepsilon_p} - \frac{i 2_{dr} \varepsilon_{dr} \omega_1 \alpha'[T_1]}{2_A \varepsilon_p} \\ & + \frac{\alpha[T_1] 2_{dr} \varepsilon_{dr} \omega_1 \alpha'[T_1]}{2_A \varepsilon_p} = 0 \end{aligned} \quad \text{eqn (31)}$$

The overall visualisation graph for this equation is shown in Figure 6-50. The fundamental quantities are given in the upstream part of the equation, and highlighted in blue. Most of the terms in the equation of motion for the primary beam are involved in this equation.

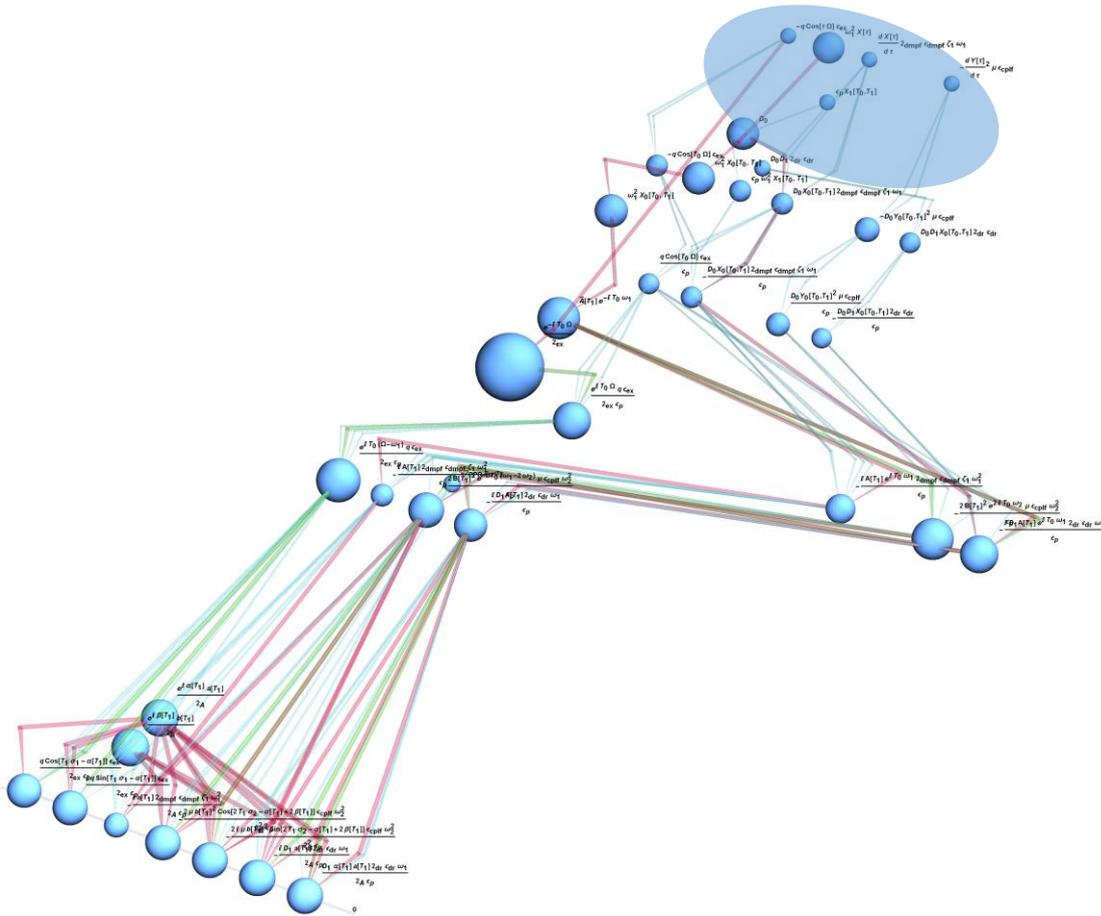
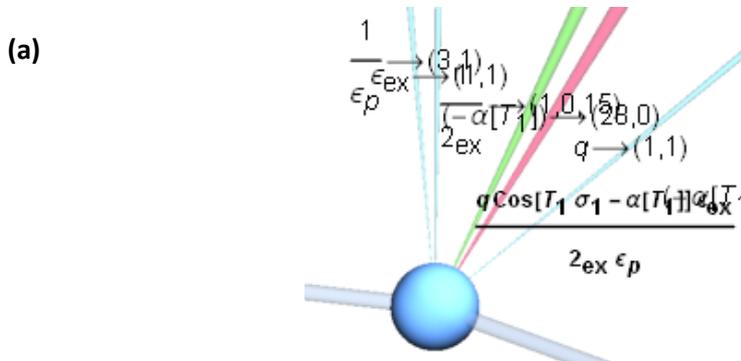


Figure 6-50 The overall visualisation links for eqn (31).

The details of the SEEM information for some selected terms are shown in Figure 6-51. The trigonometrical form of the external excitation term is shown in Figure 6-51-a. The interaction between the phase angle,  $\alpha$ , and the detuning parameter can be noticed in this graph. A perturbational correction term, which is highly dependent on the choice of the solution method, is given in Figure 6-51-b.





$q$ (1,1)	$\sin[T_1]$ (18,1)	$\sigma_1$ (28,0)	$-\alpha[1]$ (28,0)	$\epsilon_{ex}$ (1,1)	$a[T_1]$ (28,0)	$2_{dmpf}$ (1,1)	$\epsilon_{dmpf}$ (1,1)	$\xi_1$ (1,1)	$\omega_1^2$ (1,0,13)
				$2_{ex}$ (1,0,15)			$2_A$ (28,0)	$\epsilon_p$ (3,1)	

Figure 6-52 Details of the SEEM encodings information for a term in eqn (32), where; (a) is the external excitation term, and (b) is the damping term.

The real part of the modulation equation (31) is also taken out and set to zero, resulting in:

$$\frac{q \cos[T_1 \sigma_1 - \alpha[T_1]] \epsilon_{ex}}{2_{ex} \epsilon_p} + \frac{-2\mu b^2 \cos[2T_1 \sigma_2 - \alpha + 2\beta[T_1]] \epsilon_{cplf} \omega_2^2}{2_B^2 \epsilon_p} + \frac{a[T_1] 2_{dr} \epsilon_{dr} \omega_1 \alpha'[T_1]}{2_A \epsilon_p} = 0 \tag{33}$$

The overall visualisation graph is provided below:

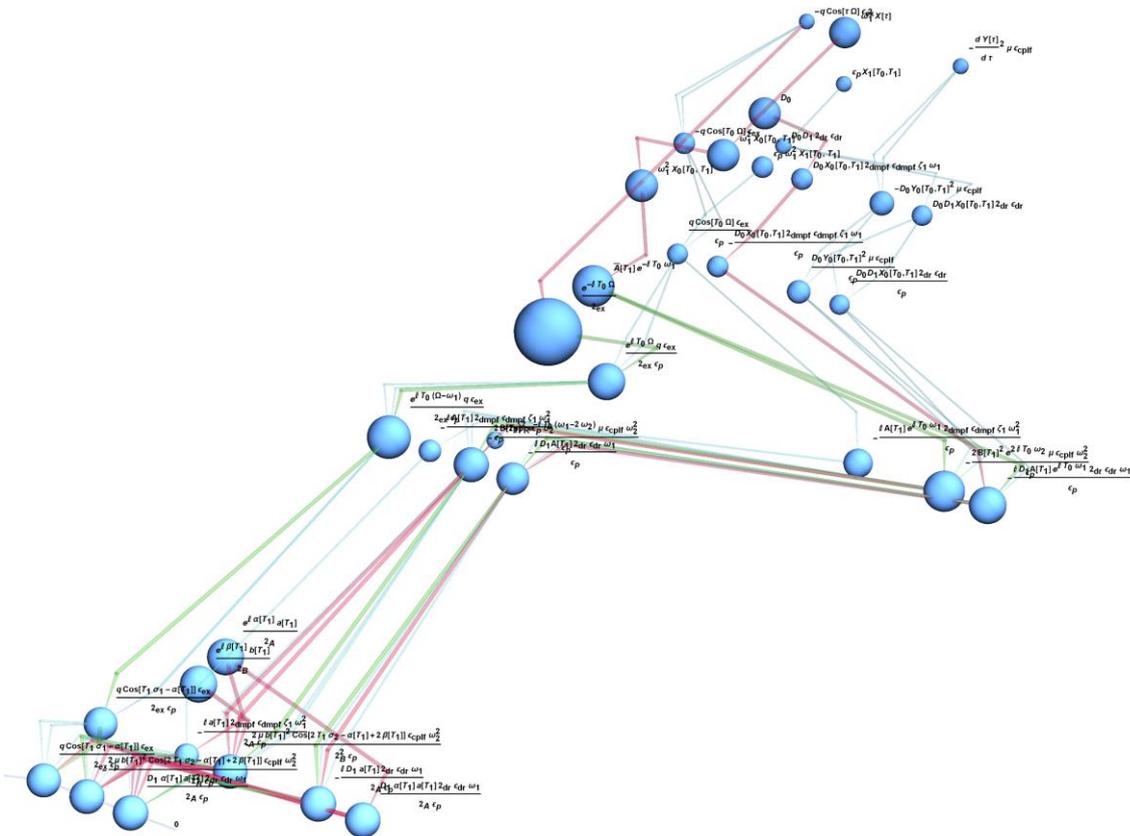
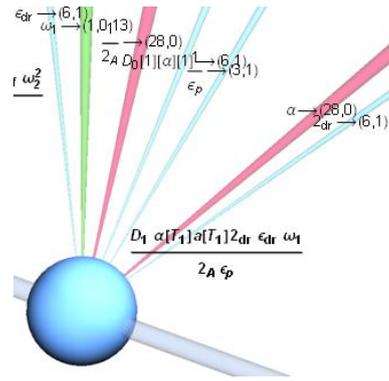


Figure 6-53 The overall visualisation links for eqn (33).

Details of the SEEM encoding for a selected term are provided in Figure 6-54. This term is a perturbational correction term and mainly related to equation (6).



$D_1$	$\alpha$	$a [T_1]$	$2_{dr}$	$\epsilon_{dr}$	$\omega_1$
$(6,1)$	$(28,0)$	$(28,0)$	$(6,1)$	$(6,1)$	$(1,0,13)$
		$2_A$			
		$(28,0)$	$\epsilon_p$		
		$(3,1)$			

Figure 6-54 Details of the SEEM encodings for a selected term in eqn (33).

This analysis can be continued by determining the solutions for the solvability conditions, these being the solutions of the amplitudes and phases for both the primary and secondary beams. This selected part of the analysis shows the capability of the developed features in the SCD solver for extracting the implicit information embedded within the equation structure.