<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Analytical and Computational Methods for Analyzing Feedback Structure in System Dynamics Models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Huang, Jinjing</td>
</tr>
<tr>
<td><strong>Publication Date</strong></td>
<td>2012-07-10</td>
</tr>
<tr>
<td><strong>Item record</strong></td>
<td><a href="http://hdl.handle.net/10379/3466">http://hdl.handle.net/10379/3466</a></td>
</tr>
</tbody>
</table>

Downloaded 2020-10-03T19:11:02Z

Some rights reserved. For more information, please see the item record link above.
Analytical and Computational Methods for Analyzing Feedback Structure in System Dynamics Models

Ph.D Dissertation

Jinjing Huang, B.Eng

Discipline of Information Technology
School of Engineering & Informatics
National University of Ireland, Galway, Ireland

Dr. Jim Duggan (Supervisor)
Prof. Gerard Lyons (Internal Examiner)
Prof. Pål Davidsen (External Examiner)
Abstract

The system dynamics approach is based on the observation that dynamic behaviour arises as a result of endogenous interactions, particularly the interplay of various feedback loops. There are many open questions relating to the relationship between system structure and its behavior, and there are opportunities to add to the range of formal mathematical methodologies which can identify dominant feedback structures that give rise to the system behavior. The objective of this thesis is to propose methodologies that can identify the structural origins of complex behavior, and investigate their application in system dynamics models. Our contributions to the field are:

- First, we have found a drawback of a widely adopted loop selection algorithm named SILS which is required by the eigenvalue elasticity analysis to identify dominant loops. We propose a specific algorithm is to address this problem, and demonstrate its applicability through an individual-oriented example.

- Second, with an increasingly emphasis on the study of eigenvector, an analytic eigenvector analysis method is proposed. As is known that, besides eigenvalue, the eigenvector plays an important role in determining the system behaviour as well. We develop an analytical approach, and suggest considering that both eigenvalue and eigenvector could provide an overall assessment of a structural change over the behaviour of interest.

- Third, for the behavioural method we first propose a new loop deactivation method which enables the behavioural method to operate even when no unique edge exists in a candidate loop. An example is provided to validate this method by comparing its analysis results with that from eigenvalue analysis. The introduction of this loop deactivation method makes the original method more robust and extends its applicability.

- Fourth, we propose an alternative criteria by measuring the sensitivity from different loops. We name this as the variant of the behavioural method. A series of testing values are used for each control variable in multiple simulations, and various trajectories of the variable of interest are obtained. The sensitivity from each loop thus can be measured. We hope by blending the sensitivity measurement, the behavioural method can be examined from another perspective and possibly more insights are gained.

Finally, an extensive software library which automates the behavioural method and further has potential to implement its variant method has been designed and developed.
List of Publications


Acknowledgement

I am grateful to my supervisor, Dr. Jim Duggan for his guidance, encouragement, and continuous support throughout my PhD program.

I would like to thank Prof. Pål Davidsen and Prof. Gerard Lyons for their valuable feedbacks on my dissertation and for serving my dissertation committee.

I would like to convey my special gratitude to Owen Molloy, Niall Madden, Michael Madden, Mohammed Saleh, and Mei Chen who have given me help during my study in National University of Ireland, Galway.

I further thank all colleagues and employees at the Discipline of Information Technology, school of Engineering & Informatics, National University of Ireland, Galway, for being so friendly, helpful and cooperative to me.

I greatly appreciate the three year scholarship offered by Science Foundation of Ireland.

I would like to express my gratitude to my friends in Galway, Enda H, Declan, Enda. B, Garry, Sungim, Lijun, Jiafeng, Wenjing, Fanqing, for making my PhD study so wonderful.

Finally, I deeply thank my parents and my family for their love, support and sacrifice.
## Contents

Abstract iii  
Acknowledgements iv  
List of Figures xii  
List of Tables xiv  

1 Introduction 1  
1.1 Simulation ................................................... 1  
1.2 The Sources of Complexity ................................. 4  
1.3 System Dynamics - A Solution To Complex Feedback Systems ............... 6  
1.4 Research Problems and Objectives of Thesis .................. 12  
1.5 Organization of Thesis .................................. 14  

2 Background and Related Research 17  
2.1 Overview .................................................. 17  
2.2 An Overview of Dynamics ................................ 19  
2.2.1 Behaviours in Linear Systems ....................... 20  
2.2.2 Behaviours in Nonlinear Systems .................... 26  
2.3 Characteristics of Feedback Loops ........................ 31
4.2.3 A Solution to the Inhomogeneous System ........................................ 90
4.3 A Comprehensive Sensitivity Analysis of the State Behaviour .............. 91
4.4 An Analytical Method to Computing the Eigenvector-related Sensitivity . . 93
  4.4.1 Computation of the Right Eigenvector Sensitivity .......................... 93
  4.4.2 Computation of the Left Eigenvector Sensitivity ........................... 95
  4.4.3 Proof of the Analytical Solution to the Eigenvector-related Sensitivity 99
  4.4.4 Analytical Solutions to the Eigenvector-related Sensitivity with respect to Various Structure Components ................................. 102
4.5 A Case Study of the Eigen-based Elasticity Analysis ............................ 108
  4.5.1 Analytical Eigenvalue Elasticity Analysis .................................... 108
  4.5.2 Analytical Eigenvector (Weight) Elasticity Analysis ........................ 109
  4.5.3 The Labour-inventory Model .................................................... 111
4.6 Summary and Discussion ............................................................... 129

5 An Extension of Loop Deactivation in the Behavioural Method ................. 131
  5.1 A Brief Revisit to the Behavioural Method ...................................... 132
  5.2 A New Loop Deactivation Method .................................................. 134
    5.2.1 Deactivating a Loop by Its Unique Consecutive Two Edges ............ 135
  5.3 A Case Study of the Long Wave Model .......................................... 139
    5.3.1 Why This Model? .............................................................. 139
    5.3.2 The Loop Deactivation Method .............................................. 140
    5.3.3 Experiment result ............................................................ 149
  5.4 Conclusions and Future Work ...................................................... 154

6 A Sensitivity Analysis Extension to the Behavioural Method ..................... 155
  6.1 A Variant of the Behavioural Method: A Sensitivity Approach ............... 156
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.1 A Selection of Testing Values for Control Variables</td>
<td>157</td>
</tr>
<tr>
<td>6.1.2 A Measurement to Quantify the Sensitivity</td>
<td>157</td>
</tr>
<tr>
<td>6.2 A Case Study of the Yeast Model</td>
<td>159</td>
</tr>
<tr>
<td>6.3 Conclusions and Future Research</td>
<td>169</td>
</tr>
<tr>
<td>7 Software Design for the Dominant Loop Analysis Methods</td>
<td>171</td>
</tr>
<tr>
<td>7.1 The Design of the Computational Analysis Methods</td>
<td>171</td>
</tr>
<tr>
<td>7.1.1 iSimPlus</td>
<td>172</td>
</tr>
<tr>
<td>7.1.2 Analyser: Implementation of the Behavioural Method</td>
<td>173</td>
</tr>
<tr>
<td>7.2 A Conceptual Design of the Behavioural Method Based Sensitivity Analysis</td>
<td>177</td>
</tr>
<tr>
<td>7.3 Summary</td>
<td>180</td>
</tr>
<tr>
<td>8 Conclusions</td>
<td>181</td>
</tr>
<tr>
<td>8.1 Loop Partition Algorithm</td>
<td>181</td>
</tr>
<tr>
<td>8.2 Eigenvector Analysis Method</td>
<td>182</td>
</tr>
<tr>
<td>8.3 Research Work Related to the Behavioural Method</td>
<td>183</td>
</tr>
<tr>
<td>8.4 Future work</td>
<td>184</td>
</tr>
<tr>
<td>A Conjugated eigenvalues and the oscillatory behaviour</td>
<td>195</td>
</tr>
<tr>
<td>B Computation of eigenvalue sensitivity with respect to the compact link gain</td>
<td>196</td>
</tr>
<tr>
<td>C Computation of Eigenvalue Elasticity with respect to a Loop</td>
<td>197</td>
</tr>
<tr>
<td>D Results on the Fully-connected Goal Diffusion Model (Three Agents)</td>
<td>200</td>
</tr>
<tr>
<td>E Equations of the Labor-inventory Model</td>
<td>202</td>
</tr>
<tr>
<td>F Equations for the Simple Long Wave model</td>
<td>204</td>
</tr>
</tbody>
</table>
List of Figures

1.1 S-shaped growth: structure and behaviour ................................................. 9
1.2 An intendedly rational pricing policy can lead to an inadvertent price war. .... 11
2.1 Eigenvalues with associated behaviour modes ............................................ 23
2.2 Complex conjugated eigenvalue characterized behaviour ............................. 24
2.3 The stock and flow diagram of the floating goals model ............................... 25
2.4 The solution to the floating goals model .................................................... 25
2.5 Three atomic behaviour patterns ............................................................... 26
2.6 Phase trajectories in the neighborhood of various limit cycles, which are de- 
    picted as closed curves: (a) stable (b) unstable (c) and (d) half-stable (from 
    The Great Soviet Encyclopedia (1979)). ................................................. 27
2.7 Van der pol oscillation plot $\ddot{x}+\mu(x^2-1)\dot{x}+x=0$ when $\mu=1.5$ ......... 28
2.8 Basic structure of the production-distribution system considered in the beer 
    distribution game. Orders for beer propagate from right to left, while the brew-
    ery products are delivered in the opposite direction. ................................... 30
2.9 Simulation results from four sectors for $B=0.05$ and $D=0$ from (Mosekilde and 
    Larsen, 1988). The strokes of the oscillatory behaviour are not in step, i.e.,
    aperiodic oscillations, which produces a typical chaotic behaviour. ............. 31
2.10 Open loop gain calculation ........................................................................ 33
2.11 A major loop structure .............................................................................. 35
2.12 Seven behaviour modes of PPM Analysis .................................................. 40
2.13 A multi-feedback loop structure ........................................... 41
2.14 A minor loop ................................................................. 41
2.15 Stock and flow diagram of a logistic growth model ...................... 42
2.16 Behaviour of a logistic growth model ..................................... 43
2.17 Schematic view of the loop elasticity computation ......................... 48

3.1 Schematic view of the loop elasticity computation ......................... 64
3.2 The stock and flow diagram of the goal diffusion model .................... 71
3.3 The simplified graph of the goal diffusion model. Blue squares: GoalAgents, red circles: TargetGoals, and black parallelograms: ChangeInGoals. .... 72
3.4 Illustration of the searching process for a new independent feedback loop. Dotted blue line: current valid search path; black dash line: visited but discarded (still considered as unvisited) edges. ........................................ 77

4.1 Current development of the eigenvector analysis research ................... 84
4.2 Stock and flow diagram of the linear system .................................. 105
4.3 Stock and flow diagram of the labor-inventory model ....................... 112
4.4 State variables behaviour and the decomposed behaviour .................. 114
4.5 Eigenvalue vs. WIPAT ....................................................... 119
4.6 Comparisons of state trajectories on the change of MCT ..................... 125
4.7 Comparisons of state trajectories on the change of SWW ................... 127
4.8 Comparisons of state trajectories on the change of IAT ..................... 128

5.1 The yeast model .................................................................. 133
5.2 A hypothetical model .......................................................... 135
5.3 Loop deactivating process of L2 (a) ......................................... 137
5.4 Loop deactivating process of L2 (b) ......................................... 137
5.5 Loop deactivating process of L2 (c) .................................................. 138
5.6 Stock and flow diagram – the simple long wave model ....................... 139
5.7 The behaviour of Desired Production ................................................. 142
5.8 A cycle of Desired Production ......................................................... 143
5.9 Loop capital expansion (L3) is deactivated ....................................... 145
5.10 Highlight of loop economic growth (L9) in the reference model ........... 146
5.11 Loop economic growth (L9) is deactivated ....................................... 146
5.12 Highlight of loop order fulfillment (L11) in the reference model ........... 147
5.13 Loop order fulfillment (L11) is deactivated ....................................... 147
5.14 Highlight of loop backlog expansion (L14) in the reference model ........ 148
5.15 Loop backlog expansion (L14) is deactivated .................................... 149
5.16 Atomic behaviour patterns in phase I .............................................. 150
5.17 Atomic behaviour patterns in phase II ........................................... 151
5.18 Atomic behaviour patterns in phase III ........................................... 151
5.19 Atomic behaviour patterns in phase IV ........................................... 152
5.20 Atomic behaviour pattern of L1 deactivated in phase IV ..................... 152
5.21 Atomic behaviour patterns in phase V ............................................ 153

6.1 The stock-flow diagram for the yeast model ..................................... 159
6.2 Behavior of Cells over the simulation time ........................................ 161
6.3 Behavior of variables in the reference model ..................................... 162
6.4 Sensitivity analysis in phase I ......................................................... 163
6.5 Sensitivity analysis – standard variance in phase I ............................... 164
6.6 Sensitivity analysis in phase II ......................................................... 165
6.7 Sensitivity analysis – standard variance in phase II ............................. 165

xi
6.8 Sensitivity analysis in phase III ........................................ 166
6.9 Sensitivity analysis in phase III–standard variance .................. 167
6.10 Sensitivity analysis on edges in phase IV ............................. 168
6.11 Sensitivity analysis in phase IV–standard variance .................. 168
6.12 Feedback loop analysis with the default values ...................... 169
7.1 The overview architecture of the software design ..................... 172
7.2 The architecture design of iSimPlus .................................... 174
7.3 The structured key components of the Analyser ....................... 175
7.4 The conceptual design of the behavioural method based sensitivity analysis 178
# List of Tables

2.1 Summary of structural dominance analysis approaches .................. 61

3.1 Two independent feedback loop sets in the goal diffusion model ........ 72

3.2 Agent population vs. independent loop sets in a fully connected network ... 80

4.1 The eigenvalues and eigenvectors of the linear system ..................... 89

4.2 Eigenvector-related sensitivity and eigenvalue sensitivity of 1st mode .... 106

4.3 Various sensitivities with respect to pathway gains of 1st mode .......... 107

4.4 Sensitivity with respect to causal link gains of 1st mode ................ 107

4.5 System parameters with initials and values ............................... 112

4.6 Eigenvalues, right eigenvectors and left eigenvectors of the labor-inventory model ........................................ 113

4.7 Analytical and numerical results of eigenvalue elasticity to parameters .... 117

4.8 Behaviour mode weights for four state variables .......................... 120

4.9 Analytical and numerical results of weight elasticity to parameters associated with 1st and 2nd mode. The leverage points with highest elasticity (regardless of the sign) are marked with a box. .......................... 122

4.10 Analytical and numerical results of weight elasticity to parameters associated with 3rd oscillatory behaviour mode. The leverage points with highest elasticity (regardless of the sign) are marked with a box. ............... 123
4.11 Behaviour mode weights for four state variables after modifying SWW. The left column under each weight shows the reference weight while the right column shows the modified weight. ............................................. 126

5.1 Equations for the yeast model ................................................. 133

5.2 Information for the loop deactivation in the yeast model. \( c \) refers to the constant value of the control variable or the starting point of the unique edge at the starting time of a given phase. .................................................. 134

5.3 The loop matrix for the hypothetical example .............................. 135

5.4 Feedback loops in the SILS – the simple long wave model (a) ........... 141

5.5 Feedback loops in SILS – the simple long wave model (b) ............... 142

5.6 Phases and atomic behaviour patterns of Desired Production ............. 143

5.7 Dominant loop analysis by different approaches ........................... 153

6.1 Equations for the yeast model ................................................. 160

6.2 Time intervals over simulation length and behavioral patterns ............ 160

6.3 Candidate unique edges or control variables in each loop .................. 161

6.4 Boundary for control variables in the yeast model .......................... 161

6.5 Dominant feedback loops comparison with different approaches ........ 167

D.1 Loop elasticity in the 3-agent goal diffusion model. 1: floating goal spiral in A1; 2: state adjustment in A1; 3: interaction between A1 & A2; 4: interaction between A1 & A3; 5: floating goal spiral in A2; 6: state adjustment in A2; 7: interaction between A1, A2 & A3; 8: interaction between A2 & A3; 9: floating goal spiral in A3; 10: state adjustment in A3. ......................... 201

xiv
Chapter 1

Introduction

“Like all systems, the complex system is an interlocking structure of feedback loops. Feedback loop” is the technical term describing the environment around any decision point in a system. The decision leads to a course of action that changes the state of the surrounding system and gives rise to new information on which future decisions are based. This loop structure surrounds all decisions public or private, conscious or unconscious. The processes of man and nature, of psychology and physics, of medicine and engineering all fall within this structure.” — Jay W. Forrester, Urban Dynamics (1969), p.107.

1.1 Simulation

Simulation is a quantitative technique by which the relationships between variables can be formulated and, therefore, their behaviour over time can be generated. It is pervasive in many engineering domains. People would never attempt to send a space ship to the moon without first testing the equipment by constructing prototype models and by computer simulation of the anticipated space trajectories. No company would put a new kind of household appliance or electronic computer into production without first making laboratory tests (Forrester, 1971a). To be more specific, simulation is a computer-dependent technique to solve a quantitative model, which, cannot be solved by normal algebra or calculus. It is worth remarking that modeling is a prerequisite for simulation. This demanding task requires the selection of an appropriate level of detail, problem boundaries, and similar considerations. People are aware
CHAPTER 1. INTRODUCTION

that no model can mimic the exact real subject, comments such as all models are wrong, no
two of these models is exactly the same, and none matches precisely every detail of the real
system can be seen (Shreckengost, 1985; Sterman, 2002). Therefore, a model should always
be created for a purpose. The adequacy of the model can be only judged in terms of that
purpose (Forrester, 1969). Meanwhile, there are a number of generic model validation tests
described (Barlas, 1989, 1994; Forrester and Senge, 1980).

Human beings rely on the simulation more than ever to enhance their understanding, predict
the likely outcomes, and make decisions (Forrester, 1991, 2007a,b; Meadows et al., 1972;
Sterman, 2000). We witness time when the entire society evolves and develops at a faster
pace. In beginning of 20th century, Henry Adams formulated a “Law of Acceleration” to
predict the rapid advance in science and technology (Adams, 1918):

\[
\text{If science were to go on doubling or quadrupling its complexities every ten years, even mathematics would soon succumb. An average mind had succumbed already in 1850; it could no longer understand the problem in 1900.}
\]

This growth in the complexity of our social systems challenges the human mind and calls for
other tools to assist us to overcome our cognitive limitations. Simulation is such a tool that
provides the opportunity to conduct experiments in model design and policy testing. Forrester
defines the complex system as a high-order, multiple-loop, nonlinear feedback structure, and
also points out the nature of complex systems is they have special responses which cause many
of the failures and frustrations experienced in trying to improve their behaviour (Forrester,
1969). This nature of complexity requires the construction of realistic models and simulation
runs to anticipate the possible consequences.

Another reason for simulation is the limited human cognitive capability. While the world
becomes more complex, human evolution is much slower and our cognitive capability devel-
opment does not catch up with its pace. It has been repeatedly demonstrated that the human
mind is not suited for solving complex systems (Kampmann and Sterman, 1994; Sterman,
2000, 2002). The limitation regarding human cognitive capability comes from several aspects:
bounded rationality, incomplete mental models, and difficulty with mental processing (Ster-
man, 2000).

According to Simon, bounded rationality is a property of decision-making that reflects people’s
cognitive limitations (Simon, 1957). Individuals facing complex choices are unable to make objectively rational decisions although they attempt to be rational. People are not machines that follow the same routines day by day. Instead, people are flexible and emotional, combining different levels of knowledge, experiences and bias. As Sterman writes, humans are not only rational beings, coolly weighting the possibilities and judging the probabilities. Emotions, unconscious motivations, and other nonrational or irrational factors all play a large role in our judgments and behaviour (Sterman, 2000). With bounded rationality, one is likely to follow the rules of thumb, norms, and habits to make a decision that is far from optimal. There are a number of studies showing that decision making in complex system is far from optimal (e.g., Brehmer (1992); Diehl and Sterman (1995); Kampmann and Sterman (1994); Paich and Sterman (1993); Sterman (1989a,b)).

With limited information available, our mental models are fuzzy, incomplete and imprecisely stated (Forrester, 1971b). Due to our judgemental bias, the human mind is susceptible to selecting scraps of information which seem fit as a description of the event (Sterman, 2000). On the basis of these assumptions a person estimates the system behaviour that he believes is implied. Consequently, the problematic mental models people use to guide their decisions are deficient. In fact, any concept or assumption that can be clearly described in words, can be incorporated in a computer model (Forrester, 1971a). Once it is done, the model can be discussed and can benefit from other opinions.

Limited mental processing ability is another source of human cognitive incapability. The human mind is incapable of anticipating the consequences of the assumptions and the behaviour of the structure (Sterman, 1994). The inability to process all information required mentally, means that at most one can look a few steps ahead in future (Mojtahedzadeh, 1997). Simon (1957) has articulated the limits on human decision-making ability in his “principle of bounded rationality”, for which he won the Nobel Memorial Prize in Economics in 1979.

“The capacity of the human mind for formulating and solving complex problems is very small compared with the size of the problem whose solution is required for objectively rational behaviour in the real world or even for a reasonable approximation to such objective rationality.”
The increasingly complex dynamic system exceeds our processing ability. One is easily challenged by the multiple interconnections, nonlinearities, time delays between action and response and other elements of dynamic complexity. Without simulation, even a perfect mental model yields little insights.

In some cases, simulation is extremely important because experimentation in real systems is infeasible and simulation is then the only practical way to test models. Simulation can evaluate the consequences of the proposed solutions, thus providing insight into the cause of action. Without simulation, even the best conceptual models can only be tested and improved by relying on the learning feedback through the real world. This feedback is slow and error-prone. Moreover, it is costly because once implemented, policies are often not easily reversible, and the effect will not be revoked in a short time period. In these circumstances simulation becomes the only reliable way to test hypotheses and evaluate the likely effects of policies.

1.2 The Sources of Complexity

As we have discussed the increasing complexity of reality is one drive for the use of simulation. This section is devoted to exploring the properties of complexity in a given model. The complexity of dynamic systems comes from four major sources, feedback loops, stock and flows, time delays and nonlinearities (Forrester, 1969; Saleh, 2000).

Feedback loops are the building blocks of dynamic systems. A feedback loop is a closed path, representing a chain of causal-effect relationships. The feedback loop is a source of complexity because it fosters the interactions among its components, involving time delays, stock and flow structures, and nonlinearities. People are not good at feedback loop thinking (Forrester, 1971a,b; Richardson, 1983; Sterman, 2002). In contrast, they are prone to develop an event-based, open-loop view of causality (Sterman, 2000), which is often myopic. Experiments, such as the beer distribution game (Jarmain, 1963; Sterman, 1989b), multiplier-accelerator model of the economy (Forrester, 1982), were conducted and reflected misperceptions of feedback among the subjects. These experimental studies show that people do quite poorly in systems with even modest levels of dynamic complexity. The poor performances expose that people are “forgetful” about their past decisions and fail to take them into consideration for current decisions. According to Sterman, misperceptions of feedback originates from two aspects,
failing to: (1) adequately account for the delay between a control action and its effect, (2) understand the feedback between their own decision and the environment (Sterman, 1989a).

We are familiar with stocks and flows, for example, the bank account increases with deposits and decreases as one spends. In short, flows can increase (inflow) or decrease (outflow) the stocks, while stocks absorb the difference between inflows and outflows as time accumulates. It follows that a stock variable depends only on the associated flows and never depends directly on any other stock variable (Forrester, 1968a). Yet despite everyday experience of stocks and flows, experimental studies show that most people do not have a good grasp of the concepts of stock and flow (e.g., (Cronin and Gonzalez, 2007; Sterman, 2002)). A simple bathtub experiment was conducted by Sterman and Sweeney in MIT to test the understanding of the stock and flow. The graduate students were asked to infer the behavior of the stock, i.e., sketch the path for the quantity of water in the bathtub, given information on the flows. The flows were simple patterns of inflow and outflow (for details of this experiment, see (Sterman, 2002)). Unfortunately, only 36 percent of the graduate students answered correctly. Even the simplest structure yields poor results, not to mention a structure containing the feedbacks, time delays, or nonlinearities. Irrespective of the difficulty in understanding the stock and flow structure itself, stocks are sources of time delays and disequilibrium dynamics, which complicate the system behaviour (Sterman, 2000).

A time delay describes a process whose output lags behind its input in some fashion. The complexity of a delay lies not only in comprehending it, but also in estimating and modeling it. Time delays reduce the number of times one can cycle around the learning loop, slowing the ability to accumulate experience, test hypotheses, and improve (Rahmandad et al., 2009; Sterman, 1994). The failure of not recognizing time delays between taking action and its effects on the state of the system is common and particularly problematic. Time delays breed instability and oscillation (Forrester, 1961). An empirical rule states the oscillation usually occurs when disequilibriums propagate within a negative feedback loop involving a time delay (Graham, 1977). In practice, the time delay can be inferred from experience and real data. A wide range of statistical tools are available to help estimate the duration and distribution of lags when numerical time series data are accessible (Hamilton, 1980). For more sophisticated time delays, high-order delays or nonlinear adjustment times can be deployed in modeling (see (Sterman, 2000; Yasarcan, 2011)).
Dynamic problems are naturally harder than static problems. Variables change over time as they interact in a given system. The problem is further complicated when dynamics are created by operation of feedback loops, especially in a nonlinear system. The importance regarding nonlinearity has been recognized for centuries, but it is only since the advent of computer simulation that nonlinearity has become important in dynamic modeling (Colye, 1996). A characteristic of a nonlinear system is that its differential equations do not satisfy the superposition principle (Medio and Lines, 2001). As a result, the nonlinear system, in which the variable(s) to be solved cannot be written as a linear combination of independent components (i.e., no exact solution is known), exhibits much more complex behaviors, such as limited cycles, fractals and chaos. The pattern of nonlinear systems is elusive and overwhelmingly complicated, or even impossible for human to comprehend mentally. More advanced mathematical techniques are required to analyze the nonlinear problems (Hilborn, 2000; Medio and Lines, 2001; Mosekilde et al., 1988a; Nayfeh and Balachandran, 1995; Strogatz, 2000). Another consequence of nonlinearity is the occurrence of a shift of loop dominance. The loop dominance analysis is a primary topic of this thesis, and more will be discussed in the following chapter.

1.3 System Dynamics - A Solution To Complex Feedback Systems

Although simulation provides an effective way to forecast the system behaviour, test the hypotheses and evaluate the likely effects of policies, it cannot answer the following questions: What is the underlying cause of the observed system behaviour, do the causes vary at different period of time, how can we identify these internal structural causes, particularly feedback loops and take effective actions to alter the problematic behaviour to be satisfactory.

The formal analysis which involves a number of more sophisticated methodologies in system dynamics (SD) provides answers to these questions. These formal methodologies offer various means to quantify the contribution of the parameters or feedback loops. The philosophy of SD is its focus on feedback loop and the endogenous point of view (Richardson, 2008; Saleh, 2000). System dynamists propose that system behaviour can be explained by its internal feedback structure, therefore, structural cause can be identified, possible policies are developed to adjust and resolve the problematic behaviour. A comprehensive definition of system dynamics is outlined in (Colye, 1996):
“System dynamics deals with the time-dependent behaviour of managed systems with the aim of describing the system and understanding, through qualitative and quantitative models, how information feedback governs its behaviour, and designing robust information feedback structures and control policies through simulation and optimization.” — R. G. Coyle, System Dynamics Modelling: A Practical Approach (1996), p.10.

A characteristic that makes system dynamics different from other approaches to studying dynamic systems is the use of feedback loops and stocks and flows. These elements help describe how even seemingly simple systems display baffling nonlinearity (Radzicki and Taylor, 2008). In system dynamics, dynamic behavior is thought to arise due to the principle of accumulation. More precisely, the principle states that all dynamic behavior occurs when flows accumulate in stocks. Stocks characterize the state of the system and generate the information upon which decisions and actions are based.

System dynamics was developed during the mid-1950s by Forrester (1958, 1961), a professor of the Massachusetts Institute of Technology. Initially, system dynamics was termed “industrial dynamics”, as it was targeted to deal with industrial problems, such as supply chain instability. Later, Forrester’s experiences as a consultant led him to conclude that the biggest impediment to progress comes, not from the engineering side of industrial problems, but from the management side. This is because social systems are much harder to understand and control than physical systems (Forrester, 1971b). As a result, the name was changed to system dynamics to reflect the breadth and universal nature of the research program. The origins of system dynamics can be traced back to engineering control theory where engineering control theory focuses on the feedback loop control, transient/steady response (Levine, 1996; Ogata, 1997). System dynamics takes these concepts of engineering control theory and applies them to social, managerial domains.

System dynamics consists of both qualitative and quantitative analyses. The qualitative tools are mainly used to capture the model structure, including causal loop diagrams (Sterman, 2000), structure-behaviour diagram (Davidsen, 1992), and stock-flow diagram (originated by Forrester, 1961)). As we have discussed, understanding complex systems requires knowledge of concepts such as feedback, stocks and flows, time delays, and nonlinearity. The interactions between these elements give rise to highly counterintuitive behaviour in a complex model. In system dynamics, the stock and flow diagram is used to model these elements for simulation.
CHAPTER 1. INTRODUCTION

Simulation is governed entirely by the passage of time and is referred to as “time-step” simulation modeling. Typically, differential equations (or difference equations for discrete systems) are used to express the relationship between stocks and flows in system dynamics modeling. Meanwhile, a set of formal quantitative methodologies to analyze and optimize the dynamic feedback systems are required.

The quantitative methodologies in system dynamics focus on feedback loop analysis as the ultimate purpose is to design effective policy to adjust the system behaviour. The task of feedback loop analysis is to identify the dominant feedback loops that are responsible for the behaviour at any period of time. In the context of system dynamics, a feedback loop is defined as a closed path connecting, in sequence, a decision that controls action, the state of the system, and information about the state of the system returns to the decision-making points (Forrester, 1968b). Feedback thinking has been central to system dynamics from the beginning of the field (Richmond, 1993). Richardson (1991) shows the history of feedback loop concepts evolving from engineering to the application in the social science. Forrester asserted that all decisions take place in the context of feedback loops (Forrester, 1961). Thoughtful leaders increasingly recognize that we are not only failing to solve the persistent problems we face, but are in fact causing them. Failure to recognize the feedbacks in which we are embedded, the way in which we shape the situation in which we find ourselves, leads to policy resistance as we persistently react to the symptoms of difficulty, intervening at low leverage points and triggering delayed and distant, but powerful feedbacks (Sterman, 1989a, 2000).

Losing sight of the feedback loops often leads to policy resistance, the tendency for interventions to be delayed, diluted, or defeated by the response of the system to the intervention itself (Meadows, 1982). Policy resistance breeds a sense of futility about our ability to make a difference (Sterman, 2002). Even worse, it undermines the policy and at times even exacerbates the original problem (Ghaffarzadegan et al., 2011). The phenomenon “yesterday’s solution is today’s problem” is the embodiment of the policy resistance. There are many examples showing how the complex systems resist policy changes (Forrester, 1971a; Richardson, 1983; Sterman, 2000, 2002). At the root of this phenomenon lies the narrow, event-oriented, reductionist worldview most people live by. Forrester stresses that most people believe cause and effect are closely related in time and space, while in complex dynamic systems cause and effect are often distant in time and space (Forrester, 1971a). This not only explains our neglect
of feedback structure but also implies the fact that systems are often insensitive to the most intuitive policies. The emphasis on feedback loop thinking helps us expand the boundaries of our mental models and lengthen the time horizon we consider so we can see the patterns of behavior created by the underlying feedback structure, not only the most recent events. One of many goals of system dynamics are to avoid policy resistance, and identify high-leverage policies for sustained improvement (Forrester, 2007b).

An example of the S-shaped model (see Sterman, 2000, p.118) is presented to illustrate how a feedback loop can be analysed to understand system behaviour. The structure and behaviour of S-shaped model is plotted in Figure 1.1. The system growth is exponential at first, driven by the positive reinforcing growth feedback loop since the available resources are sufficient to support the system. Then, an inflection point occurs when \( \frac{d^2x}{dt^2} = 0 \). From that on, the system dominance is taken over by the negative balancing loop as the resource per capita diminish, thereby reducing the fractional net increase rate, and the behaviour gradually slows its growth until it reaches an equilibrium point, i.e., the carrying capacity.

![Figure 1.1 – S-shaped growth: structure and behaviour](image)

Another key idea of system dynamics lies in the endogenous point of view, treating system behavior as generated by the closed feedback loop structures (Forrester, 1991). The endogenous view informs model formulation: Exogenous disturbances are seen at most as triggers of system behavior; the problematic causes are contained within the structure of the system itself. Therefore, a solution can be found by altering the internal structure. Several researchers

---

advocate to hold an endogenous perspective (Forrester, 1991; Ghaffarzadegan et al., 2011; Richardson, 2008; Saleh, 2000; Sterman, 2002). One should keep in mind that if the system dynamics methodology is to be applicable to a problem, the dynamics of the system must be the result of operations of the internal feedback structure. When the dynamics in the real problem are too vulnerable to external influences, there is not much methodology can do (Barlas, 2007).

However these cases are rare. More often, the argument that the dynamics of the system are caused by external forces is a result of narrow system conceptualization, not a property of the real problem. So the model boundary must be wide enough to contain all the necessary factors of the internal feedback structure. Individuals who attribute adverse events to exogenous factors, and believe “the enemy is out there lack the ability to learn from the environment and improve their behavior” (Senge, 1990).

A summarized example of pricing policy (p.604, (Sterman, 2000)) clearly demonstrates this idea. This model describes the price competition between firms. Figure 1.2 plots its causal diagram. It is likely for the manager to believe that cutting price will stimulate the market share and boost profits based on his conceptual model shown in the smaller box. However, this mental model takes the competitor price as an exogenous element, is hence only part of the feedback structure of the system. The complete model suggests cutting price when utilization drops activates a reinforcing loop (R1) where the utilization of other competitors will fall, leading them to cut price as well. Therefore, the company will find it is of no help to improve the market share, and cut price again, closing the positive loop. This explanation shows choosing the appropriate boundary to capture all factors as endogenous components is crucial in order to understand and evaluate the policy effects and consequences. In summary, an endogenous perspective is necessary for individual and organizational learning. The feedback view of system dynamics can also be seen as a consequence of the closed causal boundary and the endogenous assumption.

Finally, an example of system dynamics is presented to demonstrate the system dynamics application and usage. Urban Dynamics (Forrester, 1969) is Forrester’s first modeling work dealing with social systems. It describes the major internal forces controlling the balance of population, housing, and industry within an urban area, and then simulates the life cycle of a
Figure 1.2 – An intendedly rational pricing policy can lead to an inadvertent price war.
CHAPTER 1. INTRODUCTION

city and predicts the impact of proposed remedies on the system. This book published counterintuitive conclusions, which suggested that some of the major urban policies were highly detrimental. For example, at that time, building low-cost housing was an appealing policy to both the politicians and civilians, and believed to be essential to reviving the inner cities. However, Forrester pointed out that building low-cost housing was a most damaging policy (Forrester, 2007a). The reason for this is such housing used up space where jobs could be created, while drawing in people who needed jobs. It seemed constructing low-cost housing was a powerful process for creating poverty, not alleviating it. To solve the problem from a full feedback perspective, Forrester recommends policies that increase industrial structures and reduce the stock of available housing. Such policy is tested by adding a zoning system to the model that reserves land for business structures as needed to support the population, and is turned out to be successful (Ghaffarzadegan et al., 2011).

1.4 Research Problems and Objectives of Thesis

A key challenge of system dynamics research is the formal analysis of models. These quantitative analyses concentrate on a primary problem: what is the structural source of system behaviour? There are a number of approaches to answer this question: the traditional approach involving the use of concepts from classical control theory (Ogata, 1997), pathway participation metrics method (Mojtahedzadeh, 1997; Mojtahedzadeh et al., 2004), eigenvalue elasticity analysis (Forrester, 1983; Goncalves et al., 2000; Kampmann, 1996; Saleh and Davidson, 2000; Saleh et al., 2005), eigenvector analysis (Goncalves, 2006), dynamic decomposition weight analysis (Saleh et al., 2006, 2010), the behavioral method (Ford, 1999) and the machine learning approaches. These techniques will be discussed in detail in the following chapter. A brief review outlining strengths and weaknesses of these techniques is presented in (Kampmann and Oliva, 2008). The ultimate goal of these formal analyses is to shed light on decision making and policy design. On the basis of implications derived from these quantitative tools, decision-makers can place their attention on the high leverage points, and maximize the impact of their decisions.

Despite a growing interest among system dynamicists in developing methods for formal quantitative tools, several researchers have recognized the existing shortfalls and problems. For
example, Richardson (1995b) addressed: “For a long time, practitioners have relied on a time-consuming and often incomplete process that iterates from formulation to parameterization, testing, observation, hypothesizing and back again”. This shows the technical support for understanding the connection between model structure and behaviour was weak then. Though the situation has been improved, the techniques available can be enhanced in understanding the behaviour of complex models. Furthermore, Sterman (2000) writes that “Automated identification of dominant loops and feedback structure is one of the future challenges.” An automation of dominant loop identification can show the users how the feedback structures waxes and wanes as the dynamics unfold. Another problem in the system dynamic development is pointed out by Wang (2007): “Less attention has been paid to theoretical research in recent years in SD than in applied SD field.” In fact, theoretical building in SD is as important as applied SD as it is a great impetus promoting the SD forward into the future.

As stated above, current formal quantitative tools can be improved to apply to many models, and can benefit from increased automation. The research objectives of this dissertation are to improve existing methodologies and develop analytical and computational techniques to effectively identify the influential structures causing the complex model behavior. This dissertation will explore how new system dynamics methods can analyze the models to support the generation of insights for policy design. The specific contributions of this thesis are:

1. To identify the drawback of a popular independent loop selection algorithm and propose a new implementation revising it. The shortest independent loop set algorithm which is a widely adopted loop selection algorithm in the eigenvalue elasticity analysis (EEA) method, is found to be a failure in identifying the independent loop set in some occasions. This failure case is exemplified by an agent-based model. Apart from this, the necessity of utilizing a maximal set of independent loops in EEA is thoroughly investigated.

2. To develop a novel analytical eigenvector related analysis methodology that helps people to identify leverage points. While the eigenvalue analysis has been extensively explored, the eigenvector related analysis is in its early stage of development. Compared with the numerical analysis, an analytical eigenvector related analysis approach is proposed to fast identify the high leverage parameters. We suggest looking at eigenvalue alone is not sufficient to identify effective intervention points, and it is necessary to consider the
CHAPTER 1. INTRODUCTION

impact on both eigenvalue and eigenvector.

3. To refine and enhance the existing feedback loop analysis approach – the behavioral method. The new version has greater applicability, and is enabled to handle a special scenario when the feedback loop has no unique edge. In the the original version, if no unique edge exists, the candidate feedback loop cannot be deactivated and the method fails. Whereas, we propose a loop deactivation method which enables the loop deactivation by utilizing a unique pair of consecutive two edges.

4. To extend the current loop analysis method from another perspective to complement the method. We propose a variant of the behavioural method which blends with sensitivity analysis. This variant method explores a wider search space for policy design and with potential to gain additional insightful implications for policy makers.

5. To develop a framework to automate the formal feedback loop analysis method – the behavioural method. We develop an extensive software package that contains the automation of the behavioral method. Besides, it has the potential to accommodate other dominant loop analysis methods, e.g., the variant of the behavioural method.

1.5 Organization of Thesis

This section provides a summary of each chapter. From the early chapters where the related research is introduced, through to the results chapters where the proposed approaches are applied to experiments to shed the light on the policy design. Finally, the contribution and some implications for future research are discussed. The structure of this thesis is as follows:

- Chapter 2 presents a review of system dynamics approaches for structural dominance analysis. It begins with an introduction to the behaviour characterization in both linear and nonlinear systems. Subsequently, two important aspects of feedback loop, loop polarity and loop strength are discussed. Finally, it ends with a review of relevant structural dominance analysis approaches which fall into analytic and numeric categories.

- Chapter 3 addresses a problem regarding the shortest independent feedback loop set (SILS) algorithm. This algorithm has been widely used in the eigenvalue elasticity analysis method, and plays an essential role in it. However, we find in some occasions, the
SILS algorithm cannot identify the maximal set of independent feedback loops. This issue is exemplified by an agent-based model. A series of relevant problems are subsequently investigated in this chapter. Finally, an algorithm to identify the maximal independent loops is proposed.

- Chapter 4 strengthens the role of the eigenvector analysis in identification of leverage points. The contribution of this chapter is it proposes an analytical method of eigenvector (weight) analysis to efficiently identify the highest leverage points. The analytic eigenvector analysis method is demonstrated by a linearized version of the labor-inventory model. Moreover, the EEA is conducted to identify the leverage point which is then compared with that from the eigenvector analysis. The outcome shows the eigenvector analysis yields better intervention points to improve the performance of the selected behaviour of interest. As a result, it is therefore advisable to consider both eigenvalue and eigenvector analysis to improve the system behaviour.

- Chapter 5 presents a refined version of the behavioral method regarding its loop deactivation process. The behavioural method is a computational feedback loop analysis method. The extended approach broadens the behavioural method’s applicability, and enables the loop to be deactivated when there is no unique edge within a feedback loop.

- Chapter 6 proposes a variant of the behavioural method which blends with the sensitivity analysis. A novel technique combining the feedback loop analysis and sensitivity analysis to identify the dominant feedback structure is presented. This approach intends to explore a wider search space for policy design. The kernel of this approach is the idea of using various testing values to deactivate the feedback loop and assessing its influence over the variable of interest. We hope additional insightful implications can be obtained from the behavioural method with an sensitivity perspective.

- Chapter 7 outlines architecture designs for two computational dominant loop analysis methods, the behavioural method and the behavioural method based sensitivity analysis. Initially, a simulator which accommodates the behavioural method is introduced. Subsequently, the key components and the implementation issues of the behavioural method are depicted. Finally, a conceptual modular design for the sensitivity analysis is also presented. The reusable components from the previous design are highlighted while the new components that need to be added are discussed.
Chapter 8 summarizes the contributions in light of existing research in the domain as well as highlights the potential research opportunities resulted from the work presented through this dissertation.
Chapter 2

Background and Related Research

This chapter presents reviews of many important and relevant aspects of dynamic behaviour, feedback loops, and system dynamics approaches that map structure to behaviour. A general review of system dynamics in terms of highlighting the relationship between system structure and behaviour is initially presented. The subsequent section outlines the classifications of various behaviours in both linear and nonlinear systems. Consequently, the characteristics of feedback loops are described together with a discussion of the behaviour in relation to loop polarity. A review of structural dominance analysis approaches, which is classified into feedback loop and parameter analysis, is presented. Finally, it ends with a summary emphasizing the limitations of existing research and exploring the potential research directions.

2.1 Overview

One problem, model validation, stated by Oliva (2004) is that “the most serious difficulty with a large number of calibration handles, however, is the increased difficulty of detecting formulation errors. In an endeavor to match historical data, the calibration process “adjusts” the model structure to cover formulation errors . . . Small deviations from reasonable values and wider confidence intervals make it more difficult to detect fundamental formulation errors, especially when a good fit to historical behavior has been achieved”. The formulation errors involve the feedback structure problems. A further problem is that complex systems are remarkably insensitive to changes in many system parameters (Forrester, 1971b). Given
these considerations, the parameter calibration sometimes is not sufficient in improving or adjusting the system performance. It requires the feedback loop analysis to tackle these more sophisticated problems. Feedback loop analysis can help modelers verify the model as the identified dominant feedback loops can be used to test whether the alternate of these dominant loop explains the varied behaviour. Moreover, as we previously mentioned, a feedback loop represents a chain of causal-effect relationships and thus is able to tell a coherent story with a strong explanatory power. In this dissertation, we primarily utilize the feedback loops, directly or indirectly, to explain the behaviour of the complex dynamic systems.

According to Starr (1980), designing a new policy is an activity of (i) assigning alternative values for parameters, (ii) changing linkages among system elements, and/or (iii) inserting alternative elements into a model. This dissertation focuses on the following questions: what are the effects of changing parameters or feedback structures; how can the resulting behaviour be analyzed; and which feedback loops or parameters are most significant to the behaviour of interest? Formal structure analysis serves to identify the contribution of specific parameters and feedback loop structures to a given behaviour. For example, statistical screening (Ford and Flynn, 2005; Taylor et al., 2010a), optimization-related approaches (Chen et al., 2011), and eigenvalue elasticity analysis (Forrester, 1983). These methodologies use simulation and formal mathematics to identify the dominant feedback structures which the behaviour is most sensitive to. One novelty of formal analysis in system dynamics is the focus on the feedback loop analysis. Mapping the behavioral characteristics of a system to its feedback loop structure, which in the system dynamics literature is referred to as dominant feedback structure, is a significant contribution of system dynamics in understanding complex systems. The principle “structure drives behaviour” provides the foundation for its development. Feedback thinking is necessary to explain the resulted dynamic behaviour for the reason that feedback structure is capable of keeping track of a closed chain of causes, thus producing a more comprehensive explanation than a single parameter or link.

An application of structural dominance analysis is a simplification – simplifying the complex systems. However, there has been little effort in the system dynamics community devoted to this topic (Elberlein, 1981). Elberlein presents a formal theory of model simplification as a means of increasing model understanding, which identifies important feedback loops in
linearized models with respect to a selected dynamic behaviour (Elberlein, 1981). He considers the problem of simplifying a large model based on the dynamic characteristics of the large model. In particular, the simplification that preserves specific eigenvalues of the original model. Discussions on the complex and simplified models can be found in Saysel and Barlas (2006) and Ghaffarzadegan et al. (2011).

In order to study and develop formal methodologies to identify dominant feedback structures, the issues relating to dynamic behaviour and feedback structures are pivotal. All these relevant elements are to be described in the rest of this chapter.

### 2.2 An Overview of Dynamics

*Dynamics* is the study of change over time of variables such as temperature, species population, voltage, price and so forth. This is often achieved by means of equations linking the values of variables at different, uniformly spaced instants of time, i.e., difference equations, or by systems relating the values of variables to their time derivatives, i.e., *ordinary differential equations* (ODE)\(^1\). The dynamic behavior of a system variable is actually a set of values generated during the simulation. These points result in a certain shape of lines or curves, and these lines or curves are referred to as the dynamic behavior.

This section introduces typical dynamic behaviours in both linear and nonlinear models, and the classification of various behaviours using eigenvalues, and atomic behaviour patterns. In terms of linear systems, this is a well-developed area of research and the theory of linear dynamical systems offers the key to analyzing this behavior. The typical behavior in linear system is of three basic types: reinforcing, balancing and oscillatory. On the contrary we have nonlinear system, in which the variable(s) to be solved cannot be written as a linear combination of independent components, many exhibit much more complex behaviors, e.g., limited cycles, quasiperiodic motion and strange attractors. In most cases, it cannot be solved analytically, and only simulation will do to obtain its results.

\(^1\) Dynamical phenomena can also be investigated by other types of mathematical representations, such as partial differential equations, lattice maps or cellular automata. This research, however, concentrates on the study of differential equations and their dynamical behavior.
2.2.1 Behaviours in Linear Systems

2.2.1.1 Mathematical Representation of Linear Systems

Broadly speaking, we say a phenomenon represented by a stimulus-response mechanism is 
linear if, to a given change in the intensity of the stimulus, there corresponds a proportional 
change in the response. As concerns dynamical systems, a continuous system regardless of a 
linear or nonlinear system can be expressed as:

\[
\begin{align*}
\dot{x}_1 &= f_1(x_1, x_2, \ldots, x_n) \\
\dot{x}_2 &= f_2(x_1, x_2, \ldots, x_n) \\
&\quad \vdots \\
\dot{x}_n &= f_n(x_1, x_2, \ldots, x_n)
\end{align*}
\]

It associates the net change in the state variable with current system states. The above system 
can be denoted in a matrix form:

\[
\frac{d\mathbf{x}}{dt} = \dot{\mathbf{x}} = F(\mathbf{x}) \quad \mathbf{x} \in \mathbb{R}^n \quad t \in \mathbb{R} 
\] (2.1)

where \(\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_n \end{bmatrix}^T\), a column vector of first time derivatives of the state variables \(\mathbf{x}(t)\), \((\cdot)^T\) refers to transpose; \(\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T\).

The system is linear if function \(F : \mathbb{R}^n \to \mathbb{R}^n\) satisfies:

\[
F(\alpha \mathbf{v} + \beta \mathbf{w}) = \alpha F(\mathbf{v}) + \beta F(\mathbf{w})
\] (2.2)

for any \(\alpha, \beta \in \mathbb{R}\) and \(\mathbf{v}, \mathbf{w} \in \mathbb{R}^n\).

Assume the system in Equation 2.1 is a linear system with \(n\) state variables (corresponding to 
integrals in the stock-flow diagram). \(F(\mathbf{x})\) can be further decomposed into another form, as

\[
\dot{\mathbf{x}} = A\mathbf{x} \quad \mathbf{x}(0) = \mathbf{x}(t_0)
\] (2.3)

where \(A\) is an \(n\)-by-\(n\) compact gain matrix of this system, unfolded in Equation 2.4, and \(\mathbf{x}(0)\) 
is an \(n\)-by-one column vector representing the initial condition of the system. The gain matrix 
is used as a condensed representation of the model structure, and is the point of departure to 
derive the state trajectory. Each entry of \(A\) represents the partial derivative of the net change 
of a state variable with respect to any state variable. As it is a linear system, \(A\) is a constant
matrix. Any entry $A_{ij}$ can be viewed as a coefficient between a net flow of state variable $x_i$ and a the state variable $x_j$. We name them compact link gains afterwards as they can be thought of as hypothetical links which connect a flows to a state variable.

\[
A_{(n,n)} = \begin{pmatrix}
\frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\
\frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} & \cdots & \frac{\partial \dot{x}_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial \dot{x}_n}{\partial x_1} & \frac{\partial \dot{x}_n}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n}
\end{pmatrix}
\]

(2.4)

### 2.2.1.2 Behaviour Modes Characterized by Eigenvalues

Characterizing the behavior to distinguish one behavior from others is a fundamental task in exploring dynamic systems. It is also a prerequisite to ultimately identify what types of feedback structures give rise to various behavior and why. For a linear system represented in Equation 2.3, its eigenvalues are derived from the gain matrix $A$. An eigenvalue, denoted by $\lambda$, is a special scalar that satisfies the following equation:

\[|A - \lambda I| = 0\]

(2.5)

where the symbol “|” represents the determinant, and $I$ is an n-by-n identity matrix. Equation 2.5 is known as the **characteristic equation**. There are $n$ solutions for $\lambda$. We assume these $n$ $\lambda$s are always distinct. For any eigenvalue $\lambda_i$, there exists two types of vectors, that satisfy:

\[A r_i = \lambda_i r_i\]

\[\ell^T_i A = \lambda_i \ell^T_i\]

(2.6)

where $r_i$ is a column vector $\begin{pmatrix} r_{i1} \\ \vdots \\ r_{in} \end{pmatrix}$, called **right eigenvector**, and $\ell^T_i$ is a row vector $[\ell_{i1} \ell_{i2} \ldots \ell_{in}]$ named **left eigenvector**. A nonzero scalar multiple of an eigenvector is equivalent to the original eigenvector. In other words, the eigenvectors are not unique. The solution to Equation 2.3 can be written in the expression below:

\[x(t) = \alpha_0^1 e^{\lambda_1 t} r_1 + \ldots + \alpha_n^0 e^{\lambda_n t} r_n\]

(2.7)

where $\alpha^0$ are constants determined by the initial condition, $x$, $\lambda$ and $r$ are defined beforehand. The procedure of obtaining this equation can be found in chapter 4. To be clearer, Equation
2.8 shows a solution to a single state variable.

\[ x_i = \alpha_i^0 e^{\lambda_i t} r_{i1} + \ldots + \alpha_i^n e^{\lambda_i t} r_{ni} \]  

(2.8)

It is evident to us that eigenvalues (\(\lambda\)) are the crucial factor in determining the behaviour since they are exponents of each term in the system behaviour. Consequently, there are five behaviour modes characterized by eigenvalues, shown in Figure 2.1.

(a) Monotonic divergent behaviour mode: a real positive eigenvalue;

(b) Monotonic convergent behaviour mode: a real negative eigenvalue;

(c) Sustained oscillatory behaviour mode: a complex conjugate pair of eigenvalues with zero parts;

(d) Convergent oscillatory behaviour: a complex conjugate pair of eigenvalues with negative real parts;

(e) Divergent oscillatory behaviour: a complex conjugate pair of eigenvalues with positive real parts; and

(f) Linear behaviour: a zero-valued eigenvalue.

Given that each eigenvalue characterizes a certain behaviour mode, each individual term in Equation 2.7 can be viewed as the contribution from a specific behaviour mode. Real-valued eigenvalue tells if the behaviour runs away from or approaches zero. Any eigenvalue with a positive real part will finally dominate the behaviour and drive it to infinity as time goes. A system is stable only when there is no positive real part of any eigenvalue. One important behaviour mode characterized by eigenvalues is the oscillation, which corresponds to a pair of complex conjugate eigenvalues. The details of how to map the conjugate eigenvalues to the oscillation can be found in Appendix A. (Note: for a real-valued square matrix, its complex eigenvalues always appear in conjugate (Trefethen and Bau, 1997)). The decaying oscillation, e.g. in Figure 2.2, can be thought of as a pure sinusoid with a frequency equal to the magnitude of the eigenvalue which is squeezed to fit into an envelop of two decaying curves. In addition, there are some terms regarding the oscillatory behaviour:

- The frequency of the pure sinusoid is called the natural frequency. More specifically, it is the frequency at which the if the real part component of the eigenvalue, the attenuation
or amplification, had been reduced to zero. The natural frequency is equal to the absolute value of the complex eigenvalue, $\sqrt{\text{Re}(\lambda) + \text{Im}(\lambda)}$.

- The observed frequency in the continuous decaying cycles in Figure 2.2, is called the damped frequency, which is equal to the absolute value of the imaginary parts of the eigenvalues.

- The damping ratio measures the speed of convergence of an oscillation to equilibrium. It is closely related to the relative height of successive cyclic peaks, i.e., $-M2/M1$. The higher ratio, the more attenuation occurs between successive peaks. The damping ratio is calculated as the cosine of the angle in the complex plane from the positive real axis to the eigenvalue.

The above measurements of oscillation are discussed comprehensively in (Forrester, 1982) where they are used as various criteria to assess the impact of the policy in stabilizing a system.

An example of the floating goals model (Sterman, 2000) is used to illustrate the procedure of the eigenvalue computation in a linear system.
Figure 2.2 – Complex conjugated eigenvalue characterized behaviour

\[
\dot{S} = \frac{(G - S)}{sat}
\]
\[
\dot{G} = \frac{(S - G)}{gat}
\]
\[
sat = 8 \quad gat = 6 \quad G(0) = 10 \quad S(0) = 0
\]

First, the gain matrix of the above system is:

\[
A = \begin{bmatrix} \frac{\partial S}{\partial S} & \frac{\partial S}{\partial G} \\ \frac{\partial G}{\partial S} & \frac{\partial G}{\partial G} \end{bmatrix} = \begin{bmatrix} -1/st & 1/st \\ 1/gt & -1/gt \end{bmatrix}
\]

The characteristic equation for the gain matrix is then written as:

\[
0 = |A - \lambda I| = \begin{vmatrix} -1/8 - \lambda & 1/8 \\ 1/6 & -1/6 - \lambda \end{vmatrix} = (-1/8 - \lambda)(-1/6 - \lambda) - (1/8 \times 1/6)
\]
\[\therefore \lambda_1 = 0, \lambda_2 = -0.3\]

Subsequently, substituting \(\lambda\) into Equation 2.6 to compute the right eigenvectors: \(r_1 = [1 \ 1]^T\), \(r_2 = [-0.6 \ 0.8]^T\). Finally, in light of Equation 2.8 and the system initial condition, the solution is:

\[
S = \frac{30}{7} - \frac{30}{7}e^{-0.3t}
\]
\[
G = \frac{30}{7} + \frac{40}{7}e^{-0.3t}
\]

These two eigenvalues suggest the behaviour is a superposition of a linear mode, i.e., \(\lambda_1 = 0\),
and a monotonic convergent mode, i.e., $\lambda_2 = -0.3$. Both state variables converge to the same value, $30/7$. The behaviour can be solved by an analytical solution or simulation, both of which is plotted in Figure 2.4.

### 2.2.1.3 Atomic Behaviour Pattern

Another mathematical rigorous method for characterizing behaviour is proposed by Ford (Ford, 1999) where he utilizes the derivatives of the variable of interest to distinguish various behaviour patterns. The net rate of change is the first derivative of the variable $x$, $(dx/dt)$. The second derivative describes the movement of the net rate of change; a positive second derivative indicates an accelerating speed in the net rate of change, while a negative second derivative
indicates a decreasing speed in the net rate of change and a second derivative equal to zero indicates a constant rate of change. As a result, three atomic behaviour patterns (ABP) emerge and are formulated by the variable (denoted $x$) derivatives with respect to time:

- **Exponential ABP** (growth or decay): $\frac{d|\dot{x}|}{dt} > 0$;

- **Logarithmic ABP** (growth or decay): $\frac{d|\dot{x}|}{dt} < 0$;

- **Linear ABP** (growth, decay or equilibrium): $\frac{d|\dot{x}|}{dt} = 0$.

**Note:**

$$\frac{d|\dot{x}|}{dt} = \frac{d(|dx/dt|)}{dt}$$

Figure 2.5 plots three unique ABPs where the logarithmic behaviour shows a growth with a decreasing speed, the exponential behaviour shows a growth with an increasing speed. With this formal classification of behaviour pattern, the dynamic behaviour can be partitioned based on ABP and analyzed within an individual phase. A merit of the ABP classification lies in its applicability to both linear and nonlinear systems. However, one drawback is that it does not have expression for oscillatory behaviour. Oscillation is detected by a repeated sequential behaviour patterns of exponential growth, logarithmic growth, exponential decline and logarithmic decay.

### 2.2.2 Behaviours in Nonlinear Systems

A nonlinear system does not satisfy Equation 2.2, and its gain matrix $A$ (if it can be written as in Equation 2.3) usually contains the current state variables, resulting in a non-constant
matrix. Therefore, the behaviour becomes much more intractable. When we discuss the non-linear system behaviour in this dissertation, the phase plot is frequently used in which state variables are represented in the same coordinate. It is different from the linear system behavior, where a single variable behaviour is usually portrayed with respect to the time axis (see figure 2.7). As we have mentioned before, more often than not, the nonlinear systems cannot be precisely evaluated. Its stability then is more concerned, especially the local stability properties of “attractors”. For any deterministic nonlinear model, in the steady state – i.e. after all the transients dynamics die out, the model is “attracted” to one of the following four kinds of attractors (Saleh, 2002):

1. An equilibrium point;
2. A limit cycle (periodic motion);
3. A quaisperiodic motion;
4. Aperiodic oscillations or a strange attractor - functions that do not repeat values after some period.

![Phase trajectories in the neighborhood of various limit cycles](image)

**Figure 2.6** – Phase trajectories in the neighborhood of various limit cycles, which are depicted as closed curves: (a) stable (b) unstable (c) and (d) half-stable (from The Great Soviet Encyclopedia (1979)).

A limit cycle is an isolated closed trajectory. Its neighboring trajectories move either toward or away from the limit cycle (Figure 2.6). An example of a limit cycle, the Van de Pol oscillation,
is presented in Figure 2.7. Figure 2.7(a) shows three neighboring behaviours with different initial values are finally attracted into the limit cycle. The behaviour can also be viewed from another perspective in Figure 2.7(b) which is more straightforward, and shows $x$ and its first derivative $\dot{x}$ both oscillate. It explains why the cycle will be formed as portrayed in Figure 2.7(a). Limit cycles are inherently nonlinear phenomena, and they cannot occur in linear systems. Although a linear system can have closed orbits, they are not isolated.

$$\ddot{x} + \mu(x^2 - 1)x + x = 0$$ when $\mu = 1.5$.

Strange attractors can be described using a geometric object called fractal set. Fractals are complex geometric shapes with fine structure at arbitrarily small scales. They have some degree of self-similarity. Strange attractors have two properties: trajectories on the attractor remain confined to a bounded region of phase space, yet they separate from their neighbors exponentially fast. They are typically characterized by the fractal dimension which is the minimum number of coordinates needed to describe every point in the set. The fractal dimension can be computed by the box-counting algorithm (Grassberger and Procaccia, 1983b), pointwise and correlation dimension calculation (Grassberger and Procaccia, 1983a). Strange attractors, in almost all cases, indicate chaotic systems. Chaotic systems are deterministic, their evolutions are governed by dynamical equations, but they are unpredictable owing to their extreme sensitivity to the initial conditions. A slightest difference in the initial condition will drive the two trajectory to diverge exponentially after a while. Logistic maps (May, 1976), Lorenz map (Lorenz, 1963), the beer distribution game (Mosekilde et al., 1988a) under certain
parameter settings are examples of chaotic systems.

The Lyapunov exponent is to quantify the stochasticity properties, related essentially to the exponential divergence or convergence of nearby orbits in phase space (Benettin et al., 1980). The formal definition to the Lyapunov exponent $\lambda$ is (Strogatz, 2000):

$$|\delta_n| = |\delta_0|e^{\lambda t}$$

where $x_0$ is the initial condition, $\delta_0$ is the distance to a certain nearby point and is extremely small, $\delta_n$ is the distance after $n$ time iterations. However, in most cases, the Lyapunov exponent is calculated using the expression below:

$$\lambda = \lim_{n \to \infty} \left\{ \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \right\}$$

Note that $\lambda$ depends on $x_0$, but it is same for any $x_0$ that is in the basin of attraction of a given attractor. Moreover, the equation above solves for the sum of all Lyapunov exponents in a system. Methods to determine the largest, non-negative, and spectrum of several Lyapunov exponents (including positive, zero, and negative ones) from the observed time series of a single variable are proposed in (Rosenstein et al., 1993), (Wolf et al., 1985), and (Sano and Sawada, 1985) respectively. One or more positive Lyapunov exponents indicates a chaotic system. If all exponents are negative, then all trajectories are attracted to a fixed-point attractor. One exponent zero, and the rest negative, implies the existence of a limit cycle.

There are many types of stability, an equilibrium point $x^*$ can be:

- An **attracting** point if all trajectories start near $x^*$ approach it as $t \to \infty$;

- Lyapunov **stable** if all trajectories that start sufficiently close to $x^*$ remain close to it for all time;

- Neutrally **stable** if $x^*$ is Lyapunov stable but not attracting;

- **Stable** if $x^*$ is both attracting and lyapunov stable;

- **Not stable** if $x^*$ is neither attracting nor Lyapunov stable.

In order to determine the stability of an equilibrium point in a nonlinear system, linearization near the equilibrium point is necessary. Taylor series expansion is a technique for linearization. Interested readers can find the procedure of the linearization analysis in (Saleh, 2002).
CHAPTER 2. BACKGROUND AND RELATED RESEARCH

Figure 2.8 – Basic structure of the production-distribution system considered in the beer distribution game. Orders for beer propagate from right to left, while the brewery products are delivered in the opposite direction.

System dynamicists encounter the chaotic systems a lot in their research (Allen; Mosekilde et al., 1988b; Mosekilde and Laugesen, 2007; Sterman, 1988; Toro and Aracil, 1988). Two of the most extensively analyzed chaotic models are Sterman’s economic long wave model (Rasmussen et al., 1985; Sterman, 1985) and the beer distribution game (Forrester, 1961; Mosekilde and Larsen, 1988). Mosekilde and Larsen (1988) study the beer distribution game which involves piecewise-linear relations associated with non-negativity conditions between four sectors (see Figure 2.8). They find that using different ordering policies, a great variety of complex dynamic behaviours including limit cycles of different periodicities and chaos can occur. These policies are expressed in terms of two parameters, the fraction of unfilled orders accounted for in the ordering policy (B), and the fraction of anticipated shipments taken into account (D). The higher parameter values corresponds to a better policy capable of stabilizing the system. Particularly, Figure 2.9 plots a chaotic behaviour that the beer distribution game would exhibit under a certain set of parameters.

Furthermore, an interesting sequence of subsequent Hopf bifurcations are identified in the beer game by Laugesen and Mosekilde (2006). The migratory model that describes the migration between three subpopulations differing by their socioeconomic background are studied by (Reiner et al., 1988; Richardson and Sterman, 1988). The study finds the dynamics of the migratory system is sensitively dependent on the assumed number of subpopulations and their interactions. For example, in the case of one homogeneous population in three regions, the attractors are fixed points only, but for two or three populations in three regions, fixed points, limit cycles, and chaotic strange attractors can all exist based on the specification of the interaction trends between the populations (Reiner et al., 1988).
In this section, we have presented a broad discussion of relevant aspects of the linear/nonlinear systems, their typical behaviour and the behaviour classification. Initially, an ODE representation of a linear system is described, where the eigenvalue is induced to characterize the behaviour. The eigenvalues are important indicators in a linear system since they determine the system stability and the trend of the state variables. The procedure to calculate the eigenvalue is outlined and illustrated by the floating goals example. Subsequently, another method, the ABP, utilizing the variable’s first and second derivatives to classify various behaviour patterns is presented. This behaviour classification plays an important role in the structural analysis. Finally, various types of nonlinear systems behaviour (using phase plot to show an equilibrium point, a limit cycle or the strange attractor) and their stability (such as Lyapunov stable, neutrally stable and so on) are introduced. The computation of the Lyapunov exponents is a key step to identify a chaotic system.

2.3 Characteristics of Feedback Loops

“Structure drives behaviour” is considered as a primary principle in the system dynamics paradigm (Güneralp, 2004a; Saleh and Davidsen, 2000). A structure specifies the causal relationships between the elements of the system in terms of stocks and flows and feedback loops (Mojtahedzadeh and Richardson, 1995). The principle can be interpreted as the behaviour of
CHAPTER 2. BACKGROUND AND RELATED RESEARCH

the system is determined by the interaction of various feedback loops. Feedback loops are
the basic building blocks of dynamic systems. A feedback loop is defined as a closed path
connecting, in sequence, a decision that controls action, the state of the system, and infor-
mation about the state of the system returns to the decision-making points (Forrester, 1968b).
This section focuses on characterizing the feedback loop with respect to its gain and polarity.
The loop polarity can be determined by the sign of the loop gain, and is defined to be only
positive or negative. As the dynamic behaviour is classified into various modes, we attempt
to understand how the shifts in loop dominance would give rise to the change in behaviour
modes. Initially, several definitions of the loop polarity and loop gain are introduced. Among
this, a special definition of the dominant loop polarity is presented and is further interpreted
through two scenarios: the single-loop structure and multi-loop structure. Finally, we summa-
rize the work that studies the relationship between the feedback loop polarity and the system
behaviour.

2.3.1 Loop Polarity and Loop Gain

As is stressed by Richardson (1991), “it is the concept of the loop polarity that gives the
feedback loop its perceived analytic and explanatory power” (p.5). It is therefore crucial to
study the loop polarity to see what role it plays in altering the observed behaviour. There are
several definitions of loop polarity in the system dynamics literature. One is defined based on
the open loop gain. According to Sterman (2000), the open loop gain can be calculated in the
following procedure. One first breaks the loop at any point, e.g., \( x_1 \) (in Figure 2.10), which
splits into an input \( x_{i1} \), and an output \( x_{o1} \). The open loop gain is defined as the partial derivative
of \( x_{o1} \) with respect to \( x_{i1} \). Together with the chain rule, it is calculated in Equation 2.12.

\[
\frac{\partial x_{o1}}{\partial x_{i1}} = \frac{\partial x_{o1}}{\partial x_4} \frac{\partial x_4}{\partial x_3} \frac{\partial x_3}{\partial x_2} \frac{\partial x_2}{\partial x_{i1}}
\]

(2.12)

The gain can be thought of as strength of a disturbance generated by the loop: A gain of two
means a disturbance will be doubled when it returns to the variable after traveling along this
loop; a gain of negative two means a disturbance will be twice the magnitude but in opposite
direction when returning to itself. The polarity is defined as the sign of the open loop gain, i.e.,
if the open loop gain is greater (less) than zero, the polarity is positive (negative). According to
this definition, there is a quick way to determine the loop polarity by just counting the number
of negative links. The link polarity is defined by taking the partial derivative of the effect
Figure 2.10 – Open loop gain calculation

variable with respect to the causal variable (see Equation 2.13). If the number is even, the polarity is positive, if the number is odd, the polarity is negative.

\[
polarity \text{ of a causal link} = \text{sign} \left( \frac{\partial \text{ effect variable}}{\partial \text{ causal variable}} \right)
\] (2.13)

In the following loop polarity definitions, we assume that every feedback loop contains at least one stock variable (integration). A \textit{minor loop} refers to the loop with only one stock variable, and a \textit{major loop} refers to the loop with more than one stock variable. In contrast to defining the loop polarity by computing its gain, loop polarity may be defined through a flow and the change rate of the flow as in (Richardson, 1995a). One rigorous definition is proposed by Richardson (1995a), where the polarity of a minor loop and a major loop is treated differently. We first look at the \textit{minor loop} case. Let \( x \) denote a single stock variable and a flow \( \dot{x} = \frac{dx}{dt} \) (\( \dot{x} \) represents any flow including an inflow, an outflow, or a net flow, the type depends on the flow variable involved in the loop under analysis). The loop polarity is defined as:

\[
\text{loop polarity} = \text{sign}(\frac{d\dot{x}}{dx}) = \begin{cases} 
+ & \text{for } \frac{d\dot{x}}{dx} > 0 \\
- & \text{for } \frac{d\dot{x}}{dx} < 0
\end{cases}
\] (2.14)

According to Richardson, he describes the following situation: a small change, \( \dot{x} \), is traced around the loop until it results in a small change in \( \dot{x} \), \( d\dot{x} \). If the change in the rate, \( d\dot{x} \) is in the same direction as the change in the state, \( \dot{x} \), then this loop reinforces the change and so it is a positive loop, otherwise, it is a negative loop (Richardson, 1995a). To expand the definition to feedback loops whose \( \dot{x} \) is an outflow, it merely needs to add a negative sign to the result. The results for the feedback loop with an inflow or a netflow still holds as they are added to the corresponding stock variable. More generally, in a minor loop structure with multiple feedback loops, instead of tracing the disturbance via a single feedback loop,
the aggregated disturbance which is generated by multiple feedback loops converging on \( \dot{x} \) is traced. Therefore, the loop polarity indicates which type of loops is dominant, and it is named as the \textit{dominant loop polarity}. In particular, the dominant loop polarity may shift in a nonlinear system. This definition also refers to \textit{the dominant polarity of the system}.

Consider the \textit{major loop} case. Assume a major loop composed of stocks \( x_1, x_2, \ldots, x_n \) connected in order: \( \dot{x}_1 \rightarrow x_1 \rightarrow \dot{x}_2 \rightarrow x_2 \rightarrow \cdots \rightarrow \dot{x}_n \rightarrow x_n \rightarrow \dot{x}_1 \). The polarity of such a major loop is:

\[
\text{sign}\left( \frac{\partial \dot{x}_2}{\partial x_1} \frac{\partial \dot{x}_3}{\partial x_2} \cdots \frac{\partial \dot{x}_n}{\partial x_{n-1}} \frac{\partial \dot{x}_1}{\partial x_n} \right)
\]

This definition provides a precise statement to formulate the shifts in loop dominance. However, this definition also has a limitation. As noted in (Richardson, 1995a), “this definition does not capture all possible shifts in loop dominance — only those that involve a change in dominant polarity. Presumably, it is entirely possible for a system to show a shift in dominance between two negative loops or two positive loops. Such a shift in dominance between loops of the same polarity would not show up as a change in dominant polarity and would have to be defined and detected by other means”. The loop polarity definition is equivalent to the common perspective of positive and negative: the positive loop corresponds to the reinforcing behaviour, and the negative loop corresponds to the balancing behaviour. It is not difficult to see the mathematic formula matches the statement. For a positive loop, a disturbance returning to itself should be in the same direction and with a larger absolute value (\(|\dot{d}x + dx| > |dx|\)), so that the behaviour of this variable will exhibit accelerating growth or decline.

The subsequent loop polarity definition is described in Mojtahedzadeh (1997), where the author expresses his understanding of polarity: “the polarity of a feedback loop conveys how information about a stock variable travels around the feedback loop through other variables and returns to a flow variable associated with the stock variable” (p.60). In order to formulate the polarity of each individual feedback loop, the link polarity is first defined. The polarity of one type of links, flow-to-stock links, is determined by the direction of the flow:

\[
\text{polarity of an inflow} = +; \quad \text{polarity of an outflow} = -
\]

For other links, the polarity is obtained by Equation 2.13. A positive link polarity does not necessarily mean a change in the causal variable will lead to a change in the same direction in the effect variable as other variables also affect the effect variable. A more precise statement is,
“if the causal variable increases, the effect variable increases above what it would have been” (Sterman, 2000). The polarity of a feedback loop is formulated as the product of its constituent link polarities\(^2\). By this definition, a positive (negative) loop is capable to retain (alter) the sign of changes in the stock as it travels through the links and returns to the associated flow variable. In contrast to the definition by Richardson, this loop polarity is individual loop based which means the loop polarity is determined in terms of each single loop. Compared with the major loop polarity defined by Richardson, this definition works with every link in the loop and pursues the polarity in a single loop level.

\[ \begin{align*}
\dot{x} &= ay \\
\dot{y} &= f(x) \ast x 
\end{align*} \]

where \(a\) is a constant. According to Richardson, the major loop polarity is a product of sequential partial derivatives between flows and stocks. Given Figure 2.11, the major loop polarity

\(^2\) The product of signs are defined to be aligned with the product of numbers: \((+)^* (+) = +\), \((+)^* (-) = -\), \((-)^* (+) = -\), \((-)^* (-) = +\).
concerning the variable of interest \( x \) is calculated as:

\[
\frac{\partial \dot{x}}{\partial y} = \frac{\partial (ay)}{\partial y} = a
\]

\[
\frac{\partial y}{\partial x} = \frac{\partial (f(x) \cdot x)}{\partial x} = xf'(x) + f(x)
\]

**loop polarity** \( = \) \( \text{sign} \left( \frac{\partial \dot{x}}{\partial y} \frac{\partial y}{\partial x} \right) \) \( = \text{sign} \left( axf'(x) + af(x) \right) \)

According to the definition by Mojtzhedzadeh, there are two major loops \( L_1 \) and \( L_2 \). For \( L_1 \), \( x \rightarrow f(x) \rightarrow y' \rightarrow y \rightarrow x' \), its polarity is:

\[
\text{sign} \left\{ \text{sign}(x' \rightarrow x) \cdot \frac{\partial f(x)}{\partial x} \cdot \frac{\partial y'}{\partial x} \cdot \text{sign}(y' \rightarrow y) \cdot \frac{x'}{y} \right\}
\]

\[
= \text{sign} \left\{ (+) \cdot f'(x) \cdot x \cdot (+) \cdot a \right\}
\]

\( = \text{sign} \left\{ axf'(x) \right\} \)

For \( L_2 \), \( x \rightarrow y' \rightarrow y \rightarrow x' \), its polarity is:

\[
\text{sign} \left\{ \text{sign}(x' \rightarrow x) \cdot \frac{\partial y'}{\partial x} \cdot \text{sign}(y' \rightarrow y) \cdot \frac{x'}{\partial y} \right\}
\]

\[
= \text{sign} \left\{ (+) \cdot f(x) \cdot (+) \cdot a \right\}
\]

\( = \text{sign} \left\{ af(x) \right\} \)

If we add up the gain of \( L_1 \) and \( L_2 \) in the polarity computation, we will find it is equal to the polarity defined by Richardson: \( \text{sign}(axf'(x) + af(x)) \). This shows, in the major loop case, the polarity definition by Richardson can also be interpreted as the dominance loop polarity, which is important in understanding the relationship between the structure and behavior of dynamic systems.

### 2.3.2 Principles over the Relationship Between Loop Polarity and Behaviour

As Richardson (1991) states, “it is the concept of polarity that gives the feedback loop its perceived analytic and explanatory power”. Loop polarity is an important characteristic of a feedback loop, and it qualitatively determines the properties of a loop. The negative loops are naturally considered to be associated with goal-seeking behaviour while the positive loops are related with exponential growth behaviour. It is not difficult to infer the variable behaviour with only one or two feedback loops, but the difficulty soars when the loop interactions increase. The principles that shed lights on the relationship of the loop polarities and the behavior are of
great help. Nevertheless, the pitfalls of predicting behaviour from loop polarity alone has been recognized (Richardson, 1986).

The attempt at developing the polarity and behaviour principles is first seen in (Graham, 1977). These principles are empirical and conceptual rather than theoretical and mathematical, such as negative feedback loops with a delay can exhibit oscillating behaviour, and delays usually attenuate the amplification power of a positive feedback loop, etc. Such principles help one to arrive quickly at hypotheses about the particular system structure that may cause the given behaviour. For instance, in most cases, the underlying cause of oscillations is a disturbance that propagates around a loop formed of two or more phase-lag subsystems. Another example is to increase damping. One can increase the rate of movement of a single state variable toward its steady-state value, which can diminish the propagation of a disturbance through that state variable, so that the magnitude of future oscillations decreases. A minor negative loop (a loop with only one state variable) is such a structural change. Following Graham’s work, Richardson (1995a) studied the loop polarity and the behaviour of the variable of interest, where the most significant contributions are the clear definition of the loop polarity, and establishing the calculations of the loop dominance, and the dominant polarity through the variable behaviour. Later, phase plane analysis was used to stress the importance of the system eigenvalues and to make a distinction between them and loop polarities in terms of behaviour (Güneralp, 2004a). This paper endeavours to identify what kinds of feedback loops give rise to what kinds of behaviour under what conditions. Furthermore, through learning a generic second-order linear system, Güneralp concluded that the relative locations and magnitudes of feedback loops have more significance than their polarities in the generation of system behaviour (Güneralp, 2004b).

In addition to developing these principles, formal approaches in system dynamics has been studied to identify the dominant structures that drive the model behavior in a mathematical rigorous way. “Dominant” in this context means the structure being primarily responsible for the observed behaviour over some time intervals (Richardson and Pugh, 1981). Nevertheless, this measure of dominance varies between different methods. These formal structural analysis methods are to be introduced in the following section.
CHAPTER 2. BACKGROUND AND RELATED RESEARCH

2.4 A Review of Analytical Approaches for Dominant Feedback Loop Analysis

From this section, we begin to introduce approaches in system dynamics that study the relationship between the system behaviour and its structure. These approaches perform the dominant system structure analysis to identify the leverage points, such as feedback loops and parameters. We classify them into two categories: analytical and computational. The analytical approaches are composed of pathway participation metrics and eigen-based analysis, while the behavioural method, and statistical screening fall into the computational category. The analytical approaches are first discussed within this section.

2.4.1 Pathway Participation Metrics

One methodology in formal analysis for detecting most influencing feedback loops is the pathway participation metrics (PPM) which is proposed by (Mojtahedzadeh, 1997, 2001). A pathway is defined as “a series of links connecting two system state variables” by Mojtahedzadeh (1997), the two states are not necessarily different.

Mojtahedzadeh et al. (2004) stated: “the basic behavioural building block of the PPM is a single phase of behaviour for a single variable. A single phase of behaviour for a selected variable is a time slice where the selected variable maintains the same polarity for the slope and curvature, although the magnitude of the time derivatives may change”. Therefore, there are seven patterns of behaviour that may exist within a single phase and could be represented by a combination of the selected variable’s first derivative ($\dot{x}$) and second derivative ($d\dot{x}/dt$):

- reinforcing growth, ($\dot{x} > 0$) & ($d\dot{x}/dt > 0$);
- reinforcing decline, ($\dot{x} < 0$) & ($d\dot{x}/dt < 0$);
- balancing growth, ($\dot{x} > 0$) & ($d\dot{x}/dt < 0$);
- balancing decline, ($\dot{x} < 0$) & ($d\dot{x}/dt > 0$);
- linear growth, ($\dot{x} > 0$) & ($d\dot{x}/dt = 0$);
- linear decline, ($\dot{x} < 0$) & ($d\dot{x}/dt = 0$);
equilibrium, \( \dot{x} = 0 \) & \( \dot{d}/dt = 0 \).

The above behaviour classification resembles the ABP a lot as both of them make use of the derivatives to formulate various patterns (i.e., only different terms are used to represent the same pattern, exponential versus reinforcing, and logarithmic versus balancing). A minor difference lies where the ABPs do not distinguish between a growth or a decline behaviour. The core idea of PPM is described as follows: It sets out with the selected variable of interest, e.g., \( x_1 \) in Figure 2.13, and calculates how much its net flow (\( \dot{x}_1 \) or \( dx_1/dt \)) could change given a small change in (\( dx_1 \)), \( d\dot{x}_1/dx_1 \), is called the “total pathway participation metrics”. The total pathway participation metrics for the variable \( x_1 \) is then partitioned among pathways coming into the net flow, Loop1 and pathway2. The most influential structure is defined as the one whose partition is the largest in magnitude and has the same sign as the total changes in the net flow \( \dot{x}_1 \) when driven by an infinitesimal change in the state variable. Pick the variable at the tail of the identified influential structure (\( x_1 \) or \( x_2 \)) as the new variable of interest and repeat the process until a feedback structure is discovered.

We take a close look at the PPM and suppose the variable of interest is \( x_k \), the equation of \( k^{th} \) state may look like: \( \dot{x}_k = f_k(x_1, x_2, ... x_n; P) \), where \( P \) are the parameters of the system. Taking the derivative of the net change in the state variable of interest, \( x_k \), with respect to the state variable of interest, \( x_k \), yields the total pathway participation metrics:

\[
\frac{d\dot{x}_k}{dx_k} = \frac{\partial f_k}{\partial x_1} \frac{\dot{x}_1}{x_k} + \frac{\partial f_k}{\partial x_2} \frac{\dot{x}_2}{x_k} + ... + \frac{\partial f_k}{\partial x_k} \frac{\dot{x}_k}{x_k} + ... + \frac{\partial f_k}{\partial x_n} \frac{\dot{x}_n}{x_k}
\]

or simply:

\[
\frac{d\dot{x}_k}{dx_k} = \sum_{i=1}^{n} \frac{\partial f_k}{\partial x_i} \frac{\dot{x}_i}{x_k} \quad (\text{for} \ \dot{x}_i \neq 0)
\]

Each term in Equation 2.16 represents a pathway leaving the \( i^{th} \) state variable and going into \( k^{th} \) state variable except the one with \( i = k \) represents a minor feedback loop (see Loop1 in Figure 2.14). Then, take the state variable at tail (i.e., the causal variable) of the identified pathway, repeat the procedure, we find another most influencing pathway. This process continues iteratively until the identified dominant pathways form a feedback structure. It is worth noting that the identified dominant structure can contain more than one feedback loop, with addition of other pathways. The assessment of the relative importance of a pathway \( i \) (or a
Figure 2.12 – Seven behaviour modes of PPM Analysis
minor feedback loop when \( i = k \) coming into the variable of interest \( x_k \) is formulated:

\[
\frac{\partial f_k}{\partial x_k} \dot{x}_k = \frac{\partial f_1}{\partial x_1} \dot{x}_1 + \frac{\partial f_2}{\partial x_2} \dot{x}_2 + \cdots + \frac{\partial f_n}{\partial x_n} \dot{x}_n
\]

(2.17)

where the denominator is determined by the sum of the absolute value of the strength of minor loops and pathways.
2.4.1.1 Logistic growth model

A first order nonlinear logistic growth model used by another researcher (Mojtahedzadeh, 1997) is adopted here in order to brief the PPM method. Its equations are outlined as follows:

\[ \dot{x} = ix - ox \]
\[ x_0 = 0.01 \]
\[ ix = a \cdot x \]
\[ ox = b \cdot x^2 \]
\[ a = 0.4 \]
\[ b = 0.2 \]

In addition to the knowledge of the stock flow diagram in Figure 2.15, we know that there are two minor feedback loops, a positive loop 1 and a negative loop 2 (as long as \( x \) is positive). Taking derivative of \( \dot{x} \) with respect to the state variable \( x \), yields:

\[ \frac{d\dot{x}}{dx} = \frac{d(ix - ox)}{dx} = a - 2bx \]
\[ = 0.4 - 0.4x \]

The first term in the above equation represents the strength of loop 1 while the second term refers to the strength of loop 2. Furthermore, we have:

\[ PPM \ for \ loop \ 1 = \frac{0.4}{0.4 + |x| - 0.4x} \]
\[ PPM \ for \ loop \ 2 = \frac{0.4}{0.4 + |x| - 0.4x} \]
CHAPTER 2. BACKGROUND AND RELATED RESEARCH

From the computation of PPM, we know the dominant loop is determined by the value of $x$. A simulation is run by Vensim to and its behaviour is plotted in Figure 2.16. In this figure, the red vertical line indicates the point where the dominance shift from loop 1 loop 2. As $x$ is less than 1, the PPM of loop 1 is greater than that of loop 2, which suggests $\frac{dx}{dx}$ is positive. This is also confirmed by the steepening behaviour in the initial stage. On the contrary, when $x$ increases further, the PPM of loop 2 decreases to make $\frac{dx}{dx}$ become negative. As a result, the trajectory of $x$ becomes a balancing growth in the later stage, which can be seen in Figure 2.16.

![Figure 2.16 – Behaviour of a logistic growth model](image)

2.4.1.2 Summary

PPM can also be applied to a non-state variable of interest. It has to calculate its net change which follows the same procedures (elaborated in (Mojtahedzadeh, 1997)), and we omit this part here. One strength of the PPM is its capability of dealing with nonlinear systems. Another advantage is that it can identify a dominant feedback structure without a candidate loop set required. This is particularly useful when no systematic loop selection algorithm was provided. There are, however, three main drawbacks associated with the PPM: (1) The emphasis on identifying a single “dominant” structure in (Kampmann and Oliva, 2008). In reality, it is the interaction between loops rather than one loop that will give rise to the observed behaviour. (2) It considers only the partial system structure, the local maximum pathway, rather than the global system. (3) It is not suitable for oscillatory systems. This is a significant limitation,
given the prevalence and importance of oscillation in system dynamics analysis.

2.4.2 Eigenvalue Elasticity Analysis

In control theory, the behaviour of a system is usually characterized by eigenvalues, eigenvectors, poles, zeros, gain, and phase (Ogata, 1997). The eigen-based approach is derived from the control theory, involving eigenvalue elasticity analysis (EEA) and eigenvector analysis.

The relationship between the eigenvalue and system behaviour is intensively discussed (Forrester, 1982; Medio and Lines, 2001; Ogata, 1997). Eigenvalue analysis was first introduced into feedback loop analysis by Forrester (1982). As previously discussed in the solution to the linear system, we find the eigenvalues have substantial influence in determining the behaviour (see Equation 2.7). Thereby, Forrester proposed the eigenvalue elasticity analysis, to use eigenvalue elasticity, which assesses the percentage change in the eigenvalue for a given percentage change in the structural component (parameter or link in the system gain matrix). The elasticity is a dimensionless measure that relates the impact of the change in the cause to the magnitude of the effect and cause. It is formulated in Equation 2.18.

\[ \varepsilon_i = \frac{\delta \lambda_i / \lambda_i}{\delta g / g} = \frac{\partial \lambda_i}{\partial g} \frac{g_i}{\lambda_i} \]  \hspace{1cm} (2.18)

where \( g \) represents the value of the structural component, e.g., a parameter, a feedback loop gain or a compact link gain. The compact link is a series of links which link a state to a flow, and its gain corresponds to an entry in the system gain matrix. An elasticity evaluates the percentage changes in the eigenvalue in response to the percentage changes in the structure component. Particularly, the eigenvalue sensitivity (evaluates the partial derivative only) with respect to the compact link gain, e.g., \( A(p, q) \), can be calculated analytically in Equation 2.19:

\[ S_{\lambda_i}^{\lambda_i} = \frac{\partial \lambda_i}{\partial A(p, q)} = \ell_{ip}^T \times r_{iq} \]  \hspace{1cm} (2.19)

where \( r_i \) and \( \ell_i^T \) are the right and left eigenvectors of \( A \), and \( p, q \) indicate the components in each eigenvector (see Appendix B for the proof). The structure components with larger elasticities are considered more important to the behaviour mode in question. For the complex elasticity which is generated by the complex eigenvalue from Equation 2.18, the interpretation is not that simple. The real part of the elasticity shows the effects of altering the strength of a structural component on the natural frequency of the oscillation behaviour, while the imaginary
part expresses the effect on the damping mode. The magnitude of the elasticity measures the overall sensitivity of the oscillation mode to the structure. A thorough discussion regarding complex elasticity can be found in (Saleh and Davidsen, 2001). In addition to the original elasticity definition, there are other ways to define the complex eigenvalue elasticity. For example, the measure used by (Güneralp, 2006a) is:

\[
\begin{align*}
\varepsilon_{re} &= \frac{\partial \text{Re}(\lambda)}{\partial g} \frac{g}{\text{Re}(\lambda)} \\
\varepsilon_{im} &= \frac{\partial \text{Im}(\lambda)}{\partial g} \frac{g}{\text{Im}(\lambda)}
\end{align*}
\]

(2.20)

where \(\varepsilon_{re}\) refers to the percentage changes in the oscillation amplitude in response to the percentage changes in the structure \(g\); \(\varepsilon_{im}\) shows how much will change in the damped frequency. Another measure used by Kampmann (1996) and Saleh (2002):

\[
\begin{align*}
\varepsilon_{re} &= \frac{\partial \text{Re}(\lambda)}{\partial g} \frac{g}{|\lambda|} \\
\varepsilon_{im} &= \frac{\partial \text{Im}(\lambda)}{\partial g} \frac{g}{|\lambda|}
\end{align*}
\]

(2.21)

To indicate the sign of each component, the above measure is complemented in (Phaff et al., 2006) by:

\[
\begin{align*}
\varepsilon_{re} &= \frac{\partial \text{Re}(\lambda)}{\partial g} \frac{g}{|\lambda|} \text{sign}(\text{Re}(\lambda)) \\
\varepsilon_{im} &= \frac{\partial \text{Im}(\lambda)}{\partial g} \frac{g}{|\lambda|} \text{sign}(\text{Im}(\lambda))
\end{align*}
\]

(2.22)

An influence measure: \(\mu_{re} = (\partial \lambda_{re}/\partial g) g, \mu_{im} = (\partial \lambda_{im}/\partial g) g\) is proposed by Kampmann and Oliva (2006) to avoid the technical difficulties when eigenvalues are close to zero. The interpretation is similar as that to Equation 2.20.

In a system, there are a number of behaviour modes which correspond to different eigenvalues. The most influential structure will vary in different behaviour modes. Therefore the dominant structure is addressed in terms of a certain behaviour mode. According to the original EEA in (Forrester, 1982), only the dominant eigenvalue will be investigated. “Dominant” indeed depends on the context, but usually it refers to the eigenvalue with greatest real part. Once it is chosen, the elasticity to the compact link gain is obtained by the result from Equation 2.19 multiplying their ratio. The paths between states that have large-magnitude eigenvalue elasticities are particularly important within the corresponding behaviour mode. Sometimes, these important links happen to form feedback loops in the model. Otherwise, one can consider
the loops which involve those important links as dominant loops. Although the dominant loop identification procedure is not that formal, the framework of EEA is formed for further development and improvement.

A breakthrough in EEA came from Kampmann who proposes an idea of the “independent loop set” (ILS) to analytically calculate the eigenvalue elasticity with respect to feedback loops (Kampmann, 1996). The mechanism of how it works is to be presented in a separate paragraph below. Numerous papers study the loop elasticity and its applications on the basis of ILS’s development. A comprehensive introduction to EEA is first presented in Saleh and Davidsen (2000) where it summaries EEA to be a two-stage approach. In the first stage, one ranks the eigenvalues by their values (particularly, complex eigenvalues are usually of great importance), and selects the dominant eigenvalue(s). In the second stage, dominant loop(s) are picked by their loop elasticities to the dominant eigenvalue(s). Later, an automated version of EEA is developed to facilitate the implementation of EEA (AbdelGawad et al., 2005). Meanwhile, a variation of EEA is proposed to take account of all eigenvalues instead of one dominant eigenvalue when calculate the loop elasticity (Güneralp, 2006b). This is achieved by comparing the contributions of behavior modes ($\lambda_i$) to the changes in slope of the variable of interest. Assume the variable is $x_p$, Equation 2.23 shows how to measure such a contribution from one behaviour mode.

$$\Delta S_{ip} = S_{ip}^{dt} - S_{ip}^0$$

$$= \alpha_i^0 e^{\lambda_i dt} r_{ip} - \alpha_i^0 e^{\lambda_i 0} r_{ip}$$

$$= \alpha_i^0 e^{\lambda_i dt} r_{ip} - \alpha_i^0 r_{ip}$$  (2.23)

Then sum up the absolute values of all $\Delta S_{ip}$ to determine each mode’s relative contribution according to their ratio in Equation 2.24.

$$c_{ip} = \frac{\Delta S_{ip}}{\sum_{m=1}^{n} |S_{mp}|}$$  (2.24)

This method uses an aggregated loop elasticity which adds up a weighted elasticity of each behaviour mode in terms of its contribution to the behaviour of the variable. It provides an alternative perspective to explore EEA. A chaotic system, the Lorenz model, is studied in (Kampmann and Oliva, 2006) where the authors point out that EEA fails when eigenvalues are not sufficiently stable (i.e., eigenvalues change dramatically in a short interval) to explain behaviour over a period of time. Apart from the feedback structure, there are also some studies
concentrating on identifying system parameters that the behaviour of interested is most sensitive to. For example, closed-form\textsuperscript{3} analytic functions relating the eigenvalues to all parameters in a linear model can be developed, as the eigenvalues are determined by the gain matrix which can be further represented by the parameters. A procedure (implemented in MATLAB) to calculate the eigenvalue sensitivity with respect to the parameter is proposed in (Saleh et al., 2005). The sensitivity analysis results to a linearized version of the classical market growth model is presented and analyzed.

An illustration of the EEA method can be found in Güneralp (2006b) where two test models are presented to demonstrate its application step by step from computing the system eigenvalues to the loop elasticity.

\subsection{Independent loop set: a solution to feedback loop analysis methodologies}

In the introduction to EEA, we mentioned, before the advent of ILS, a lack of a formal method to identify a candidate feedback loop set exists in the feedback loop analysis field. It is inappropriate to adopt a complete set of feedback loop in a model as a candidate loop set. The reason can be found in (Kampmann, 1996), “In a model with $n$ state variables and $p$ auxiliary variables under the condition that it is maximally connected, the number of loops grows to $2^{np} \ast (n - 1)!$”. Even if in realistic models, which have far fewer links, it is still too time-consuming to analyze all of them. The ideal loop set therefore, should represent the core feedback structure of the system with an appropriate number of loops, thus are applicable to all feedback loop analysis methodologies.

Another particular desire of developing a feedback loops selection algorithm arises from the loop elasticity computation (short for the eigenvalue elasticity with respect to the loop gain). Although the loop elasticity is possible to be analytically solved by representing the eigenvalue in an expression of loop gains (Goncalves et al., 2000), this possibility is slim. Moreover, there is no closed-form solution to the eigenvalue when the system has more than five state variables.

For high order system, the eigenvalues has to be computed by iterative algorithms (Demmel, 1997). An alternative means to compute the loop elasticity is by taking the steps shown in

\textsuperscript{3} If can be expressed analytically in terms of a bounded number of certain “well-known” functions. Typically, these well-known functions are defined to be elementary functions-constants, one variable x, elementary operations of arithmetic ($+ - \times \div$), $n$th roots, exponent and logarithm (which thus also include trigonometric functions and inverse trigonometric functions).
Figure 2.17. The computation starts from the compact link elasticity, through the pathway elasticity to the causal link (an individual link connects two nodes) elasticity by making use of the chain rules. However, the loop elasticity cannot be solved due to the under-determined equation system. The reason is explained as follows. Assume we have obtained the causal link elasticity. A relationship between the causal link elasticity and loop elasticity is: 

\[ \text{the causal link elasticity equals to the sum of all the loop elasticities who encompass this causal link.} \]

This is formulated by introducing the **directed cycle matrix** \( C_r \) in Equation 3.2:

\[
\begin{bmatrix}
\varepsilon_{k1}^i \\
\varepsilon_{k2}^i \\
\vdots \\
\varepsilon_{kN}^i
\end{bmatrix} = C_r
\begin{bmatrix}
\varepsilon_{c1}^i \\
\varepsilon_{c2}^i \\
\vdots \\
\varepsilon_{cM}^i
\end{bmatrix}
\] (2.25)

where \( \varepsilon_{ki}^i \) and \( \varepsilon_{cj}^j \) express the \( k^{th} \) eigenvalue elasticity with respect to \( i^{th} \) link and \( j^{th} \) loop respectively, and \( C_r \) is a N-by-M matrix. \( C_r(i, j) \) is 1 if link \( i \) is a component of loop \( j \), 0 otherwise. The solution to \( \varepsilon_{ki}^i \) is determined by the rank\(^4\) of \( C_r \). A problem follows immediately, in most cases, feedback loops outnumber the edges, i.e., \( \text{Rank}(C_r) \leq N < M \), the

\(^4\) The column rank of a matrix is the maximum number of its linearly independent column vectors. The row rank of a matrix is the maximum number of its linearly independent row vectors. The fundamental theorem of linear algebra states that the column rank is always equal to the row rank.
system is then under-determined (unknowns outnumber the equations), so the loop elasticity will have no solution. The idea of “independent loop set” is consequently proposed to solve this problem by Kampmann (1996). “Independent feedback loops” actually refers to the loop matrix in Equation 3.2 to be a full rank matrix. ILS is formed by accepting a feedback loop into the set only if it contains at least one edge not included in the previously accepted loops. Accompanied with the ILS, an important proposition is yielded: In a strongly connected digraph (any pair of nodes could reach each other), the number of independent loops (M) in ILS is \( e-n+1 \), where \( e \) is the number of edges, \( n \) is the number of nodes. For example, the floating goal model in Figure 2.3, there are 6 edges and 4 vertices in the strongly connected digraph, whereas the two parameters \( sat \) and \( gat \) are not taken account as they are not strongly connected. Thus \( 6 - 4 + 1 = 3 \) is the number of independent feedback loops, and it is also equivalent to the possible loops can be found in this example. However, in a more complex system with a great number of possible feedback, applying this rule will reduce the size of feedback loops to arrives at a manageable number. We can see that \( rank(C_r) = M \), and it therefore assures a unique solution to the loop elasticity. Details of the complete procedure of how to obtain the loop elasticity from the compact link elasticity is presented in Appendix C.

An implementation of ILS is simplified by an algorithm named shortest independent loop set (SILS) proposed by Oliva (2004). The key idea of its construction rule is introducing a new cycle that makes the minimal contribution of new edges into the ILS. The SILS algorithm is constrained to work with geodetic cycles. Given two vertices, a geodetic cycle is a shortest cycle in which these two vertices are involved. A complete geodetic cycle list has to be compiled for SILS to use (for its algorithm, see Oliva (2004)). Algorithm 1 shows the pseudo code of the SILS, and all operators are in ordinary matrix algebra. \( lps \) is a geodetic cycle list. It can be thought of as a two dimensional array, and each array records the loop vertices in sequence. \( A \) is a cycle partition of an adjacency matrix. The adjacency matrix representation of a digraph is a square matrix with the size of all vertices, where each row (and column) represents a vertex. The entries are restricted to be one, if there is a link between them, and zero, if no link exists between them. \( B \) is an initially empty adjacency matrix to keep track of the edges which have been introduced into the ILS up to date. \( E \) is a three dimensional matrix, which records the geodetic cycles by their edges.

It is worth noting that SILS is not unique if there are two or more “shortest” path of equal
Algorithm 1 Pseudo code of SILS derived from a geodetic cycle list

```
sils ← SILS (lps, A)  
B ← zeros(A)          
k ← 1                 
S ← lps               
for each vertex i ∈ A do
    A(i, i) = 0         
end for
for each loop i ∈ lps do
    len ← length(lps(i))  
    for j=1: len-1 do
        E(i, lps(i, j), lps(i, j+1)) = 1
    end for
    E(i, lps(i, len), lps(i, 1)) = 1
end for
while S ≠ ∅ do
    for j ∈ S do
        c(j) ← count((E(j, :, :) - B) > 0)  
        if c(j) == 0 then
            S ← S - S(j)
        end if
    end for
    m ← i ∈ min(nonzero(c(i)))  
    n ← m(1)  
    if n ≠ ∅ then
        B ← B + E(n, :, :)  
        sils(k) ← S(n)  
        k ← ++  
        S ← S - S(n)  
        c(n) = 0
    end if
end while
```
length between two variables. For various models, the SILS contains the loops identified as most influential by PPM (Oliva and Mojtahedzadeh, 2004). This result verifies the hypothesis that the SILS captures the core dynamics. Moreover, it shows how choosing the SILS allows for relatively more intuitive interpretation of the loops.

2.4.2.2 Summary

EEA is mathematically rigorous, and capable of providing an unambiguous and complete measure of the influence of the entire feedback structure on all behaviour modes. One key strength of EEA is it can work as a “filter”, i.e., it provides a possible way to get rid of the undesirable behaviour mode (e.g., oscillation) and keep the desired one. We can “pick up” the eigenvalue of interest and modify it by altering its high leverage structure points (it is also necessary to consider if other eigenvalues are extremely sensitive to such leverage points). Another merit of EEA is that it is good at handling oscillation, important information such as damping ratio, damping frequency and natural frequency can be assessed and used to analyze the stability of the system (Forrester, 1982). However, there are some drawbacks. One is that the concepts of eigenvalues and elasticities are rather abstract and counterintuitive (Ford, 1999) since the eigenvalue is a highly abstract concept in mathematics and do not link to any system structures. Another more fundamental weakness is that EEA manipulates the behaviour to the system-wide perspective instead of fine-tuning an individual variable.

2.4.3 The Eigenvector (Weight) Analysis

Eigenvector analysis is still in its formative stage, but it has attracted much attention from system dynamists (Goncalves, 2006, 2009; Güneralp, 2004a; Saleh et al., 2006, 2010). Besides eigenvalues, eigenvectors also contribute to the behaviour. As we can see in Equation 2.7, the eigenvector is a coefficient of each behaviour mode. It hence shows how much a behaviour mode is expressed in a particular state variable.

A number of researchers have attempted to develop eigenvector analysis methods to identify dominant system structures. It is first seen in (Goncalves, 2006) that the computation of sensitivity, the change in behaviour of a state variable due to the change in loop gain, has been
brought to focus:

\[
\frac{\partial x(i)}{\partial g_k} = \frac{\partial(r_1 e^{\lambda_1 t} \alpha_1^0 + ... + r_n e^{\lambda_n t} \alpha_n^0)}{\partial g_k} / \frac{\partial g_k}{\partial g_k} = \frac{\partial r_{ji}}{\partial g_k} e^{\lambda_{ji} t} \alpha_1^0 + ... + \frac{\partial r_{ni}}{\partial g_k} e^{\lambda_{ni} t} \alpha_n^0 + \frac{\partial \lambda_{ji}}{\partial g_k} r_{ji} e^{\lambda_{ji} t} \alpha_1^0 + ... + \frac{\partial \lambda_{ni}}{\partial g_k} r_{ni} e^{\lambda_{ni} t} \alpha_n^0
\]

(2.26)

Two differential terms are derived by taking the derivative of each behaviour mode with respect to the loop gain:

1. The contribution associated with the eigenvector sensitivity, \( j^{th} \) component in \( r_i \), with respect to the loop gain, i.e., \( \partial r_{ji} / \partial g_k \).

2. The contribution associated with the eigenvalue sensitivity, i.e., \( \partial \lambda_{ji} / \partial g_k \).

The state sensitivity shows not only the eigenvalue but also the eigenvector plays a role in influencing the behaviour. The contribution from the eigenvector is constant while that from the eigenvalue is time-dependent. As a result, compared with the influence from the eigenvector, the eigenvalue sensitivity will dominate the behaviour as time goes.

A method to analytically calculate the eigenvector sensitivity is proposed in (Goncalves, 2006). In addition, a linear system the inventory-workforce model is presented to illustrate this method. In the example, the eigenvalues and eigenvectors of the system are first represented by functions of loop gains and system parameters, in forms as:

\[
\lambda_i = f_i(g_1, ..., g_m, p)
\]

\[
r_i = h_i(g_1, ..., g_m, p)
\]

(2.27)

where \( p \) is a vector with parameters. The eigenvector sensitivity with respect to the loop then can be obtained by simply taking partial derivative of \( r_i \) with respect to the particular loop gain in Equation 2.27. This eigenvector analysis method is actually an extension of the eigenvalue analysis and appear in (Goncalves et al., 2000), it thus share the similar problems. First, representing the eigenvalue and eigenvector in the form of the loop gains, i.e., Equation 2.27 can not always be obtained, and it is quite depended on the model structures. Second, the eigenvalue and eigenvector cannot be analytically solved when the number of system state variables exceeds five. Third, a more crucial problem in this approach is that the eigenvector is
not fixed, and it has “a degree of freedom”, the eigenvector then is impossible to be uniquely represented by the loop gain, so is the eigenvector sensitivity. In addition, the eigenvector elasticity is defined by (Goncalves, 2006):

$$
\varepsilon_{ki}^r = \frac{\partial r_i}{\partial g_k} |g_k| \left/ \|r_i\| \right|
$$

(2.28)

where $|g_k|$ is the absolute value of the loop gain, and $\|r_i\|$ is the Euclidean norm of the eigenvector $r_i$. Each component of the eigenvector elasticity is normalized by the magnitude of the eigenvector. As the eigenvector varies by a scalar, it is tricky to define and calculate the eigenvector sensitivity and elasticity. If one always normalizes the eigenvector to be $\|r\| = 1$, the elasticity in Equation 2.28 becomes $\frac{\partial r_i}{\partial g_k} |g_k|$.

Another contribution to the eigenvector analysis came from Saleh et al. (2006, 2010) where they termed their approach Dynamic Decomposition Weight (DDW) analysis. DDW concerns what happens to the weights when changes are made to the system structures. The weight is defined in Equation 2.29, and $w$ comprises an eigenvector component:

$$
x_i(t) = r_1 \alpha_0^1 e^{t\lambda_1} + r_2 \alpha_0^2 e^{t\lambda_2} + \cdots + r_n \alpha_0^n e^{t\lambda_n}
$$

(2.29)

The weight represents how significant a certain behaviour mode can impose on a variable’s behaviour. Compared with the eigenvector analysis alone, it is a more comprehensive analysis as the weight consists the eigenvector as part of it. Saleh et al. (2010) explore the policy design space by assessing the influence of model parameters on the weight and identify leverage points by focusing on parameters that most affect the weights of the behaviour modes for the variable of interest. They examine the weight elasticity in the linear inventory-workforce model where the weight elasticity, $\frac{\delta w_{ji}}{\delta g_j}$, is calculated numerically by giving a slightly perturbation to the structure component $g_j$, then compare the new values to the base case. Moreover, the pseudo code of implementing the numerical procedure of the weight elasticity (with respect to the parameter) computation is presented. Their analysis highlights the fact that changes in parameters affect both the eigenvalues and the weight, i.e., their relative presence in the behaviour of a given state variable. The role of eigenvector analysis is thus emphasized in order to explore a wider policy space and generate insightful policy interventions.
2.4.3.1 Model Linearization

One concern of the eigen-based analysis is it is not directly applicable to nonlinear models. In a nonlinear system, the gain matrix depends on the current state of the model, which leads to the gain matrix change over time. This also indicates both eigenvalues and eigenvectors vary over time. In order to extend the eigen-based analysis to nonlinear models, it is necessary to linearize these models. A most frequently used method is the Taylor Series expansion. We first divide the simulation time into consecutive intervals. The Taylor Series expansion is applied at each time interval. The size of the analysis interval is sufficiently small to be able to assume that the gain matrix remains constant during each interval. A nonlinear system is expressed in Equation 2.1. Assume current system states are $x^*$, we have:

$$\dot{x}^* = F(x^*)$$

A small perturbation vector, $[\delta x_1, \delta x_2, ..., \delta x_n]^T$, is then added to the state vector. Using Taylor Series expansion to unfold the equation:

$$\dot{x}^* + \delta \dot{x} = F(x^* + \delta x) = F(x^*) + \frac{1}{1!} \left\{ \left. \frac{\partial F}{\partial x_1} \right|_{x^*} \delta x_1 + ... + \left. \frac{\partial F}{\partial x_n} \right|_{x^*} \delta x_n \right\}$$

$$+ \frac{1}{2!} \left\{ \left. \frac{\partial^2 F}{\partial x_1^2} \right|_{x^*} \delta x_1^2 + ... + \left. \frac{\partial^2 F}{\partial x_n^2} \right|_{x^*} \delta x_n^2 \right\} + ... $$

(2.31)

where we omit the higher order terms. The more orders incorporated into the expansion, the more precise result it yields. For the linear system, the first order term represents the exact amount which is induced by the perturbation. In general, compared with the first order term, the amount of higher order terms are very small and can be ignored. Therefore the linearization takes only the first order derivative into consideration. In nonlinear systems, the first order term will contain the state variables $x^*$ which is considered as a constant vector over a small time interval. Therefore, the first order derivative matrix has to be dynamically assessed with state variables updated at each analysis point. Moreover, in some special cases, higher order terms are required, cases can be found in (Strogatz, 2000). Second order terms of the Taylor Series expansion are considered in approximating nonlinear models in (Saleh and Davidsen, 2007).

Many of the EEA applications based on model linearization have been carried out in (Güneralp, 2006b; Kampmann and Oliva, 2006; Saleh and Davidsen, 2000). Though the linearization is
an approximation method, it paves the way for the eigen-based analysis approaches to analyze the nonlinear systems.

### 2.4.3.2 Summary

The eigenvector analysis attempts to improve the EEA method by considering how much an eigenvalue or behaviour mode is expressed in a particular state variable. Meanwhile, the eigenvector analysis shares similar limitations as EEA. For example, the application to the nonlinear systems requires linearization of the system at every step of the simulation, calculations of the gain matrix, and numerical evaluation of eigenvalue and eigenvector, which is computationally heavy and difficult. Existing eigenvector analysis is still in its early stage and there is a lack of analytical method to compute the eigenvector elasticity with respect to various structural elements. Despite current limitations and challenges, it is a promising tool and provides a useful step on the analysis of how structure influences behaviour. It is advisable to consider both EEA and eigenvector analysis to afford an assessment of the overall impact of a policy on a variable performance.

### 2.5 A Review of Computational Approaches for Dominant Feedback Loop Analysis

This section presents a survey of computational approaches for dominance analysis, including the behavioural method, and statistical screening. The details of these methods are unfolded as follows.

#### 2.5.1 The Behavioural Method

Previous methodologies are analytical solutions to identifying the dominant structures, we now start to introduce simulation-based approaches. The first one is the behavioural method proposed by Ford (1999). It is a formalized version of “dynamic hypothesis”, which means trying to deactivate every candidate loop and evaluating its significance to the behaviour of interest. The behavioural method is a further development of the idea of the loop isolation technique. Ford made the approach more rigorous and systematic from several aspects.
- Strengthen the definition of the behaviour mode. The method formulates the atomic behaviour patterns in a similar way as PPM but expressed the behaviour pattern in a compressed form. Therefore, it reduces to be three atomic behaviour patterns which we have discussed in section 2.2. This method uses the change of the absolute value of the net flow (the change of the magnitude of the slope) to distinguish the behaviours. These behaviour modes are especially important in the linear systems whose behaviour is a superposition of the ABPs and oscillations.

- Adopt the ABP to partition the time interval and carry out the analysis within each time interval individually. The change of the behaviour pattern indicates the potential shift of the dominant loop, which establish a solid foundation for performing the analysis.

- Clarify the procedure of deactivating (isolating) a feedback loop. The key is the concept of the control variable. It is a variable not in any other loops and uniquely belong to one loop. The control variable is manipulated to evaluate the loop influence via the loop deactivation process. To deactivate this loop, the control variable is set to be constant in the scope of its loop but maintain unaffected in all other loops. The constant is which the variable has in the reference simulation at the deactivating point. Ford also named it as default value.

- Outline the criteria for determining the dominant loop. If deactivating a feedback loop varies the behaviour mode in the reference model, it is attributed to be a dominant loop.

The procedure of the behavioural method is as follows:

1. Identify the variable of interest and simulate its behaviour over time.

2. Identify the time interval during which the behaviour of variable of interest display only one atomic behaviour pattern.

3. Use the feedback structure of the model to identify the feedback loops.

4. Choose one of the feedback loops as a candidate loop, and identify the control variable to deactivate the loop.

5. Simulate the behaviour of variable of interest over the reference time interval with the candidate loop deactivated. The atomic behaviour pattern of the variable of interest is
6. In each time interval, if the atomic behaviour pattern is inconsistent with which in the reference model, the candidate loop is a dominant loop.

7. After deactivating all the individual candidate loops and if no dominant loop is identified, multiple loops are deactivated at the same time to identify the co-dominant loops.

8. Apply the above 4 steps in all the time phases and find out dominant loops in each phase.

A recent improvement in the behavioural method is from Phaff (2008) where an extended version of the behavioural approach was introduced. It is referred to as the Generalized Loop Deactivation Method (GLDM). It improves the method from the following two aspects:

- Make use of the shortest independent loop set and let it be the candidate loop set to be analyzed. In the original version of the behavioural method, no method is mentioned to select the candidate feedback loops. For a small model, the loops can be partitioned by observation. However, in a big system with a great number of feedback loops, it is impossible to fully analyze every single loop. Consequently, it is necessary to limit the number of feedback loops and use a certain strategy to pick loops out. SILS is such a strategy which does not only control the number of loops at a reasonable size but also captures the core dynamics, i.e., contains dominant feedback loops (Oliva and Mojtahedzadeh, 2004).

- Deactivate a feedback loop by fixing its unique edge instead of the control variable. Like the control variable, a unique edge is an edge that belongs to only one feedback loop. It is easy to verify that a control variable suggests two unique edges, hence identifying a unique edge is a less strict constraint for a feedback loop. This is an improvement which makes the behavioural method applicable to a much wider range of models. In conjunction with the regulation of how to set up the candidate loop set, GLDM offers us an opportunity to automate the behavioural method.

The behavioural approach as well as its extended version is explicit and applicable to both linear and nonlinear systems for the analysis of any selected variable. It identifies simultaneous domination by multiply loops and the procedure is intuitive appealing and simple. However,
it also has some weakness. Firstly, a comparison between the behavioural method and EEA is conducted in (Phaff et al., 2006), where the authors point out that the restriction of the analysis to separate time intervals limits its explanatory power for an oscillating model. Secondly, the linear behaviour is sometimes difficult to be detected. The reason is that ABP may not be strict zeros throughout an interval of genuine linear behaviour. This can be caused by the precision of the simulation results.

2.5.2 Statistical Screening

Statistical screening is proposed by Ford and Flynn (2005) to describe a structured process to assist in identifying high-leverage parameters and structures. It is particularly useful to identify which parameters are important when a large number of highly uncertain parameters exist in the model. As statistical screening is a simulation based method, one challenge is to design a small number of simulations that could be executed with limited computer resources. Latin Hypercube Sampling (LHS) is reported to be the most efficient design for computer with large numbers of uncertain parameters (McKay et al., 2000) after comparing with two other typical sampling methods: random sampling and stratified sampling. Before running LHS, each parameter’s range and distribution need to be assigned. Various parameter combinations are generated through LHS for every simulation. Furthermore, previous experiments show 50 runs should be sufficient to start with (Ford and Flynn, 2005). To verify this, one can run 100 simulations and compare the results. In light of the simulation results, the relative importance of each parameter is displayed by calculating the correlation coefficients between the models main output (the variable of interest) and the values assigned to each parameter. The correlation coefficient is calculated as:

\[ r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}} \]  

(2.32)

where \( \bar{X} \) and \( \bar{Y} \) are the sample means of \( X \) and \( Y \), and the subscript \( i \) refers to \( i^{th} \) simulation. Value \( r \) ranges from \(-1\) to \(+1\). It measures the strength of the linear relationship between two variables without accounting for other variables that might be influential. The correlation coefficient is \(+1\) in the case of a perfect positive (increasing) linear relationship (correlation), \(-1\) in the case of a perfect decreasing (negative) linear relationship (anticorrelation), and the value between \(-1\) and \(+1\) indicating the degree of linear dependence between the variables. In
particular, 0 means these two variables have no relationship and they are linearly independent. The closer the coefficient approaches to either $+1$ or $-1$, the stronger the correlation between the variables.

There are six steps in performing the statistical screening analysis summarized in (Taylor et al., 2010b):

1. Select a number of model parameters and a variable of interest for analysis. Determine ranges of possible parameter values based on an understanding of the real system.

2. Perform statistical screening of the model to calculate correlation coefficients for the selected exogenous model parameters using Equation 2.32. Plot the correlation coefficients over time in a graph.

3. Plot the behaviour of the variable of interest. Pick a time period where one would like to investigate for more insights.

4. Examine the correlation coefficients and create a list of high-leverage parameters. The list contains the parameters whose correlation coefficients are highest absolute values within the time interval under focus.

5. Identify the high-leverage model structure(s) for each parameter identified in the previous step. These structure(s) are directly connected to the high-leverage parameter.

6. Use additional structure-behaviour analysis methods to explain how the the causal structure gives rise to the resulted behaviour, such as verbal reasoning, behavioural analysis. This step verifies and consolidates the analysis results, and helps to better understand the dynamics in the system.

One strength of this approach is most of the above procedure can be implemented through the Vensim software, e.g., parameters range and distribution set-up, parameter sampling (LHS), simulations, and data export for correlation coefficient calculations. Moreover, if the data is exported to the Excel, one can calculate the correlation coefficient by using its built-in function in the statistical function category.

Finally, it is possible that the correlation coefficients may fail to recognize a critical parameter for the reason that the distribution of the uncertainty is not the same as one assign. If one senses
a parameter significance is underestimated by the correlation coefficient, one should examine a scatter plot between this particular parameter and the variable of interest. In the scatter plot, the coordinate accommodates the parameter and the variable of interest. A pattern would take on once we add all the points in the plot, a line or just randomly scattered. However, if one would like to examine the scatter plots for a large number of parameters, it is advised to use statistical software such as S-PLUS to save efforts.

2.6 Summary

This chapter has outlined a number of important topics which are fundamental to the research delivered in this dissertation. A broad ranging discussion covering the behaviour classification, structure characterization and formal structural analysis methodology are presented. Initially, typical behaviours such as reinforcing, balancing and oscillatory behaviour are introduced as a means of behaviour classification. Two approaches of formulating the definitions to such behaviour patterns are described in details. In addition, more complicated behaviour in nonlinear systems is also discussed.

In the domain of system dynamics, the internal structure is classified into three types of variables, namely stocks, flows, and auxiliaries. Nevertheless, a structural component in a more aggregated level, the feedback loop, is on focus. A feedback loop represents a chain of causal-effect relationship, thus owning a strong explanatory power to unearth the underlying cause to the observed behaviour. Subsequently, two important properties associated with the feedback loops, loop polarity and loop strength, are discussed in detail. General principles which formulate the relationship between behaviour and loop polarity are introduced.

Finally, the analytical and computational methodologies for structural dominance analysis are reviewed. The analytic approaches outlined are PPM, EEA and the eigenvector analysis. A detailed discussion of EEA is provided with special attention to its loop selection method, the SILS algorithm. Three simulation-based approaches, the behavioural method, and statistical screening are subsequently introduced. Importantly, the behavioural method is described in detail. For all these methods, their strengths and weaknesses are discussed within each individual section. In summary, Table 2.1 outlines the features of all these methods mentioned and pinpoints the research opportunities.
CHAPTER 2. BACKGROUND AND RELATED RESEARCH

<table>
<thead>
<tr>
<th>Approaches</th>
<th>Features</th>
<th>Weaknesses</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPM</td>
<td>Straightforward, and implemented in a software package, Digest</td>
<td>Not suitable for oscillatory models</td>
</tr>
<tr>
<td>EEA</td>
<td>Able to compare the loop’s dominance directly</td>
<td>Computing Eigenvalue sometimes is not easy. Modify a system’s behaviour but cannot fine-tune an individual variable’s behaviour</td>
</tr>
<tr>
<td>Eigenvector Analysis</td>
<td>Able to fine-tune an individual variable’s behaviour</td>
<td>Lack of method to relate system structure (except parameter) to eigenvector analysis. Lack of analytic methods to perform the eigenvector analysis</td>
</tr>
<tr>
<td>Behavioural method</td>
<td>Works in both linear and nonlinear systems</td>
<td>If no unique edge exists, this loop cannot be analyzed. Lack of software to automate this method</td>
</tr>
<tr>
<td>Statistical screening</td>
<td>Easy implementation in Vensim</td>
<td>Limited to parameters optimization, and ignore the structure deficiency.</td>
</tr>
</tbody>
</table>

Table 2.1 – Summary of structural dominance analysis approaches

This dissertation focuses on a number of the research opportunities outlined in Table 2.1. The topics of the SILS algorithm, eigenvector analysis, and the behavioural method are particularly important with respect to the later sections of this dissertation.
Chapter 3

A Search Algorithm to Identify the Maximal Set of Independent Loops

This chapter presents a detailed description of a loop partition algorithm specially designed for eigenvalue elasticity analysis. As we previously discussed in chapter 2, EEA is an important modal control method with a wide application in the structural analysis of system dynamics models. An exceptional advantage of EEA is its accurate calculation of the loop elasticity, which allows direct comparison of various feedback loops in terms of their influence on the selected behaviour pattern. The loops can be ranked by their eigenvalue elasticity. Higher elasticity results in a more significant impact on the behaviour of interest. The loop with highest eigenvalue elasticity\(^1\) is usually referred to as the dominant feedback loop.

There was no systematic method to select the candidate loops so that their elasticity can be successfully evaluated until the publication of the paper from Kampmann (1996). Kampmann proposed the idea of independent loop set (ILS) which enables the analytical computation of the loop elasticity. Subsequently, an implementation of ILS is proposed by Oliva (2004) to facilitate the utilization of ILS. This implementation is carried out by an algorithm termed as the “shortest independent loop set” (SILS). At present, the SILS is adopted by almost all the applications of EEA in the dominant loop analysis of SD models, see (Goncalves, 2009; Güneralp, 2006b; Huang et al., 2010a; Kampmann and Oliva, 2006; Phaff et al., 2006). However, in

\(^1\) When the loop elasticity is real valued, the highest means the greatest absolute value. Otherwise, if the elasticity is complex valued, the mean of highest can vary depending on the context. It may refer to the greatest absolute value, or greatest absolute value of the real part.
certain situations, we find that the SILS does not guarantee an identification of a complete independent loop set, i.e., the number of loops in SILS is less than the maximum number of the independent loops in a given system. This matters as the failure of the full ILS identification will result in generating erroneous loop elasticity.

This piece of work is inspired by the idea of applying EEA to the agent-based model. Because the typical models that EEA applies to are not agent-based. Exploring EEA in different application areas helps to gain insightful perspectives of the method itself and extend its applicability. However, a close investigation on a fully connected agent-based model reveals a potential problem associated with the SILS: the number of loops in the SILS is less than the actual size of the independent loop set.

This chapter presents a series of studies to examine this problem, involving the relationship between the ILS and the loop elasticity, and proposing an algorithm to identify the complete set of independent feedback loops. This chapter is therefore organized as follows: a discussion on the origin of the ILS, which explains how ILS works to solve the loop elasticity. In addition, the reason for the necessity of utilizing the maximal set of independent loops is provided. Subsequently, a description of SILS and a case study of the agent-based goal diffusion model are presented. Furthermore, we propose an algorithm to identify the complete set of independent feedback loops. Finally, a summary of contributions together with a discussion on future research is outlined.

3.1 The Origin of the Independent Loop Set

The independent loop set is specially designed for EEA to solve the loop elasticity problem. The loop elasticity is a short name for the eigenvalue elasticity with respect to the loop gain. Elasticity is a dimensionless ratio, and the symbol “ε” is used to denote such a measurement which evaluates the percentage change in the eigenvalue in response to the percentage change in the loop gain (g), shown in Equation 3.1.

\[ \varepsilon = \left( \frac{d\lambda}{\lambda} \right) / \left( \frac{dg}{g} \right) = \frac{d\lambda}{\lambda} \frac{g}{dg} \]

(3.1)

Recall a review of EEA in Section 2.4.2, we know that the loop elasticity has to be calculated following the route map shown in Figure 2.17, and we show it again below.
3.1.1 Independent Feedback Loop Set

In light of the known result of the compact link sensitivity (i.e., $\partial \lambda / \partial A(p, q)$), its elasticity can be obtained by multiplying their ratio. The pathway elasticity and causal link elasticity can be consequently derived based on the chain rule, see Appendix C. The relationships between these structural elasticity are summarized as follows:

- The **pathway elasticity** equals to its corresponding compact link (where the pathway lies) elasticity multiplied by their gain ratio (pathway gain over the compact link gain).

- The **causal link elasticity** can be obtained by adding up all the pathway elasticities which pass through the causal link.

- The **causal link elasticity** is equal to the sum of the elasticities of the eigenvalue with respect to the gains of all the loops that pass through this link.
Owing to the last item, we can formulate the relationship between the causal link elasticity and the loop elasticity as:

\[
\begin{bmatrix}
\varepsilon_{e_1} \\
\varepsilon_{e_2} \\
\vdots \\
\varepsilon_{e_N}
\end{bmatrix} = C_r
\begin{bmatrix}
\varepsilon_{c_1} \\
\varepsilon_{c_2} \\
\vdots \\
\varepsilon_{c_L}
\end{bmatrix}
\]  

(3.2)

where \( L \) refers to the number of loops, \( N \), the number of edges, and \( C_r \) is an \( N \times L \) binary matrix. \( C_r \) stands for the directed cycle matrix (DCM) (Kampmann, 1996). Each entry, say \( C_r(i, j) \), indicates if the causal link \( e_i \) belongs to the loop \( c_j \). Given the causal link elasticity vector, the loop elasticity vector can be solved only when

\[
\text{rank}(C_r) = L \leq N
\]  

(3.3)

In general, the number of possible feedback loops of a given system is much larger than the number of its edges, thus \( \text{rank}(C_r) \leq N < L \). In this scenario, there will be no unique solution to the loop elasticity (number of unknowns are greater than the number of equations). The challenge is to select appropriate feedback loops to satisfy the constraint in Equation 3.3. It actually consists of two conditions that the candidate feedback loop set should meet: (1) Loop size has to be less or equal to the edge size, \( L \leq N \); (2) The loops have to be linearly independent of each other, or equivalently, \( \text{rank}(C_r) = L \).

The independent loop set (ILS) was proposed by Kampmann (Kampmann, 1996) to be “a maximum set of loops whose incidence vectors are linearly independent”.

The incidence vector of a loop \( S \) can be denoted as \( V_s = (v_1, \ldots, v_n) \), where \( v_i = 1 \) if the edge \( e_i \in S \) and 0 otherwise. From the above statement, we know that there are two requirements for the ILS: the incidence vectors of the loop member in ILS are linearly independent, and no such a loop can be found in the system whose incidence vector is linearly independent of the current collection of all incidence vectors in ILS. In addition, an important theorem regarding the ILS is derived to guide the identification of these loops:

“In a strongly connected digraph, the number of maximal independent loops are \( N-n+1 \).” \( N \) refers to the number of edges, and \( n \) denotes the number of vertices (variables).

\(^{2}\) In linear algebra, a family of vectors is linearly independent if none of them can be written as a linear combination of finitely many other vectors in this collection.
A strongly connected digraph termed $G$ is one in which, for any pair of nodes $x, y \in G$, there is both a directed path from $x$ to $y$ and a directed path from $y$ to $x$. The proof of this theorem can be found in Kampmann (1996). It is not difficult to see the ILS satisfies the constraint in Equation 3.3, and it therefore assures a unique solution to the loop elasticity.

### 3.1.2 Further Investigation: Why ILS needs to be maximal?

Although the ILS allows Equation 3.2 to yield a unique solution to the loop elasticity, it leads us to think a step further, should the ILS be maximal? The constraint in Equation 3.3 demands an independent loop set, but does not require a complete set of independent loops. In other words, any independent loop set rather than the ILS is sufficient to solve loop elasticity in Eq. 3.2. To seek an answer to this question, we have to approach the loop elasticity computation from another perspective through Mason’s rule (Reinschke, 1988). Within this application, what we need to know about Mason’s rule is that, the computation of the eigenvalues requires the information of all feedback loops.

We assume that a dynamical system whose corresponding digraph has only one strongly connected component $G$. Let $J$ be the $n \times n$ gain matrix (termed as Jacobian matrix as well) for the given system, and $P(\lambda)$ be the characteristic polynomial for the eigenvalues $\lambda$.

$$
P(\lambda) = |\lambda I - J| = \lambda^n + p_1\lambda^{n-1} + p_2\lambda^{n-2} + \cdots + p_{n-1}\lambda + p_n 
$$

(3.4)

According to Mason’s rule, any coefficient $p_i$ can be represented by a proper combination of loop gains. In order to express the coefficient in terms of the loops, we hence need the information about all the feedback loops that a given system can have (interested readers can find an example to illustrate this theorem in Section 3 (Kampmann, 1996)). We then modify Equation 3.4 where all the coefficients are replaced by the loop gains:

$$
P(\lambda, g_{i_1}, \ldots, g_{i_m}) = P(\lambda; g_i) = 0 
$$

(3.5)

where $m$ refers to the number of all possible loops the system can have. To calculate the loop elasticity, we first take derivative of Equation 3.5 with respective of a loop gain $g_{i_1}$, which
yields,
\[ \frac{dP(\lambda, g_{\ell_i})}{dg_{\ell_i}} = \frac{\partial P(\lambda; g_{\ell_i})}{\partial \lambda} \frac{\partial \lambda}{\partial g_{\ell_i}} + \frac{\partial P(\lambda; g_{\ell_i})}{\partial g_{\ell_i}} + \ldots + \frac{\partial P(\lambda; g_{\ell_i})}{\partial g_{\ell_i}} + \ldots = 0 \] (3.6)

The above equation can be simplified as Equation 3.7 given the assumption that all the feedback loops are independent. This assumption results in \( \frac{\partial g_{\ell_k}}{\partial g_{\ell_i}} = 0 \) when \( i \neq k, 0 < k \leq m \).

We emphasize again that the loop set \( g_{\ell} \) has to be a complete loop set with all possible loops in a system. This is the underlying reason why the maximal ILS is required instead of a smaller ILS. Since a distinction between the ILS and the smaller size ILS is that only the ILS can represent all possible feedback loops, i.e., any other feedback loops that exist in the system can be expressed as a linear combination of the loops in the ILS. In other words, all feedback loops in the system can be expressed in terms of the loops in ILS. Therefore, the ILS is the smallest loop set that is able to sufficiently represent \( g_{\ell} \). Only when the eigenvalue can be formulated correctly, can the loop elasticity be calculated. The above analysis shows the necessity of utilizing the ILS to compute the eigenvalue and the loop elasticity.

In order to provide the reader a more comprehensive perspective of the loop elasticity calculation, we would like to briefly show the relationship between the loop elasticity and the causal link (i.e., a link connects any two vertices in \( G \)) elasticity. Assume \( g_{e_j} \) denotes the gain of a causal link \( e_j \). Mojtahedzadeh defines a loop gain to be a product of all its constituent causal link gains (Mojtahedzadeh, 1997) (see Section 2.3 for references). The causal link elasticity of \( e_j \) can be computed as follows:

\[ \frac{\partial \lambda}{\partial g_{e_j}} = \frac{\partial P(\lambda; g_{\ell_i})}{\partial \lambda} \frac{\partial \lambda}{\partial g_{e_j}} + \frac{\partial P(\lambda; g_{\ell_i})}{\partial g_{e_j}} = 0 \]

\[ \frac{d\lambda}{dg_{e_j}} = - \frac{\partial P(\lambda; g_{e_j})}{\partial g_{e_j}} \left( \frac{\partial P(\lambda; g_{e_j})}{\partial \lambda} \right)^{-1} \] (3.9)

\[ \frac{\partial P(\lambda; g_{\ell_i})}{\partial g_{e_j}} = \sum_{e_j \in \ell_i} \frac{\partial P(\lambda; g_{\ell_i})}{\partial g_{\ell_i}} g_{e_j} \] (3.10)
Equation 3.10 is obtained by using the chain rule, i.e., the characteristic equation is a function of loops while the loop is a function of the causal links. With the edge sensitivity derived in Equation 3.9, its elasticity is obtained:

$$\varepsilon_{ej} = \frac{d\lambda}{dg_{e_j}} = \lambda \left\{ \sum_{e_j \in \ell_i} \frac{\partial P(\lambda; g_{e_i})}{\partial g_{e_j}} \frac{g_{e_i}}{g_{e_j}} \right\} \left( \frac{\partial P(\lambda; g_{e_i})}{\partial \lambda} \right)^{-1} \frac{g_{e_i}}{\lambda}$$

$$= \sum_{e_j \in \ell_i} \left\{ -\frac{\partial P(\lambda; g_{e_i})}{\partial g_{e_i}} \left( \frac{\partial P(\lambda; g_{e_i})}{\partial \lambda} \right)^{-1} \frac{g_{e_i}}{\lambda} \right\}$$

$$= \sum_{e_j \in \ell_i} \varepsilon_{e_i}$$

(3.11)

Finally, Equation 3.8 - 3.11 have proven that the causal link elasticity is the sum of the elasticities of all loops that encompass that link. This solves the under-determined problem pointed out in Figure 3.1.

The above discussion has clarified the fundamental reason of utilizing the ILS in the loop elasticity computation in EEA. As ILS is crucial to generate the loop elasticity, another algorithm which is a simpler and more granular implementation of ILS is widely used and known as shortest independent loop set (SILS). The following sections are to examine this algorithm closely.

### 3.2 Shortest Independent Loop Set

The shortest independent loop set algorithm is a simple and granular implementation of ILS proposed by Oliva (2004). The essence of SILS is based on the idea of working with the geodetic cycles. A geodetic cycle is a shortest loop in which the two vertices under consideration are involved. Geodetic cycles are a collection of such shortest loops involving any two vertices in a given system. They are a subset of all the possible feedback loops. In general, the number of possible loops in a system is much greater than the size of the geodetic cycles. In the process of constructing ILS, the SILS is constrained to introduce new loops from the geodetic cycles only, i.e., the member loops of SILS are all geodetic cycles. Furthermore, the selected loop to be introduced into ILS always has the fewest new edges (compared with other candidate loops in the geodetic cycles). This is consistent with Kampmann’s construction rule: “accept a loop in the ILS if it contains at least one edge not included in the previously accepted loops”. The SILS is a special implementation of ILS and not unique (due to the fact that the geodetic
cycles are not unique when there are two or more “shortest” paths of equal length between two vertices).

As the geodetic cycles are crucial to SILS, its computation is described. For a given system represented by a diagraph \( \mathcal{G} \), the adjacency matrix \( A \) is first to be generated based on its structure (or equations). The adjacency matrix is a square matrix with the size of vertices in \( \mathcal{G} \). Each row (and column) represents a vertex, and the entries are restricted to zero and one, where \( A_{ij} = 1 \) IFF there is a link from vertex \( i \) to vertex \( j \). The ones in row represent the successors while the ones in column represent the predecessors.

A subsequent step is to derive the distance matrix. Each cell of a distance matrix \( D \) shows the length of the shortest path (a sequence of non-repeating edges and vertices) between two vertices (Warfield, 1989). It can be obtained by virtue of a well known result from graph theory which states: “an adjacency matrix with its main diagonal filled with ones — to the \( i \)th power, yields the matrix of a digraph with the relationship reachable with \( i \) steps”. The above statement is formulated in (Oliva, 2004) as:

\[
B = A + I
\]
\[
D = A + \sum_{i=2}^{\lfloor |A| \rfloor - 1} i(B^i - B^{i-1})
\]

where \( |A| \) is the dimension of \( A \), \( I \) is an identity matrix of the same dimension as \( A \), and all operators are in ordinary matrix algebra, with the exception of the power of \( B \), which is a Boolean product. Every time after one iteration of the power operation, any newly found reachable paths between two vertices will be marked with 1 in corresponding cells in the resulted matrix of \( B^i - B^{i-1} \). After \( |A| - 1 \) times iterations, all reachable paths are searched and the shortest paths between any two vertices are identified. Nevertheless, a sequential series of edges involved in each shortest path have not identified, which remains to be solved in Section 3.3. Finally, the geodetic cycles matrix \( L \) is calculated by adding up the distance matrix and its transpose, i.e., \( D + D^T \). The geodetic cycles matrix is symmetric, and either its upper or lower triangular block contains the information about the fewest edges needed to form a loop involving any two vertices.

The key idea of the SILS construction rule is to introduce a new cycle that makes the minimal contribution of new edges into the ILS, and the algorithm is constrained to work only with geodetic cycles. We assume the resulted independent loop set is termed as SILS. The procedure
to implement the SILS algorithm can be briefly summarized as follows:

1. Set up the geodetic cycle matrix for a given system.

2. Identify the shortest feedback loop among the geodetic cycles, and transfer it to the SILS. This is the first loop introduced into the SILS.

3. Among the remaining geodetic cycles, select the shortest feedback loop that makes the smallest addition of new edges to the identified SILS. Add it to the loop set.

4. Repeat step 3 until all the geodetic cycles are visited.

The pseudo-code of the above description on constructing the SILS from the geodetic cycles is outlined in Algorithm 1 in Section 2.4.2.

3.2.1 Does SILS Capture the Maximal Set of Independent Loops?

The SILS algorithm has become the most widely adopted approach in feedback loop analysis. For example, the behavioural method (Phaff, 2008), and the eigenvalue elasticity analysis (Güneralp, 2006b; Kampmann and Oliva, 2006; Saleh et al., 2005). However, a question to be addressed is can the SILS algorithm capture the ILS, because it constrains the candidate feedback loops to be the geodetic cycles only. Would this constraint affect its capability to identify the ILS?

The suspicion remains until we attempt to apply EEA to an individual-based version (Duggan, 2008; Feola et al., 2011; Osgood, 2009) based on the floating goals model (Sterman, 2000). The original model is adapted to allow individual agents to pursue different goals by modifying their own target goal in equation $F_i$. Each agent has one state variable $GoalAgent_i$, representing its current goal level. $GoalAgent_i$ is adjusted through interactions with its neighbours via $TargetGoal_i$, and the adjustment time is $AT_i$. This three-agent model with a fully connected network is termed goal diffusion model and shown in Figure 3.2. For individual agent, its equations are formulated in Equation 3.13.
CHAPTER 3. A SEARCH ALGORITHM TO IDENTIFY THE MAXIMAL SET OF INDEPENDENT LOOPS

GoalAgent_i = INTEGRAL(ChangeInGoal_i, GoalAgent_i^0)

ChangeInGoal_i = (TargetGoal_i - GoalAgent_i) / AT_i

TargetGoal_i = F_i(GoalAgent_1, GoalAgent_2, GoalAgent_3)  \hspace{1cm} (3.13)

In addition to the stock-flow diagram, we present a graph representation of the model in Figure 3.3 (the adjustment times are not under consideration as they are not part of the strongly connected digraph). Applying Kampmann’s theorem 2.1, the maximal independent loop number of the goal diffusion model is: 18 - 9 + 1 = 10. However, the SILS algorithm can only identify 9 loops\(^3\). The independent feedback loop sets identified by the SILS algorithm and the ILS are shown in Table 5.3. Particularly, the identification sequence of each loop in the ILS algorithm is marked with circled indices. The one that is not identified by the SILS algorithm is highlighted in yellow dashed lines in both Figure 3.2 and 3.3. The missing loop is the longest.

\(^3\) However, we are still curious about the loop elasticities produced from these two different independent loop sets. A comparison table showing their discrepancies is provided in Appendix D.
Figure 3.3 – The simplified graph of the goal diffusion model. Blue squares: *GoalAgents*, red circles: *TargetGoals*, and black parallelograms: *ChangeInGoals*.

<table>
<thead>
<tr>
<th>Method</th>
<th>Loop name</th>
<th>Variables involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILS</td>
<td><strong>Agent1</strong> floating goal spiral state adjustment</td>
<td>1, 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1, 3, 2</td>
</tr>
<tr>
<td></td>
<td><strong>Agent2</strong> floating goal spiral state adjustment</td>
<td>4, 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4, 6, 5</td>
</tr>
<tr>
<td></td>
<td><strong>Agent3</strong> floating goal spiral state adjustment</td>
<td>7, 8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7, 9, 8</td>
</tr>
<tr>
<td></td>
<td>interaction of A1 &amp; A2</td>
<td>1, 6, 5, 4, 3, 2</td>
</tr>
<tr>
<td></td>
<td>interaction of A1 &amp; A3</td>
<td>1, 9, 8, 7, 3, 2</td>
</tr>
<tr>
<td></td>
<td>interaction of A2 &amp; A3</td>
<td>4, 9, 8, 7, 6, 5</td>
</tr>
<tr>
<td></td>
<td>interaction of A1, A2 &amp; A3</td>
<td>1, 9, 8, 7, 6, 5, 4, 3, 2</td>
</tr>
</tbody>
</table>

Table 3.1 – Two independent feedback loop sets in the goal diffusion model.
loop that encompasses three agents. Close scrutiny shows that the missing loop is even not a geodetic cycle as any two variables involved in this loop can be encircled by a geodetic cycle. In this model, any vertices in different agents can be linked by a shortest path encompassing the corresponding two agents without involving a third agent. This shows that constraining the candidate loop pool to be the geodetic cycles is a drawback that can impact the number of independent loops which can be identified. This is the reason why the SILS will not achieve the full coverage of ILS in some cases.

Further investigation provides a deeper explanation to this drawback of the SILS algorithm: the loops introduced to ILS by Kampmann’s construction rule are not always the geodetic cycles. Recall the first statement in point 2 of this rule (p.8, (Kampmann, 1996)) dealing with adding a new loop, it is expressed as “choose a shortest path outside that comes back to the current ILS”. A geodetic cycle is a combination of two shortest paths. Nevertheless, this construction rule needs only the path (part of the new loop) outside the ILS to be the shortest. It does not require the path inside the ILS to be the shortest. Therefore, a non-geodetic cycle may be introduced into the ILS. It is worth noting that any two vertices in ILS are connected, but not necessarily by the shortest path. This scenario can be observed from the case example, with first 8 loops introduced in Table 5.3, for loop 9, the edge 7→6 is added. This new edge is part of the geodetic cycle involving vertices 7 and 6 (the geodetic cycle is 7→6→5→4→9→8→7). However, the path inside ILS from 6 to 7 is 6→5→4→3→2→1→9→8→7, and not the shortest. Consequently, a non-geodetic cycle is introduced into ILS. Finally, the last remaining edge 4→9 is introduced to form loop 10.

### 3.3 An Algorithm for Identifying ILS

In this section, we will specify an ILS algorithm by making use of Kampmann’s construction rule, i.e., “accept a loop in the ILS if it contains at least one edge not included in the previously accepted loops”. There are several implementation issues which have not been specified in Kampmann’s work (Kampmann, 1996). For example, there is no clear starting point of the ILS algorithm, and no specific procedure to introduce a new independent feedback loop into the ILS. In our algorithm, we select a shortest feedback loop as the first loop, which can be obtained by the geodetic cycles matrix. The geodetic cycles matrix is computed as follows.
CHAPTER 3. A SEARCH ALGORITHM TO IDENTIFY THE MAXIMAL SET OF INDEPENDENT LOOPS

(variables in Figure 3.2 are indexed in Figure 3.3, each row/column in a matrix corresponds to an indexed variable):

1. Adjacency matrix \( A \), is a computational representation of a digraph in binary form, and can be extracted from the model structure in Figure 3.3.

\[
A = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
7 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
8 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
9 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\]

2. Distance matrix \( D \) is then obtained by utilizing Equation 3.12 together with the adjacency matrix \( A \) (this can be computed in MATLAB):

\[
D = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 0 & 1 & 1 & 3 & 2 & 1 & 3 & 2 \\
2 & 1 & 0 & 3 & 4 & 3 & 2 & 4 & 3 \\
3 & 2 & 1 & 0 & 5 & 4 & 3 & 5 & 4 \\
4 & 3 & 2 & 1 & 0 & 1 & 1 & 3 & 2 \\
5 & 4 & 3 & 2 & 1 & 0 & 2 & 4 & 3 \\
6 & 5 & 4 & 3 & 2 & 1 & 0 & 5 & 4 \\
7 & 3 & 2 & 1 & 3 & 2 & 1 & 0 & 1 \\
8 & 4 & 3 & 2 & 4 & 3 & 2 & 1 & 0 \\
9 & 5 & 4 & 3 & 5 & 4 & 3 & 2 & 1
\end{pmatrix}
\]

3. The geodetic cycles matrix is calculated by adding the upper block triangular component of the distance matrix to its lower triangular component (the lower triangular component...
CHAPTER 3. A SEARCH ALGORITHM TO IDENTIFY THE MAXIMAL SET OF INDEPENDENT LOOPS

of the geodetic matrix is left blank as it is symmetric):

\[ L = D + D^T = \]

\[
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
0 & 2 & 3 & 6 & 6 & 6 & 6 & 6 & 6 \\
0 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
0 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
0 & 2 & 3 & 6 & 6 & 6 & 0 & 2 & 3 \\
0 & 3 & 6 & 6 & 6 & 0 & 3 & 6 & 6 \\
0 & 6 & 6 & 6 & 6 & 0 & 6 & 6 & 6 \\
0 & 2 & 3 & 6 & 6 & 0 & 2 & 3 & 6 \\
0 & 3 & 6 & 6 & 6 & 0 & 3 & 6 & 6 \\
0 & 6 & 6 & 6 & 6 & 0 & 6 & 6 & 6 \\
\end{pmatrix}
\]

Algorithm 2 Vector pathTrack (Matrix \( D \), int u, int v)

<table>
<thead>
<tr>
<th>Vector vec, pre, suc</th>
<th>{initialize the vectors}</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec ← u</td>
<td>{add the first vertex of the path}</td>
</tr>
<tr>
<td>len ← ( D(u,v) )</td>
<td>{get the length of the shortest path}</td>
</tr>
<tr>
<td>for ( i = 0 ) to len-1 ( \text{do} )</td>
<td></td>
</tr>
<tr>
<td>pre ← findNPre ( D, v, \text{length-}i )</td>
<td>{find all predecessors of v with length-}i steps away}</td>
</tr>
<tr>
<td>suc ← findNSuc ( D, u, 1 )</td>
<td>{find all immediate successors of u}</td>
</tr>
<tr>
<td>for each vertex ( j \in \text{suc} )</td>
<td></td>
</tr>
<tr>
<td>if ( j \in \text{pre} ) then</td>
<td></td>
</tr>
<tr>
<td>vec ← j</td>
<td>{add the vertex to the vector vec}</td>
</tr>
<tr>
<td>break</td>
<td></td>
</tr>
<tr>
<td>end if</td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
<tr>
<td>end for</td>
<td></td>
</tr>
<tr>
<td>return vec</td>
<td></td>
</tr>
</tbody>
</table>

If there exists more than one loop with the equal shortest length, we will choose the one which comes first in the geodetic cycles matrix. The loop is divided into two paths, e.g., \( u \to v \) and \( v \to u \) (\( D_{uv} = \text{shortest length} \)), thus, a loop track can be carried out by tracking the elements in individual path. Algorithm 2 shows the pseudo-code for the path track by taking advantage of the distance matrix.

Once the first feedback loop is selected, a number of key issues are clarified before proceeding, namely to:

1. Distinguish the vertices which have been added into the ILS (or call them visited vertices) from those which have not been visited.
2. Maintain two adjacency matrices: one contains the edges in current ILS (or named as visited edges), and the other is composed of the edges outside of ILS.

3. Compute the distance matrix $D$ with visited edges only, and update it every time after a new loop is introduced into the ILS. This is one distinction from the SILS where the distance matrix is calculated with all the edges.

These updates should be performed every time after one feedback loop is introduced into ILS.

Algorithm 3 Stack addLoop (int cur, Vector vertices, Matrix $A$)

Stack $S$ = new Stack()
Vector path = new Vector()
repeat
  if $cur \in$ vertices then
    $S$.push($cur$) {if the $cur$ vertex is a visited vertex, push it in stack}
    return $S$
  else if $cur \neq -1$ then
    $S$.push($cur$) {if the succeeding edge exists, push the $cur$ vertex in stack}
    path ← $cur$ {add it to the search path}
    $cur$ ← findSuc($A$, $cur$, path)
  else
    $S$.pop() {if no succeeding edge exists, pop out the vertex}
    $q$ ← $S$.peek() {obtain the new top vertex}
    $cur$ ← findSuc($A$, $cur$, path)
  end if
until $\neg S$.isEmpty()
return $S$

We now proceed to the core of Kampmann’s construction rule: start from a visited vertex and track a shortest path back to the existing loop set. Our algorithm takes a slightly different strategy on how to introduce new loops to the ILS. We start from a vertex, then proceed until we identify that one vertex is visited (Figure 3.4). As a result, this path may not be the shortest. However, it still meets the constraint that “every new loop has at least one new edge” (Kampmann, 1996). This procedure is implemented by “depth-first search” (Cormen et al., 2001) and a stack data structure is used to record all the vertices in the search path. The search terminates whenever it encounters a vertex that is visited (i.e., back in the ILS). One scenario needs special treatment is when a vertex outside ILS have been visited twice in the current search path (such a search path is marked in dashed/dotted lines in Figure 3.4(d)). This edge has to
CHAPTER 3. A SEARCH ALGORITHM TO IDENTIFY THE MAXIMAL SET OF INDEPENDENT LOOPS

(a) A loop in ILS.  
(b) Two outside vertices connected to a vertex in ILS.

(c) A new search path with dotted blue line.  
(d) Identify a path that forms a loop outside ILS.

(e) Revoke the last edge in the search path.  
(f) Revoke the edges in the search path one by one if there is no alternative succeeding edges available.

(g) Find an alternative succeeding edge.  
(h) Identify a path back to a vertex in ILS.

Figure 3.4 – Illustration of the searching process for a new independent feedback loop. Dotted blue line: current valid search path; black dash line: visited but discarded (still considered as unvisited) edges.
be discard (see Figure 3.4(e)), because a loop is formed in the external path. Subsequently, another unvisited succeeding edge is attempted if there exists any alternative succeeding edges starting from vertex $r$. In some cases, all its succeeding edges may go back to the vertices in the search path or there are no other succeeding edges, then this vertex has to be discarded by popping the top node out from the stack. A search starting from the new top node in the stack continues (see Figure 3.4(f) and 3.4(g)). Figure 3.4 exemplifies the above description: initially we follow the edge to $p$ to explore a path back to ILS, however, we have to revoke the edge which forms a loop in the search path. After revoking this edge, we cannot find an alternative succeeding edge after $r$, the revocation continues until we return to the starting point $x$. There is another succeeding edge ending with $q$ available. Applying DFS, we finally find a path back to a vertex $y$ in ILS. An implementation of the above procedure is presented in Algorithm 3 which takes the starting vertex, visited vertices the adjacency matrix (with unvisited edges only) as inputs.

So far, we have shown how to track a path back to ILS, the remaining task is to close this new feedback loop by seeking an internal path from $y$ to $x$ in ILS. This procedure is to be described as follows. As the stack’s top and bottom positions hold the beginning and end vertices of this path, we can make use of Algorithm 2 to trace the rest part of the loop. We intend to track a shortest path that connects these two vertices in ILS. However this time, the distance matrix $D$ is computed from an adjacency matrix which is constructed by visited edges only. Therefore, every time after a new feedback loop is introduced into ILS, the adjacency matrix, distance matrix, vertices set and edge set are updated. The shortest path between two vertices therefore may change every time after one new loop is introduced. This is a significant difference with the SILS algorithm where the geodetic cycles are fixed. Moreover, this gives rise to generating more feedback loops than the geodetic cycles, and ensures the algorithm identifies the ILS.

Finally, a formal pseudo-code for the main function of identifying the ILS is provided in Algorithm 4. It takes four parameters as the input. $A$ and $D$ are the system adjacency matrix and distance matrix. Vertices is a vector containing all the vertices while Edges is a matrix with all edges in the system.

A subsequent experiment is carried out to examine the missing loops in the goal diffusion model when increasing the agent population. Table 3.2 shows that the gap of the loop size between the ILS and the SILS grows when the agent population rises in the context of a fully
Algorithm 4 Main($A$, $D$, Vertices, Edges)

Vector $vis_v$, $non\_vis_v$ ← Vertices \{Initialize vectors to store visited and non-visited vertices\}
Vector $vis_e$, $non\_vis_e$ ← Edges \{Initialize vectors to store visited and non-visited edges\}
Vector $g\_cycles$
Vector $ils$
$g\_cycles$ ← calcGeodeticM($D$) \{Obtain geodetic cycles by utilizing pathTrack func\}
$ils$ ← $g\_cycles$ \{Introduce a shortest loop to $ils$\}
$vis_v$ ← updateV($ils$, Vertices) \{Update the visited vertex vector\}
$non\_vis_v$ ← Vertices$–$ $vis_v$ \{Obtain the non-visited vertex vector\}
$vis_e$ ← updateE($ils$, Edges) \{Update the visited edge vector\}
$non\_vis_e$ ← Edges$–$ $vis_e$ \{Obtain the non-visited edge vector\}
$A$ ← updateAdj($non\_vis_e$) \{Update adjacency matrix with edges outside the ILS\}
$D$ ← updateDis($vis_e$) \{Update distance matrix with edges in the ILS\}

while $non\_vis_e$ != null do
    int $cur$ ← findCur($non\_vis_v$, $A$) \{Pick one vertex in $non\_vis_v$ and it has an edge with a vertex in $vis_v$\}
    int $v$ ← findNPre($A$, $cur$, 1) \{Record the vertex in $vis_v$\}
    $S$ ← addLoop($cur$, $vis_v$, $A$) \{Identify a path going back to current ILS\}
    $u$ ← $S$ \{Get the vertex that is in both ILS and $S$\}
    $S$.add(pathTrack($D$, $u$, $v$)) \{Identify a path inside current ILS to form a new loop\}
    $ils$.add($S$) \{Add the new loop\}
    $vis_v$ ← updateV($S$, Vertices) \{Identify a path going back to current ILS\}
    $non\_vis_v$ ← Vertices$–$ $vis_v$
    $vis_e$ ← updateE($S$, Edges) \{Identify a path inside current ILS to form a new loop\}
    $non\_vis_e$ ← Edges$–$ $vis_e$
    $A$ ← updateAdj($non\_vis_e$)
    $D$ ← updateDis($vis_e$)
    $S$.clear() \{Clear the vector $S$\}
end while
CHAPTER 3. A SEARCH ALGORITHM TO IDENTIFY THE MAXIMAL SET OF INDEPENDENT LOOPS

connected network.

<table>
<thead>
<tr>
<th>Agent population</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>(i(\geq 2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of edges (N)</td>
<td>10</td>
<td>18</td>
<td>28</td>
<td>40</td>
<td>54</td>
<td>70</td>
<td>88</td>
<td>108</td>
<td>130</td>
<td>((i+3)i)</td>
</tr>
<tr>
<td>No. of variables (n)</td>
<td>6</td>
<td>9</td>
<td>12</td>
<td>15</td>
<td>18</td>
<td>21</td>
<td>24</td>
<td>27</td>
<td>30</td>
<td>(3i)</td>
</tr>
<tr>
<td>N-n+1</td>
<td>5</td>
<td>10</td>
<td>17</td>
<td>26</td>
<td>37</td>
<td>50</td>
<td>65</td>
<td>82</td>
<td>101</td>
<td>(i^2 + 1)</td>
</tr>
<tr>
<td>No. of loops (ILS)</td>
<td>5</td>
<td>10</td>
<td>17</td>
<td>26</td>
<td>37</td>
<td>50</td>
<td>65</td>
<td>82</td>
<td>101</td>
<td>(i^2 + 1)</td>
</tr>
<tr>
<td>No. of loops (SILS)</td>
<td>5</td>
<td>9</td>
<td>14</td>
<td>20</td>
<td>27</td>
<td>35</td>
<td>44</td>
<td>54</td>
<td>65</td>
<td>((i+3)i/2)</td>
</tr>
<tr>
<td>No. of missing loops</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>10</td>
<td>15</td>
<td>21</td>
<td>28</td>
<td>36</td>
<td>(\sum_{j=2}^{i}(j-2))</td>
</tr>
</tbody>
</table>

Table 3.2 – Agent population vs. independent loop sets in a fully connected network

3.4 Summary and Discussion

This chapter presents a number of studies concerning the application of SILS in EEA, and makes a series of contributions from the following perspectives. Initially, a drawback of the SILS is discovered by studying a fully-connected goal diffusion model, i.e., the SILS is not guaranteed to identify the maximal independent loops. Close scrutiny reveals the underlying cause is due to the fact that the SILS constrains the candidate feedback loops to be only geodetic cycles, which rules out other possible feedback loops, and results in a smaller sized ILS. As the specific conditions of such models that make the SILS fail are unknown, we have to be careful when using it in EEA. In addition, the necessity of utilizing the complete set of independent loops to solve the loop elasticity in EEA is investigated thoroughly. Finally, we propose an algorithm that guarantee to identify the ILS.

This study also brings up a number of questions for further research. 1. The specific model characteristics that make the SILS algorithm fail are not know yet. It is worth the effort to identify these characteristics. 2. The ILS is not unique and the resulted loop set is contingent on many factors, for example, the model structures (whether there exists equal size of shortest paths between two variables or not), and the indexing sequence of the variables (this affects the searching path in looking for a new loop). This implies there would exist other possible loop partitions based on the same model. One consideration is whether the identified dominant feedback loops are consistent with the analysis from other approaches. A relevant study was conducted in (Oliva and Mojtahedzadeh, 2004) showing that, for various models,
the SILS contains the dominant loops identified by PPM. Therefore, similar research has to be performed to test if our ILS algorithm captures the dominant loops. 3. We strive to extend the applications of EEA to network-based individual models (Mungovan et al., 2011; Rahman-dad and Sterman, 2008). This includes extending the existing techniques and exploring new approaches to formally analyze feedback loop structures across disaggregated equation-based models.
Chapter 4

An Analytical Eigenvector Approach for Analysing Linear Systems

In this chapter, we present a number of studies concerning the analytic eigenvector analysis to identify dominant structures that give rise to the state behaviour of interest in a linear system. Elements of the work outlined in this chapter has been disseminated in (Huang et al., 2010a). As the EEA is always criticized (Ford, 1999) by its identified dominant feedback structure at the level of the model which fails to relate it to any selected variable of interest, the focus on the eigenvector analysis starts to emerge. In general, the eigenvector analysis attempts to improve the EEA by considering how much a behaviour mode is expressed in a particular state variable.

This chapter proposes an analytical methodology to compute the eigenvector sensitivity, and is organized as follows: initially, a survey on the related research is presented. Consequently, a new representation of the state variable’s trajectory in the linear system is introduced. In light of the new expression, a generic solution to the comprehensive state behaviour sensitivity is developed. As this solution contains the eigenvector sensitivity components, the subsequent section is dedicated to solving the eigenvector sensitivity analytically. Then a case study of the labour-inventory model is presented to examine the validity and efficiency of the analytical eigenvector analysis method. In the end, a summary of this chapter is outlined and a discussion on the limitations of this piece of work is provided.
4.1 Background Research

As we have previously discussed, eigenvalues (denoted by $\lambda$) characterize various behaviour modes (i.e., divergent, convergent, or oscillatory) which are expressed in the state variable’s behaviour in the expressions below:

$$x_i = \alpha_i^0 r_{i1} + \ldots + \alpha_j^0 r_{ji} + \ldots + \alpha_n^0 r_{ni}$$

(4.1)

$$= w_{i1} e^{\lambda_{i1} t} + \ldots + w_{ji} e^{\lambda_{ji} t} + \ldots + w_{ni} e^{\lambda_{ni} t}$$

(4.2)

We can see these modes are not equally represented due to different weight they have $\alpha_j^0 r_{ji}$ (i.e., $w_{ji}$). This shows that not only the eigenvalue, but also the weight (eigenvector is part of the weight) will influence the behaviour. The weight represents how significant a certain behaviour mode is on the trajectory. The result of eigenvector analysis therefore can be used as an alternative or supplementary way to adjust the behaviour, and further be used to identify the structural cause for the observed behaviour. Such analysis enable us relate the results to a particular behaviour mode in a state trajectory, which rectifies the critics about EEA.

Although the eigenvector analysis is still in its early stage, a number of researchers have done great work in its development (Goncalves, 2006, 2009; Saleh et al., 2006, 2010). Goncalves (2006) proposed an analytical method to evaluate the eigenvector sensitivity with respect to the feedback loops, $\partial r_{ji}/\partial g_s$, and this method is illustrated by a linear workforce inventory model. In the example, the eigenvalues and eigenvectors of the system are first represented by the loop gains and constants. Then the eigenvector sensitivity with respect to the loop gain can be obtained by simply differentiating the equation on both sides. This is a similar method as the EEA proposed by Goncalves et al. (2000), where the eigenvector is substituted for the eigenvalue. The difficulty of this method is discussed in Section 2.4.2 that transforming the eigenvalue and eigenvector to the representation of the loop gains is extremely hard, especially when the system is large. Moreover, the eigenvector analysis has one more crucial problem, all the eigenvectors are not fixed, and they can be rescaled by any nonzero number. It, therefore, is impossible to represent the eigenvector by a determined expression of the loop gains (i.e., always varied by the scalar), so is the eigenvector elasticity.

Likewise, Saleh et al. (2006) focus on the behaviour mode related weight, $w_{ji}$ in in Equation 4.2, and attempt to solve the weight sensitivity in response to the change of parameters. They
term this method dynamic decomposition weights (DDW) analysis. As shown in Equation 4.1 and 4.2, DDW concerns not only the right eigenvector, but also the coefficient $\alpha$, and is considered to be more comprehensive analysis. The paper (Saleh et al., 2010) proposes a numerical method to compute the weight elasticity with respect to the parameters, and exemplifies the DDW approach by a labour-inventory model. In this model, each parameter is given small perturbation and the weight elasticity is assessed accordingly. Thus the high leverage points, e.g., the most influential parameters are identified. They argue that focusing on the weights, rather than on the eigenvalues, is a more efficient way to develop policy recommendations and computing the elasticity of the weights to parameters allows for a more efficient and discriminate way to identify policies.

Based on current development of the eigenvector analysis, we attempt to develop a method that analytically computes the weight sensitivity with respect to various structural components, such as links and parameters. This is similar as the DDW, and the difference is it is performed in an analytic way (see Figure 4.1). We believe the analytical weight analysis saves time from the simulations in numerical DDW so that it can fast identify the leverage points.

![Figure 4.1 – Current development of the eigenvector analysis research](image)

### 4.2 Solutions to Linear and Inhomogeneous Systems

This section describes the development of alternative representations of the analytical solutions to both linear and inhomogeneous systems. Initially, a procedure for modifying the
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

conventional solution (expressed in Equation 4.1) to be another new formula is presented. Subsequently, a second order linear system is provided to demonstrate this procedure to yield the solution. Finally, the solution to the inhomogeneous system is also discussed. Although the inhomogeneous system is not strictly a linear system, it shares lots of similarities with the linear system and is more commonly seen in practice.

4.2.1 An Alternative Representation of the Solution to Linear Systems

In contrast to the traditional representation of the state trajectory in a linear system shown in Equation 4.1, we are to introduce an alternative solution from which the eigenvector elasticity will be derived. Let us first consider an n-order linear system and assume it has distinct eigenvalues:

\[ \dot{x} = Ax \]
\[ x(0) = x(t_0) \]

where \( x \) is a state vector, \( A \) is a system gain matrix with each entry \( A(i,j) = \frac{\partial x_j}{\partial x_i} \).

From now on, a vector is default to be a column vector, a row vector has to be expressed as a transpose, with a superscript \( T \) (for real-valued vectors) or \( H \) (for complex-valued vectors, shorthand for Hermitian transpose\(^1\)) at its right top, and all matrices are in capital characters.

In addition to our previous discussion on the linear system and the eigenvalues (see Section 2.2.1), the following notations will be utilized:

1. The eigenvalues are placed at the diagonal to form a matrix \( \Lambda \) with all other zero entries:

\[ \Lambda = \begin{bmatrix}
\lambda_1 & 0 \\
0 & \lambda_2 \\
& \ddots \\
0 & & \lambda_n
\end{bmatrix}. \]

2. All the \( n \) right eigenvectors (in columns) of \( A \) constitute the right eigenvector matrix \( R = [r_1 \ldots r_n] \). Given the eigenvalue and eigenvector definitions, their relationship can be formulated in a matrix format as simple as:

\[ AR = R\Lambda \]

\(^1\) Hermitian transpose is also called conjugate transpose and formally defined by: \((A^H)_{i,j} = \overline{A_{j,i}} \).
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

3. All the \( n \) left eigenvectors \( \ell_1^H, \ldots, \ell_n^H \) make up the left eigenvector matrix \( L^H = \begin{bmatrix} \ell_1^H \\ \vdots \\ \ell_n^H \end{bmatrix} \). The right eigenvector and left eigenvector associated with the same eigenvalue are named as an eigenpair.

There are two general, but important properties that the eigenvectors hold:

1. If \( v \) is an eigenvector, then so is \( cv \) for any nonzero scalar \( c \in \mathbb{C} \).

2. The set of right eigenvector \( R \) and the set of left eigenvector \( L^H \) forms a bi-orthogonal system. This relationship is formulated as follows:

\[
\ell_i^H r_j \begin{cases} 
\neq 0 & : i = j \\
= 0 & : i \neq j 
\end{cases}
\]

In other words, the left eigenvector is orthogonal (i.e., perpendicular, \( \ell_i^H \perp r_j \)) to the right eigenvector as long as they are associated with different eigenvalues, otherwise they are not orthogonal. A simple proof is shown below:

\[
A r_i = \lambda_i r_i \quad ; \quad \ell_j^H A = \lambda_j \ell_j^H \\
\ell_j^H A r_i = \lambda_i \lambda_j \ell_j^H r_i \\
\lambda_i \ell_j^H A r_i = \lambda_i \lambda_j \ell_j^H r_i \\
\lambda_i \ell_j^H r_i = \lambda_j \ell_j^H r_i \\
0 = (\lambda_i - \lambda_j) \ell_j^H r_i \implies \ell_j^H r_i = 0 \text{ IFF } \lambda_i \neq \lambda_j
\]

Moreover, with property 1, it is easy to verify that there exist numerous scalars such that the eigenvector can be normalized to satisfy:

\[
\ell_i^H r_j \begin{cases} 
1 & : i = j \\
0 & : i \neq j 
\end{cases} = \ell_j^H r_i = 0 \quad (4.5)
\]

or equivalently, in a matrix form:

\[
L^H R = I \quad (4.6)
\]

It is worth noting that the normalization is not unique due to the existence of numerous possible scalars. For example, any nonzero scalar \( c \) can be applied to the normalized eigenpair so that
\[(c \ell_1^H) (\frac{1}{2} r_1) = \ell_1^H r_1* = 1,\] where we use * to distinguish the new eigenvector from the old one. In light of such a normalization, post-multiplying Equation 4.4 with \(L^H\) renders:

\[
A = R\Lambda L^H
\]  
(4.7)

We view the matrix \(A\) as a constant for the moment to solve the linear system in Equation 4.3:

\[
x(t) = e^{tA}x(0)
\]  
(4.8)

where \(e^{tA}\) is a matrix exponential. Let us start to decouple this solution by expanding the matrix exponential \(e^M\) with the power series:

\[
e^M = I + M + \frac{(M)^2}{2!} + \frac{(M)^3}{3!} + \ldots + \frac{(M)^n}{n!} + \ldots \quad (n = \infty)
\]

where \(I\) refers to an \(n\)-by-\(n\) identity matrix. In conjunction with Equation 4.6 and 4.7, the above equation can be expressed as follows:

\[
x(t) = e^{tA}x(0) = \left( tI + tA + \frac{(tA)^2}{2!} + \frac{(tA)^3}{3!} + \ldots + \frac{(tA)^n}{n!} + \ldots \right) x(0)
\]

\[
= \left( tRL^H + \frac{t^2(R\Lambda L^H)^2}{2!} + \frac{t^3(R\Lambda L^H)^3}{3!} + \ldots + \frac{t^n(R\Lambda L^H)^n}{n!} + \ldots \right) x(0)
\]

\[
= R \left( tI + t\Lambda + \frac{(t\Lambda)^2}{2!} + \frac{(t\Lambda)^3}{3!} + \ldots + \frac{(t\Lambda)^n}{n!} + \ldots \right) L^H x(0)
\]

\[
= Re^{t\Lambda}L^H x(0)
\]

This compact form has to be fully expanded to render a new solution which will be used in the analysis throughout this chapter:

\[
x(t) = \begin{bmatrix} r_1 & r_2 & \ldots & r_n \end{bmatrix} \begin{bmatrix} e^{t\lambda_1} & 0 & \ell_1^H \\ e^{t\lambda_2} & & \ell_2^H \\ \vdots & & \vdots \\ 0 & & e^{t\lambda_n} \end{bmatrix} \begin{bmatrix} x(0) \\ \ell_1^H x(0) \\ \ell_2^H x(0) \\ \vdots \\ \ell_n^H x(0) \end{bmatrix}
\]

\[
= \sum_{i=1}^{n} e^{t\lambda_i} \begin{bmatrix} \ell_1^H \alpha_i^0 \\ \ell_2^H \alpha_i^0 \\ \vdots \\ \ell_n^H \alpha_i^0 \end{bmatrix} x(0)
\]
It is apparent to see the distinction between the conventional solution and ours is the mode coefficient $\alpha_i^0$ is substituted with $\ell_j^H x(0)$. For a particular state variable $x_i$, its eigensolution is shown as follows:

$$ x_i(t) = e^{\lambda_1 r_{i1} \ell_1^H x(0)} + e^{\lambda_2 r_{i2} \ell_2^H x(0)} + \cdots + e^{\lambda_n r_{in} \ell_n^H x(0)} $$

$$ = \sum_{j=1}^{n} e^{\lambda_j r_{ij} \ell_j^H x(0)} \quad (4.9) $$

### 4.2.2 An Example: a Second Order Linear System

In order to clarify the procedure of generating this newly introduced solution, a second order linear system is presented. The system equations are shown below (Strogatz, 2000):

$$ \dot{x} = x + y \quad (4.10) $$

$$ \dot{y} = 4x - 2y; \quad (x_0, y_0) = (2, -3) $$

The gain matrix can be obtained by definition:

$$ A = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 4 & -2 \end{bmatrix} $$

Therefore, the characteristic equation of this linear system is:

$$ 0 = |A - \lambda I| = \begin{vmatrix} 1 - \lambda & 1 \\ 4 & -2 - \lambda \end{vmatrix} = (1 - \lambda)(-2 - \lambda) - 4 \times 1 = (\lambda + 3)(\lambda - 2) $$

hence the eigenvalues $\lambda_{1,2} = 2, -3$. As for the eigenvectors, they can be computed from the definitions using different eigenvalues.

$$ Ar_1 = \lambda_1 r_1 $$

$$ \ell_1^H A = \lambda_1 \ell_1^H $$

We compute the eigenpair associated with $\lambda_1 = 2$ to clarify the eigenvector normalization.

$$ 0 = (A - \lambda_1 I)r_1 $$

$$ = \begin{bmatrix} -1 & 1 \\ 4 & -4 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \end{bmatrix} $$
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

As two rows in \((A - \lambda_1 I)\) are proportional, to solve this equation is equivalent to solve:

\[-r_{11} + r_{12} = 0,\]

the solution to the right eigenvector then yields:

\[
r_1 = c \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

where \(c\) is a nonzero scalar. The left eigenvector is obtained by following the same procedure.

\[
0 = \ell_1^H (A - \lambda_1 I)
\]

\[
= \begin{pmatrix} \ell_{11}^H & \ell_{12}^H \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 4 & -4 \end{pmatrix}
\]

As two rows in \((A - \lambda_1 I)\) are proportional, we only need to solve \(-\ell_{11}^H + 4\ell_{12}^H = 0\). Therefore, the left eigenvector is computed:

\[
\ell_1^H = \beta \begin{pmatrix} 4 \\ 1 \end{pmatrix}
\]

where \(\beta\) is a nonzero scalar. Subsequently, we have to adjust the eigenpair to make them satisfy the the normalization condition. An easier way to accomplish is to let \(\alpha = \frac{1}{\beta}\), the normalized eigenpair are turned out to be:

\[
r_1 = c \begin{pmatrix} 1 \\ 1 \end{pmatrix}
\]

\[
\ell_1^H = \frac{1}{c} \begin{pmatrix} 4/5 \\ 1/5 \end{pmatrix}
\]

Table 4.1 lists the eigenvalues and normalized eigenvectors generated by MATLAB. Note that every eigenpair shown here is one of the infinite possible values.

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Right eigenvector</th>
<th>Left eigenvector</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda_1 = 2)</td>
<td>(r_1 = [0.7071 \ 0.7071]^H)</td>
<td>(\ell_1^H = [1.1314 \ 0.2828]^T)</td>
</tr>
<tr>
<td>(\lambda_2 = -3)</td>
<td>(r_2 = [-0.2425 \ 0.9701]^H)</td>
<td>(\ell_2^H = [-0.8246 \ 0.8246]^T)</td>
</tr>
</tbody>
</table>

Table 4.1 – The eigenvalues and eigenvectors of the linear system
Constructing the solution by Equation 4.9 and the system initial condition, produces:

\[
\begin{align*}
    x(t) &= e^{\lambda_1 t} r_1^T H x(0) + e^{\lambda_2 t} r_2^T H x(0) \\
    &= e^{2t} \ast 0.7071 \ast (1.1314 \ 0.2828) \begin{pmatrix} 2 \\ -3 \end{pmatrix} + e^{-3t} \ast (-0.2425 \ast (-0.8246 \ 0.8246) \begin{pmatrix} 2 \\ -3 \end{pmatrix} \\
    &= e^{2t} + e^{-3t} \\
    y(t) &= e^{\lambda_1 t} r_1^T H x(0) + e^{\lambda_2 t} r_2^T H x(0) \\
    &= e^{2t} \ast 0.7071 \ast (1.1314 \ 0.2828) \begin{pmatrix} 2 \\ -3 \end{pmatrix} + e^{-3t} \ast 0.9701 \ast (-0.8246 \ 0.8246) \begin{pmatrix} 2 \\ -3 \end{pmatrix} \\
    &= e^{2t} - 4e^{-3t}
\end{align*}
\]

**4.2.3 A Solution to the Inhomogeneous System**

One step further, we are to develop a solution to the inhomogeneous system at present to set the scene for a case study of an inhomogeneous system in later sections. Strictly speaking, the inhomogeneous systems are not linear systems (Medio and Lines, 2001). As we have described earlier, the linear system has to satisfy Equation 2.2 in Section 2.2.1. However, they share many similarities and can be solved in the same way. Similar as Equation 4.3, an inhomogeneous system\(^2\) is formulated to be:

\[
\dot{x}(t) = Ax(t) + b \tag{4.11}
\]

denote : \(x(0) = x(t_0)\)

where \(b\) is a constant vector, and \(A\) is an invertible matrix. The above equation can be viewed as a homogeneous linear system if we change the right-hand side to be \(A(x(t) + A^{-1}b)\). It thus can be solved as follows:

\[
\begin{align*}
    \frac{dx(t)}{dt} &= A(x(t) + A^{-1}b) \\
    \frac{dx^*(t)}{dt} &= A \dot{x}^*(t) \\
    \dot{x}^*(t) &= e^{tA} + C_0 \\
    x(t) + A^{-1}b &= e^{tA} + C_0 \\
    x(t) &= e^{tA}(x(0) + A^{-1}b) - A^{-1}b \tag{4.12}
\end{align*}
\]

\(^2\) \(b\) can be a constant vector or a vector expressed by time \(t\ \dot{b}(t)\). The related homogeneous system is \(\dot{x}(t) = Ax(t)\).
where the last line is obtained by making use of the system initial condition in Equation 4.3. Compare the solutions to the linear and inhomogeneous systems (Equation 4.12 and 4.8), an extra constant vector \( A^{-1}b \) appears in the solution to the later system.

### 4.3 A Comprehensive Sensitivity Analysis of the State Behaviour

Given the state behaviour solution to a linear system developed in previous section, we start to explore a generic form of the comprehensive state elasticity analysis in linear systems. It is known that the change in the structure will result in a change in the system gain matrix \( A \), which determines the eigenvalues and eigenvectors, and in turn alters the state behaviour. We first would like to develop the state sensitivity with respect to the compact link as it corresponds to each entry in the gain matrix and therefore, has a direct affect on the eigenvalues and eigenvectors.

To assess the impact resulting from a change in a single compact link gain to the state behaviour (see Equation 4.9), the sensitivity is introduced to measure this influence, i.e., taking partial derivative of the state \( x_i(t) \) with respect to a compact link gain, say \( A_{pq} \), denoted by \( a_{pq} \):

\[
\frac{\partial x_i(t)}{\partial a_{pq}} = \sum_{j=1}^{n} \left( \frac{\partial e^{t\lambda_j}}{\partial a_{pq}} r_{ji} \ell_j^H x(0) + e^{t\lambda_j} \frac{\partial r_{ji}}{\partial a_{pq}} \ell_j^H x(0) + e^{t\lambda_j} r_{ji} \frac{\partial \ell_j^H}{\partial a_{pq}} x(0) \right)
\]

In a linear system, the gain matrix \( A \) is time-invariant and constant, so:

\[
\frac{\partial e^{t\lambda_j}}{\partial \lambda_j} = te^{t\lambda_j}
\]

Substituting it to Equation 4.13 yields:

\[
\frac{\partial x_i(t)}{\partial a_{pq}} = \sum_{j=1}^{n} \left( te^{t\lambda_j} \frac{\partial \lambda_j}{\partial a_{pq}} r_{ji} \ell_j^H x(0) + e^{t\lambda_j} \frac{\partial r_{ji}}{\partial a_{pq}} \ell_j^H x(0) + e^{t\lambda_j} r_{ji} \frac{\partial \ell_j^H}{\partial a_{pq}} x(0) \right)
\]

Equation 4.14 evaluates the overall state sensitivity with respect to a compact link gain. This equation also shows the DDW analysis is actually the eigenvector related component analysis.
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

as the weight sensitivity corresponds to its last two terms. Furthermore, the equation indicates that the variation in the behaviour of \( x_i(t) \) due to the change of compact link gain \( a_{pq} \) can be attributed to three sources:

1. \( \partial \lambda_j / \partial a_{pq} \), eigenvalue sensitivity. The first term in equation captures the influence to a behaviour mode owing to the eigenvalue. The eigenvalue sensitivity with respect to the link gain \( a_{pq} \) is defined and calculated:

\[
S_{\lambda a_{pq}} = \frac{\partial \lambda_j}{\partial a_{pq}} = e_j^H \times r_{jq}
\]

(4.15)

where \( e_j^H \) is \( p^{th} \) component of the left eigenvector \( e_j^H \) and \( r_{jq} \) is \( q^{th} \) component of the right eigenvector \( r_j \) (see Appendix B for proof).

2. \( \partial r_{ji} / \partial a_{pq} \), the ratio between the change of \( i^{th} \) component in \( r_j \) and the change in the compact link gain \( a_{pq} \). It is an element in the entire right eigenvector sensitivity, which is defined as follows:

\[
S_{r a_{pq}}^{r_j} = \frac{\partial r_j}{\partial a_{pq}} = \left[ \begin{array}{c} \frac{\partial r_{j1}}{\partial a_{pq}} \\ \vdots \\ \frac{\partial r_{jn}}{\partial a_{pq}} \end{array} \right]
\]

3. \( \partial e_j^H / \partial a_{pq} \), the ratio between the change of the left eigenvector and the change of the compact link gain. Analogously, it is the left eigenvector sensitivity:

\[
S_{\ell a_{pq}}^{e_j} = \frac{\partial e_j^H}{\partial a_{pq}} = \left[ \begin{array}{c} \frac{\partial e_{j1}^H}{\partial a_{pq}} \\ \vdots \\ \frac{\partial e_{jn}^H}{\partial a_{pq}} \end{array} \right]
\]

Equation 4.14 also suggests as the time increase, the influence of eigenvalue alternation gradually dominates the change of the behaviour mode. We use these new notations to rewrite Equation 4.14:

\[
\frac{\partial x_i(t)}{\partial a_{pq}} = \sum_{j=1}^{n} e^{t \lambda_j} \left\{ tS_{\lambda a_{pq}}^{\lambda_j} r_{ji} e_j^H x(0) + S_{r a_{pq}}^{e_j} e_j^H x(0) + r_{ji} S_{\ell a_{pq}}^{e_j} x(0) \right\}
\]

(4.16)

Compared with the current related works, the left eigenvector sensitivity is introduced into calculating the overall state sensitivity for the first time. The decomposition of the constant \( \alpha^0 \) makes such a comprehensive state sensitivity analysis possible. The state sensitivity is expanded into a linear combination of the eigenvalue sensitivity and the right/left eigenvector sensitivity.
4.4 An Analytical Method to Computing the Eigenvector-related Sensitivity

This section focuses on how to solve Equation 4.16 analytically. The eigenvalue sensitivity, its first term, can be solved by Equation 4.15. However, our challenge is to calculate the right and left eigenvector sensitivity, \( S_{r_{pq}} \) and \( S_{\ell_{pq}} \). Initially, we attempted to solve the eigenvector sensitivities respectively. Nevertheless, we find instead of computing the right/left eigenvector sensitivity individually, they have to be evaluated simultaneously. More specifically, it means while neither of the eigenvector sensitivity is known, their sum as in Equation 4.16 can be evaluated. The complete procedure of developing its analytic solution is provided in the rest of this section.

4.4.1 Computation of the Right Eigenvector Sensitivity

We first attempted at solving the right eigenvector sensitivity. Let \( A \) denote a compact gain matrix of a linear system with \( n \) state variables with \( n \) distinct eigenvalues. In the following discussion, we will explicitly incorporate the eigenvector scalar to indicate its non-uniqueness and evaluate its effect in the sensitivity. From the definition of eigenvector and eigenvalue, we get:

\[
A (c_i r_i) = \lambda_i (c_i r_i)
\]

where \( c_i \) is a scalar. Rearranging the above equation yields:

\[
0 = (A - \lambda_i I) c_i r_i
\]

In order to obtain the right eigenvector sensitivity with respect to the compact link gain, we take partial derivative of both sides in the above equation with regard to an entry in \( A \), e.g., \( A(p, q) \) and use the product rule:

\[
0 = \frac{\partial (A - \lambda_i I)}{\partial A(p, q)} c_i r_i + (A - \lambda_i I) \frac{\partial (c_i r_i)}{\partial A(p, q)}
\]

\[
= c_i \frac{\partial A}{\partial A(p, q)} r_i - c_i \frac{\partial (\lambda_i I)}{\partial A(p, q)} r_i + (A - \lambda_i I) \frac{\partial (c_i r_i)}{\partial A(p, q)}
\]

\[
= c_i \frac{\partial A}{\partial A(p, q)} r_i - c_i \frac{\partial (\lambda_i I)}{\partial A(p, q)} r_i + (A - \lambda_i I) \frac{\partial (c_i r_i)}{\partial A(p, q)}
\]
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

For the first partial derivative in above equation, since each entry in \( A \) is independent of the others, so the result yields:

\[
\frac{\partial A}{\partial A(p, q)} = \begin{pmatrix}
\frac{\partial A(1,1)}{\partial A(p, q)} & \cdots & \frac{\partial A(p, q)}{\partial A(p, q)} \\
\vdots & \ddots & \vdots \\
\frac{\partial A(n,n)}{\partial A(p, q)} & \cdots & \frac{\partial A(A, n)}{\partial A(p, q)}
\end{pmatrix} = \begin{pmatrix} 0 & 0 \\ \vdots & 1 \\ 0 & 0 \end{pmatrix}
\]

The second term is a diagonal matrix with \( ith \) eigenvalue sensitivity to the link gain \( A(p, q) \) at its diagonal. In the last term, the partial derivative of \( c_i r_i \) is what we are looking for, i.e., \( ith \) right eigenvector sensitivity. Expanding the above equation to see more in details:

\[
0 = c_i \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} - c_i \begin{bmatrix} \frac{\partial \lambda_i}{\partial A(p, q)} & 0 \\ \vdots & \frac{\partial \lambda_i}{\partial A(p, q)} \\ 0 & \frac{\partial \lambda_i}{\partial A(p, q)} \end{bmatrix} \begin{bmatrix} r_{i1} \\ \vdots \\ r_{in} \end{bmatrix} + (A - \lambda_i I) \begin{bmatrix} \frac{\partial (c_i r_i)}{\partial A(p, q)} \\ \vdots \\ \frac{\partial (c_i r_i)}{\partial A(p, q)} \end{bmatrix}
\]

\[
= c_i \begin{bmatrix} -S_{\lambda_{aq}} r_{i1} \\ \vdots \\ -S_{\lambda_{aq}} r_{in} \end{bmatrix} + (A - \lambda_i I) \begin{bmatrix} S_{\lambda_{aq}}^{r_{i1}} \\ \vdots \\ S_{\lambda_{aq}}^{r_{in}} \end{bmatrix} + (A - \lambda_i I) \begin{bmatrix} \frac{\partial (c_i r_i)}{\partial A(p, q)} \\ \vdots \\ \frac{\partial (c_i r_i)}{\partial A(p, q)} \end{bmatrix}
\]

Unfortunately, Equation 4.18 cannot be solved for \( S_{\lambda_{aq}}^{r_{i}} \) as the matrix \( (A - \lambda_i I) \) has no inverse (it is singular). Moreover, the mathematical computation reflects a fundamental problem of our eigenvector sensitivity analysis: our definition of the eigenvector sensitivity is not precise enough. The definition of the eigenvector sensitivity:

\[
S_{\lambda_{aq}}^{r_{i}} = \frac{\partial c_i r_i}{\partial A(p, q)} = \frac{c_i^* r_i^* - c_i r_i}{A^* (p, q) - A(p, q)} = \frac{\Delta r_i}{\Delta A(p, q)}
\]

where \( A^* (p, q) \) denotes the new compact link gain and \( c_i^* r_i^* \) is the eigenvector of the new matrix. The problem is that neither \( c_i r_i \) nor \( c_i^* r_i^* \) is unique due to their unfixed scalars. We resort to the idea of eigenvector normalization we have chosen in Equation 4.5, however this does not provide us sufficient information to make the eigenpair unique: the normalization can only determine one eigenvector provided the other is given. Before we proceed further to solve this problem, it is worth having a look at the computation of the left eigenvector sensitivity.
4.4.2 Computation of the Left Eigenvector Sensitivity

The left eigenvector sensitivity can be derived following the same way as the right eigenvector sensitivity. From the definition of the left eigenvector, we have: \((\ell_i^H/c_i)A = \lambda_i(\ell_i^H/c_i)\). Differentiating it with respect to the compact link gain \(A(p, q)\) on both sides and bringing them to the right-hand side:

\[
0 = \frac{\partial (\ell_i^H/c_i)}{\partial A(p, q)} (A - \lambda_i I) + \frac{\ell_i^H}{c_i} \frac{\partial (A - \lambda_i I)}{\partial A(p, q)}
\]

\[
= S_{a_{pq}}^\ell (A - \lambda_i I) + \frac{\ell_i^H}{c_i} \left( \frac{\partial A}{\partial A(p, q)} - S_{a_{pq}}^\lambda I \right) \quad (4.19)
\]

Analogously, the row vector \(S_{a_{pq}}^\ell\) is not unique and has one free variable in its solution. At this point, we can make use of the normalization condition over the eigenpair:

\[
(\ell_i^H/c_i) c_i r_i = 1 \quad (4.20)
\]

which provides an extra constraint to determine that free variable. Taking partial derivative of the above equation with respect to the link gain \(A(p, q)\), renders:

\[
0 = \frac{\partial (\ell_i^H/c_i)}{\partial A(p, q)} (c_i r_i) + \frac{\ell_i^H}{c_i} \frac{\partial (c_i r_i)}{\partial A(p, q)}
\]

\[
= S_{a_{pq}}^\ell (c_i r_i) + \frac{\ell_i^H}{c_i} S_{a_{pq}}^{r_i} \quad (4.21)
\]

By virtue of Equation 4.19 and 4.21, the left eigenvector sensitivity can be solved on the condition that \(c_i\) and \(S_{a_{pq}}^{r_i}\) are determined.

4.4.2.1 The implicit constraint on the eigenvector sensitivity

Let us return to the topic of seeking a deterministic solution to the right eigenvector sensitivity, although we fail to solve it with all the information we have, an implicit constraint is identified in this specific application context, the state behaviour sensitivity with respect to the compact link gain \((\partial x_i/\partial a_{pq})\) is a unique constant in spite of the non-unique value of the eigenvector sensitivity. Let us change the state sensitivity in Equation 4.16 to gather the eigenvector-related sensitivity. Since the behaviour modes are independent of each other, we can consider them separately. Thereby the sensitivity within an individual mode should be a constant. As a result, the terms associated with the eigenvector sensitivity should be a constant. After getting rid of the system initial condition \(x(0)\) (a fixed vector), the eigenvector-related sensitivity is obtained.
to be a constant vector. For clarity, we use the formulas to demonstrate the above description as follows.

\[
\frac{\partial x_i(t)}{\partial a_{pq}} = \sum_{j=1}^{n} e^{\lambda_j t} \left\{ t S_{apq}^{\lambda_j} r_{ji} \ell_j^H x(0) + S_{apq}^{\lambda_j} \ell_j^H x(0) + r_{ji} S_{apq}^{\ell_H} x(0) \right\} \equiv \text{constant}
\]

\[
\Rightarrow e^{\lambda_j t} \left( t S_{apq}^{\lambda_j} r_{ji} \ell_j^H x(0) + S_{apq}^{\lambda_j} \ell_j^H x(0) + r_{ji} S_{apq}^{\ell_H} x(0) \right) \equiv \text{constant}
\]

\[
\Rightarrow S_{apq}^{\lambda_j} \ell_j^H x(0) + r_{ji} S_{apq}^{\ell_H} x(0) \equiv \text{constant vector} \quad (4.22)
\]

Our primary focus is on how to solve the above distilled expression. We know that for an eigenvector \( \mathbf{v} \), any scalar multiple \( c \mathbf{v} \) is also an eigenvector. The scalar is then explicitly added in Equation 4.22 which is rewritten to be:

\[
\frac{\partial (c_i r_{ij})}{\partial a_{pq}} \frac{\ell_i^H c_i}{c_i} + c_i r_{ij} \frac{\partial (\ell_i^H c_i)}{\partial a_{pq}} = S_{apq}^{\ell_H} \ell_j^H + r_{ji} S_{apq}^{\ell_H} \equiv \mathbf{d}^i
\]

(4.23)

where \( \mathbf{d}^i \) is a notation for a \( 1 \times n \) vector, and the subscripts are swapped with \( i \) denoting the mode and \( j \) referring to the state variable \( x_j \). Owing to this implicit constraint, we know that \( \mathbf{d}^i \) is a vector with constants, which will yield a constant value after multiplying \( x(0) \).

From previous analysis, we are aware that the left eigenvector sensitivity can be solved by knowing the associated right eigenvector sensitivity while the right eigenvector sensitivity has only one unknown. Therefore, we are going to express the eigenpair sensitivity in terms of that unknown, then substitute them back to Equation 4.23 and examine the outcome. The second order linear system (see Section 4.2.2) will be used to explain the procedure. Without loss of generality, a generic linear form will also be adopted when addressing this problem.

In this example, the dominant mode sensitivities (associated with \( \lambda_1 = 2 \)) with respect to the compact link gain \( a_{11} \) is chosen. In parallel, the compact link is denoted as \( a_{pq} \) and the mode of interest is \( i \) in the generalized linear system. First, let us look at the solution to the right eigenvector sensitivity. Start with the Equation 4.18. As we assume \( A \) has distinct eigenvalues, Equation 4.18 is a linear system with \( n - 1 \) independent functions and \( n \) unknowns. Thus, there is one free variable in the solution. We choose one unknown as the free variable to represent
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

the solution. Row reduction is used to produce the outcome as follows:

\[
S_{rik} : \text{ unknown} \\
S_{r1} = \alpha_{i1}S_{rik} + \beta_{i1}c_i \\
\vdots \\
S_{rin} = \alpha_{in}S_{rik} + \beta_{in}c_i \quad (4.24)
\]

where \( S_{rik} \) is the \( k \)th component of \( i \)th eigenvector sensitivity, and \( \alpha_i = \begin{bmatrix} \alpha_{i1} \\ \vdots \\ \alpha_{in} \end{bmatrix} \), \( \beta_i = \begin{bmatrix} \beta_{i1} \\ \vdots \\ \beta_{in} \end{bmatrix} \) are both constant vectors. Similar notations will be seen, and their interpretations are omitted afterwards. For the computation convenience, we use another eigenpair value associated with 1st mode:

\[
r_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \ell_1^H = \begin{pmatrix} 4 & 1 \\ 5 & 5 \end{pmatrix}
\]

The counterpart in the concrete example:

\[
S_{r11} : \text{ unknown} \\
S_{r12} = S_{r11} - \frac{1}{5}c_1
\]

At this point, the left eigenvector sensitivity is to be analyzed. By utilizing Equation 4.19, we follow the similar calculation as in the right eigenvector sensitivity and the solution is given below:

\[
S_{i1}^H \\
S_{i1}^H = u_{i1}S_{i1}^H + v_{i1}/c_i \\
\vdots \\
S_{in}^H = u_{in}S_{i1}^H + v_{in}/c_i \quad (4.25)
\]

where \( u_i \) and \( v_i \) are constant vectors. Likewise, the counterparts in the concrete example are:

\[
S_{i11}^H \\
S_{i12}^H = \frac{1}{4}S_{i11}^H - \frac{1}{25}c_1
\]
It is time to make use of the eigenpair normalization. Let us substitute Equation 4.24 and 4.25 for $S_r$ and $S^H$ in Equation 4.21. The solution to the left eigenvector sensitivity can be described by the unknown, i.e., a component in the right eigenvector sensitivity:

$$S_{\ell_{i}}^H = g_{i_{1}}S^{r_{i_{1}}}/c_{i_{1}}^2 + h_{i_{1}}/c_{i_{1}}$$

...  

$$S_{\ell_{i_{n}}}^H = g_{i_{n}}S^{r_{i_{n}}}/c_{i_{n}}^2 + h_{i_{n}}/c_{i_{n}}$$  \hspace{1cm} (4.26)

where $g_{i}$ and $h_{i}$ are both constant vectors. Analogously, our example yields similar outcomes:

$$S_{\ell_{1_{1}}}^H = -\frac{4}{5}S^{r_{1_{1}}}/c_{1_{1}}^2 + \frac{8}{125c_{1}}$$

$$S_{\ell_{1_{2}}}^H = -\frac{1}{5}S^{r_{1_{1}}}/c_{1_{1}}^2 - \frac{3}{125c_{1}}$$

Finally, recall the constraint in Equation 4.23, if we replace the eigenpair sensitivities with their solutions, the equation reduces to be:

$$d_{ij} = S_{pq}^r \ell_{i_{1}}^H/c_{i_{1}} + (c_{i}r_{ij}) S_{pq}^H$$

$$= (\alpha_{ij} S^{r_{i_{1}}} + c_{i}r_{ij}) \ell_{i_{1}}^H/c_{i_{1}} + c_{i}r_{ij} \left( \frac{S^{r_{i_{1}}}}{c_{i_{1}}^2} g_{i_{1}}^H + \frac{h_{i_{1}}^H}{c_{i_{1}}} \right)$$

$$= \frac{S^{r_{i_{1}}}}{c_{i_{1}}} (\alpha_{ij} \ell_{i_{1}}^H + r_{ij}g_{i_{1}}^H) + \beta_{ij} \ell_{i_{1}}^H + r_{ij}h_{i_{1}}^H$$

$$= \mu_{i_{1}}^H S^{r_{i_{1}}} + \sigma_{i_{1}}^H$$  \hspace{1cm} (4.27)

where $\mu_{1} = \begin{bmatrix} \alpha_{ij} \ell_{i_{1}} + r_{ij}g_{i_{1}} \\ \alpha_{ij} \ell_{i_{n}} + r_{ij}g_{i_{n}} \end{bmatrix}$ and $\sigma_{1} = \begin{bmatrix} \beta_{ij} \ell_{i_{1}} + r_{ij}h_{i_{1}} \\ \beta_{ij} \ell_{i_{n}} + r_{ij}h_{i_{n}} \end{bmatrix}$. Meanwhile, we apply the same procedure to the example and obtain:

$$d_{11}^{11} = S^{r_{1_{1}}} \ell_{1_{1}}^H/c_{1_{1}} + c_{1_{1}}r_{1_{1}} S_{1_{1}}^H$$

$$= S^{r_{1_{1}}} \left( \frac{4}{5c_{1}} \frac{1}{5c_{1}} \right) + c_{1_{1}} \left( \frac{-4S^{r_{1_{1}}}}{5c_{1}^2} + \frac{8}{125} - \frac{3}{125} \right)$$

$$= \left( \frac{8}{125} \frac{-3}{125} \right)$$  \hspace{1cm} (4.28)

Post-multiplying Equation 4.27 with the initial conditions $x(0)$ and $e^{t\lambda_{i}}$ renders the eigenvector-related sensitivity in terms of a specific mode in the state sensitivity (see Equation 4.14). $d_{ij}$ refers to $i^{th}$ eigenvector-related (or eigenpair) sensitivity in $x_{j}$ henceforth. Furthermore, we have two important conclusions derived from the above equation and analysis procedure:
1. If we choose the scalar $c_i$ in a form as in Equation 4.20, the scalar does not impact the eigenvector-related sensitivity in the evaluation of the state sensitivity. It is apparent in the concrete example, where $d_{11}$ does not involve the scalar $c_1$. In regard to the general form of the linear system, we can verify this by Equation 4.27. The term $\frac{S_{rk}}{c_i}$ can be rearranged as:

$$S_{rk} = c_i \frac{\partial (c_i r_{ik})}{\partial a_{pq}} = c_i \frac{\partial r_{ik}}{\partial a_{pq}}.$$ Subsequently, we can remove the scalar in Equation 4.23 and it becomes:

$$d_{ij}^{\text{ij}} = \frac{\partial r_{ij}}{\partial a_{pq}} \ell_i^H + r_{ij} \ell_i^H$$

(4.29)

2. The unknown disappears in the concrete example, see Equation 4.28. In the generalized system, it is not directly discernible. However if we expand all the constants along the eigenvector sensitivity computation process, we would finally come to realize that the unknown $S_{rk}$ is canceled out as $\mu_i$ is a zero vector. An important conclusion is derived with further investigation: for a specific behaviour mode $i$ and a state $x_j$, the eigenpair-related sensitivity in the state sensitivity (in Equation 4.23) yields a constant vector (before multiplying the initial condition) and can be calculated as $j^{\text{th}}$ row in Equation 4.30, where $(A - \lambda_i)^\#$ denotes the group inverse matrix of $(A - \lambda_i)$. The proof of this result is to be unfolded in the subsequent section. This conclusion also indicates that the eigenpair sensitivity must be calculated simultaneously as they are dependent of each other:

$$D_i = -(A - \lambda_i) \# \frac{\partial A}{\partial a_{pq}} r_{ij} \ell_i^H - r_{ij} \ell_i^H \frac{\partial A}{\partial a_{pq}} (A - \lambda_i) \#$$

(4.30)

### 4.4.3 Proof of the Analytical Solution to the Eigenvector-related Sensitivity

We are now to prove the analytic formulation of calculating the eigenvector-related sensitivity in Equation 4.30. In order to present the proof rigorously, we bring in some mathematics terms to facilitate our explanation. On the other hand, we change our notations somewhat for convenience: the eigenvector sensitivity notation $S_{a_{pq}}$ is replaced by $r'_{ij}$, and $S_{a_{pq}}^H$ by $\ell_{ij}^H$ as well. Let us start from Equation 4.29 and we show it with the new notations:

$$d_{ij}^{\text{ij}} = r'_{ij} \ell_i^H + r_{ij} \ell_i^H$$

(4.31)
In a broader sense, for all \( n \) state variables in a system, we have \( n \) such equations. Therefore we can rewrite them into a compact matrix form:

\[
D^i = \begin{bmatrix}
d_{i1} \\ \vdots \\ d_{in}
\end{bmatrix} = \begin{bmatrix}
r_{i1}^r \ell_1^H + r_{i1} \ell_1^H' \\ \vdots \\ r_{in} \ell_1^H + r_{in} \ell_1^H'
\end{bmatrix} = r_{i1}^r \ell_1^H + r_{i1} \ell_1^H' \quad (4.32)
\]

where \( D^i \) is an \( n \times n \) matrix associated with \( i^{th} \) behaviour mode and each row corresponds to the eigenvector sensitivities in relation with different state variable. So far, we have modified our objective function to facilitate the proof. The major conclusion we have derived in this chapter is:

For a matrix \( A \) whose entries are all real values, \( D^i \) associated with a distinct eigenvalue (\( \lambda_i \)) restricted to satisfy a constraint of the eigenpair in the form of \( r_i^H \ell_1 = 1 \) yields a constant matrix, and

\[
D^i = -(A - \lambda_i I)^\# A' r_i^H - r_i^H A'(A - \lambda_i I)^\# ,
\]

where \( A' \) is a notation for \( \partial A/\partial a_{pq} \) for short, and \( (A - \lambda_i I)^\# \) refers to the group inverse of matrix \( (A - \lambda_i I) \).

**Proof**

If \( P_{n \times n-1} \) is a matrix whose columns form an orthonormal (i.e., orthogonal and normalized to be 1) basis for \( R(A - \lambda_i I) \) (\( R(\cdot) \) denotes range and \( N(\cdot) \) denotes the nullspace). The orthonormal basis can be formed by performing the Gram-Schmidt orthogonalization. Then \( W = (r_1 | P) \) is nonsingular and it is easy to verify that:

\[
W^{-1} = \left( P^H (I - r_1 \ell_1^H) \right)^4 \quad (4.33)
\]

Matrix \( W^{-1} (A - \lambda_i I) W \) has the form

\[
W^{-1} (A - \lambda_i I) W = \begin{pmatrix}
0 & 0 \\
0 & P^H (I - r_1 \ell_1^H) (A - \lambda_i I) P
\end{pmatrix}
\]

\[
= \begin{pmatrix}
0 & 0 \\
0 & P^H (A - \lambda_i I) P
\end{pmatrix}
\]

since \( \lambda_i \) is simple, \( P^H (A - \lambda_i I) P \) is nonsingular and the group inverse of \( (A - \lambda_i I) \)

\[
(A - \lambda_i I)^\# = W \begin{pmatrix}
0 & 0 \\
0 & \{ P^H (A - \lambda_i I) P \}^{-1}
\end{pmatrix} W^{-1}
\]

3 Vector \( r_{i(1 \times n)} \) forms the first column of matrix \( W \) while \( P_{n \times n-1} \) fill in the rest column space.

4 Vector \( \ell_1^H \) forms the first row of matrix \( W \) while \( P^H (I - r_1 \ell_1^H) \) fill in the rest row space.
is well defined. We can further verify that the non-uniqueness of \( r_i \) and \( \ell_i \) will not affect the computation of \((A - \lambda I)^\#\). Additional material on group inverse can be found (Campbell and Meyer, 1979). Since we will make use of the group inverse’s properties in our analysis, so we introduce some of them now.

1. The nullspace of the original matrix and its group inverse matrix is the same, i.e., \( N(A - \lambda I) = N(A - \lambda I)^\# \)

2. If \( A \) is a group matrix and \( b \in R(A) \), then the set of all solutions for \( p \) in \( Ap = b \) is given by \( p = A^#b + N(A) \).

3. For an eigenvalue \( \lambda \),
   \[
   \lambda^# = \begin{cases} 
   1/\lambda, & \lambda \neq 0 \\
   0, & \lambda = 0
   \end{cases}
   \]
   A vector \( r \) is an eigenvector for \( A \) corresponding to the eigenvalue \( \lambda \) if and only if \( r \) is an eigenvector for \( A^# \) corresponding to \( \lambda^# \), i.e., \( Ar = \lambda r \) if and only if \( A^#r = \lambda^#r \).

From Equation 4.17, we know that
   \[
   (A' - \lambda'_i I)r_i + (A - \lambda_i I)r_i' = 0
   \]
   with addition to property 1 and 2, there must exist a scalar \( \beta \) such that
   \[
   r_i' = \beta r_i - (A - \lambda_i I)^#(A - \lambda_i I)'r_i
   = \beta r_i - (A - \lambda_i I)^#A'r_i - (A - \lambda_i I)^#\lambda_i' r_i
   = \beta r_i - (A - \lambda_i I)^#A'r_i
   \] (4.34)
   Recall the normalization condition we have established in Equation 4.21, we rewrite it to be:
   \[
   \ell_i'^H r_i' + r_i'^H \ell_i' = 0
   \] (4.35)
   Utilizing Equation 4.34 and 4.35 derives the scalar \( \beta = -r_i'^H \ell_i' \), which is substituted into Equation 4.34 to produce the following expression:
   \[
   r_i' = -r_i'^H \ell_i' r_i - (A - \lambda_i I)^#A'r_i
   \] (4.36)
   Analogously, we can obtain the general solution to the left eigenvector sensitivity by Equation 4.19:
   \[
   \ell_i'^H = \beta \ell_i'^H - \ell_i'^H A'(A - \lambda_i I)^#
   \]
Plug it into Equation 4.35 to generate \( \beta = -\ell_i^H r_i' \). As a result, the left eigenvector sensitivity can be sorted out to be:

\[
\ell_i^H = -\ell_i^H r_i' \ell_i^H - \ell_i^H A' (A - \lambda_i I)^# \tag{4.37}
\]

In light of Equation 4.36, 4.37 and 4.35, our objective Equation 4.32 is finally obtained:

\[
D_i = r_i' \ell_i + r_i \ell_i' = \begin{cases}
-(r_i^H \ell_i') r_i^H - r_i (\ell_i^H r_i') \ell_i^H - (A - \lambda_i I)^# A' r_i \ell_i^H - r_i \ell_i^H A' (A - \lambda_i I)^# \\
-(A - \lambda_i I)^# A' r_i \ell_i^H - r_i \ell_i^H A' (A - \lambda_i I)^#
\end{cases} \tag{4.38}
\]

### 4.4.4 Analytical Solutions to the Eigenvector-related Sensitivity with respect to Various Structure Components

The eigenvector-related sensitivity was discussed in terms of the compact link gain in the previous section. This section continues to study the eigenpair sensitivity. Nevertheless different structure components are investigated. This shares many similarities with the eigenvalue elasticity analysis with respect to the pathway and causal link in Appendix C. Readers who are familiar with that part can skip this section but read its conclusion in italic only.

First let us look at how to develop eigenvector sensitivity with respect to pathway gain. Pathway is the path that starts from and end with a state variable (both can be the same variables), in addition, it does not contain any other pathways within itself (Mojtahedzadeh, 1997). Let those pathways be \( P_{pq1}, P_{pq2}, \ldots, P_{pqm} \), where \( m \) is the number of pathways from \( x_q \) to \( x_p \). We use \( g_{pq1}, g_{pq2}, \ldots, g_{pqm} \) to represent those pathway gains respectively. Let the compact link gain from \( x_q \) to \( x_p \) be \( a_{pq} \). If we sum up the gains of the pathways who share the same starting state variable and the ending state variable, they are equivalent to the corresponding compact link gain:

\[
a_{pq} = g_{pq1} + \ldots + g_{pqm} \tag{4.39}
\]

From the definition of the right eigenvector sensitivity with respect to a pathway \( P_{pqj} \), we have:

\[
S_{pqj}^r = \frac{\partial r_i}{\partial g_{pqj}}
\]
By utilizing the chain rule and Equation 4.39:

$$S_{pqj}^i = \frac{\partial r_1}{\partial a_{11}} \frac{\partial a_{11}}{\partial g_{pqj}} + \ldots + \frac{\partial r_1}{\partial a_{pq}} \frac{\partial a_{pq}}{\partial g_{pqj}} + \ldots + \frac{\partial r_1}{\partial a_{nn}} \frac{\partial a_{nn}}{\partial g_{pqj}} \quad (4.40)$$

It is known that the partial derivative of the compact link gain to the pathway gain is zero when the pathway does not lie in the compact link, otherwise it is one by Equation 4.39. Furthermore, one pathway can only contribute to one compact link. Therefore, Equation 4.40 can be simplified as:

$$S_{pqj}^i = \frac{\partial r_1}{\partial a_{pq}} \frac{\partial a_{pq}}{\partial g_{pqj}} = \frac{\partial r_1}{\partial a_{pq}} = S_{a_{pq}}^i \quad (4.41)$$

The above equation suggests that the right eigenvector sensitivity with respect to the pathway is equal to that with respect to the compact link gain where this pathway lies. Likewise, their eigenvector-related sensitivities are equal:

$$d_j^{\{ap_{pq}\}} = d_j^{\{pq\}} \quad (4.42)$$

Let us focus on the causal link. The pathway gain can be computed by multiplying its causal link gains (links between auxiliaries, state variables and constants):

$$g_{pu} = \prod_{e_k \in P_u} g_{e_k} \quad (4.43)$$

We are aware that one causal link can contribute to more than one pathways. By utilizing the chain rule, we can write the eigenvector sensitivity with respect to the causal link gain:

$$S_{e_k}^i = \frac{\partial r_1}{\partial g_{e_k}} = \frac{\partial r_1}{\partial g_{p_1}} \frac{\partial g_{p_1}}{\partial g_{e_k}} + \ldots + \frac{\partial r_1}{\partial g_{p_s}} \frac{\partial g_{p_s}}{\partial g_{e_k}} \quad (4.44)$$

where subscript $s$ denotes the number of total pathways in the system. Equation 4.43 suggests:

$$\frac{\partial g_{pu}}{\partial g_{e_k}} \begin{cases} = g_{pu} / g_{e_k} : P_u \in e_k \\ = 0 : otherwise \end{cases}$$

Therefore, Equation 4.44 is rewritten as:

$$S_{e_k}^i = \sum_{e_k \in P_u} \frac{\partial r_1}{\partial g_{pu}} \frac{g_{pu}}{g_{e_k}} = \sum_{e_k \in P_u} \frac{g_{pu}}{g_{e_k}} S_{pu}^i \quad (4.45)$$

The equation above shows the eigenvector sensitivity to the causal link gain is equal to the summation of the sensitivity to the pathway which contain the causal link multiplied by their
gain ratio. For the eigenvector-related sensitivity with respect to causal link gain, we can calculate it as follows:

\[ d^{ij}(e_k) = S^{r_{ij}}i + r_{ij} S^{e_{ek}}i \]

\[ = \left( \sum_{u=1}^{s} g_{pu} S^{r_{ij}}p_{u} \right) e^H_i + r_{ij} \left( \sum_{u=1}^{s} g_{pu} S^{e_{ek}}p_{u} \right) \]

\[ = \sum_{u=1}^{s} g_{pu} \left( S^{r_{ij}}p_{u} e^H_i + r_{ij} S^{e_{ek}}p_{u} \right) p_{u} \in a_{pq} \]

\[ = \sum_{u=1}^{s} g_{pu} D^{ij(a_{pq})} \]

Equation 4.41 and 4.45 show the relationships between different types of eigenvector sensitivity and Equation 4.46 proves the eigenvector-related sensitivity with respect to the causal link gain can be obtained by a series of linear operations on the counterpart with respect to the compact link gain.

4.4.4.1 Various sensitivity computation in a linear system

We return to the linear example (Equation 4.10) for two purposes: elaborate the computation of various types of eigenvector-related sensitivity (MATLAB is used in the computation); demonstrate how to assess the eigenvector-related sensitivity with respect to different system elements. The variable of interest is \( x \) and we have obtained its solution in previous discussion.

The trajectory of \( x \) will be growing exponentially steered by the first mode whereas the second behaviour mode will dissipate as time goes. Thereby we focus on the dominant mode associated with \( \lambda_1 = 2 \). The stock-flow diagram of this system is presented in Figure 4.2 to assist our analysis.

Let us begin with constructing the group inverse of \( (A - \lambda_1 I) \), \( (A - \lambda_1 I)^\# \). The eigenvectors are based on Table 4.1. First we configure matrix \( P \) whose columns form an orthonormal basis for \( (A - \lambda_1 I) \):

\[ P = \begin{bmatrix} -0.2425 \\ 0.9701 \end{bmatrix} \]
Hence it is not difficult to obtain the following results by MATLAB. The notations are explained as follows: \([\cdot]\) isolates columns in a matrix; \([-\cdot]\) isolates rows in a matrix; \((\cdot)^\#\) represents a group inverse matrix.

\[
W = [r_1|P] = \begin{bmatrix} 0.7071 & -0.2425 \\ 0.7071 & 0.9701 \end{bmatrix}
\]

\[
W^{-1} = \left[ \ell_1^H \right] P^H (I - r_1 \ell_1^H) = \begin{bmatrix} 1.1314 & 0.2828 \\ -0.8246 & 0.8246 \end{bmatrix}
\]

\[
(A - \lambda_1 I)^\# = W \begin{bmatrix} 0 & 0 \\ 0 & \{P^H (A - \lambda_1 I) P\}^{-1} \end{bmatrix} W^{-1}
= \begin{bmatrix} -0.04 & 0.04 \\ 0.16 & -0.16 \end{bmatrix}
\]

Figure 4.2 – Stock and flow diagram of the linear system
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

106

Compact link $x \rightarrow \dot{x}: A(1,1)$

$y \rightarrow \dot{x}: A(1,2)$

$x \rightarrow \dot{y}: A(2,1)$

$y \rightarrow \dot{y}: A(2,2)$

Eigenvector-related sen. before

$0.064 -0.024$

$-0.096 0.136$

$-0.024 -0.016$

$-0.064 0.024$

Eigenvector-related sen. after $0.2$ $-0.6$ $0$ $0.2$

Eigenvalue sen. $0.8$ $0.8$ $0.2$ $0.2$

Table 4.2 – Eigenvector-related sensitivity and eigenvalue sensitivity of 1st mode

The eigenvector-related sensitivity with respect to $a_{11}$ in terms of 1st mode is given by Equation 4.30:

$$D_{1}^{(a_{11})} = -(A - \lambda_{1}I)^\# \frac{\partial A}{\partial a_{11}} r_{1}\ell_{1}^H - r_{1}\ell_{1}^H \frac{\partial A}{\partial a_{11}} (A - \lambda_{1}I)^\#$$

$$= \begin{pmatrix}
-0.04 & 0.04 \\
0.16 & -0.16
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
0.7071 \\
0.7071
\end{pmatrix}
\begin{pmatrix}
1.1314 & 0.2828 \\
0 & 0
\end{pmatrix}
\begin{pmatrix}
-0.04 & 0.04 \\
0.16 & -0.16
\end{pmatrix}$$

$$= \begin{pmatrix}
0.064 & -0.024 \\
-0.096 & -0.064
\end{pmatrix}$$

The first row of $D_{1}^{(a_{11})}$ contributes to the eigenvector-related sensitivity in terms of state variable $x$ while the second row corresponds to the counterpart in $y$. Compare the first row with the result we get in Section 4.4.2.1 by another method, we obtain identical answer: $[8/125 - 3/125]$. The eigenvector-related sensitivity with respect to the compact link gain before and after multiplying by the initial conditions are shown in Table 4.2, as well as the eigenvalue sensitivity.

From Figure 4.2, we identify four pathways and lie in four different compact links. Table 4.3 lists the pathway gains and the compact links they pass through. In addition to the conclusion that the pathway sensitivity is equal to the corresponding compact link sensitivity, so their eigenvector-related sensitivities are also equivalent as depicted in Table 4.2 and Table 4.3. The computation of the eigenvalue sensitivity with respect to pathway gain or causal link gain is out of the scope, readers can find more details in (AbdelGawad et al., 2005).

The causal links are marked in Figure 4.2. Their sensitivities can be assessed by adding up the pathways sensitivity which pass through that link multiplying their gain ratio, so does the eigenvector-related sensitivity which is computed by Equation 4.46. For example, the
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

107

Table 4.3 – Various sensitivities with respect to pathway gains of 1st mode

eigenvector-related sensitivity of $e_2$ is calculated:

$$
\mathbf{d}^{11}(e_2) = \frac{g_{p1}}{g_{e_2}} \mathbf{d}^{11}(a_{11}) + \frac{g_{p4}}{g_{e_2}} \mathbf{d}^{11}(a_{12})
$$

$$
= \frac{1}{1} \ast 0.2 + \frac{1}{1} \ast (-0.6)
$$

$$
= -0.4
$$

Furthermore, the overall state sensitivity with respect to the causal link gain in regard to the dominant mode is also calculated using Equation 4.46. Table 4.4 exhibits all the information (the eigenvector-related sensitivity is displayed after multiplying the initial condition).

Table 4.4 – Sensitivity with respect to causal link gains of 1st mode

From the above analysis, we can claim that the eigenvector-related sensitivity is predominantly affected by the initial condition and remains constant while the influence from the eigenvalue is impacted by the time factor. Table 4.4 shows $e_6$ has the most significant affect to $x$ in terms of the eigenvector. When we increase the gain of $e_6$, the amplitude of $x$ will decrease by 0.6 of that amount from the perspective of the eigenvector. In total, $e_2$ plays the most important role in the behaviour of $x$ as it gives rise to the greatest impact in terms of the eigenvalue.
4.5 A Case Study of the Eigen-based Elasticity Analysis

This section aims to show the efficient identification of high leverage parameters that the eigenvector-related analysis could perform. In addition, EEA is utilized to compare different policies generated by them. A concrete example of an inhomogeneous system, the labour-inventory model (Saleh et al., 2006) is studied to illustrate different perspectives of these two methods. In this example, the oscillatory behaviour analysis is discussed at length. We argue that it is important to examine influence of the dominant structural components in terms of both eigenvalue and behaviour weight before designing the comprehensive policy.

The policy generated by these two methods focuses on the parameters in this case study. As described in optimization related approaches in the literature review chapter, system parameters are the primary focus studied to in improve the models. They are “implementations” of various physical realities in the system, and it is straightforward to implement the modifications. This section presents a comprehensive eigen-based analysis over parameters, involving its analytic eigenvalue and eigenvector analysis.

4.5.1 Analytical Eigenvalue Elasticity Analysis

The analytic formulation of eigenvalue elasticity with respect to the parameter is first briefly introduced. Start with the eigenvalue definition:

\[ A r_i = \lambda_i r_i \]

Differentiating this equation with respect to the parameter \( p \) gives:

\[ \frac{\partial A}{\partial p} r_i + A \frac{\partial r_i}{\partial p} - \frac{\partial \lambda_i}{\partial p} r_i - \lambda_i \frac{\partial r_i}{\partial p} = 0 \]

\[ \left( \frac{\partial A}{\partial p} - \frac{\partial \lambda_i}{\partial p} I \right) r_i + (A - \lambda_i I) \frac{\partial r_i}{\partial p} = 0 \] (4.47)

Equation 4.47 is then pre-multiplied by \( \ell_i^\mu \). The second term becomes zero and it can be rewritten as:

\[ \ell_i^\mu \frac{\partial \lambda_i}{\partial p} r_i = \ell_i^\mu \frac{\partial A}{\partial p} r_i \]

\[ \frac{\partial \lambda_i}{\partial p} \ell_i^\mu r_i = \ell_i^\mu \frac{\partial A}{\partial p} r_i \]

\[ \frac{\partial \lambda_i}{\partial p} = \ell_i^\mu \frac{\partial A}{\partial p} r_i \] (4.48)
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

Elasticity measure varies in terms of real and complex eigenvalues. The real-valued eigenvalue elasticity with respect to the parameter \( p \) is defined as usual.

\[
\varepsilon = \frac{\partial \lambda_i}{\partial p} = \ell_i^H \frac{\partial A}{\partial p} r_i \frac{p}{\lambda_i}
\]  

(4.49)

where the partial derivative of the matrix \( A \) can be solved by decomposing all of its entries into terms of parameters and constants. This will be seen a lot in our experiment section. On the other hand, there are a couple of elasticity definitions associated with complex eigenvalue available (Phaff et al., 2006). We adopt the one to treat the eigenvalue separately in terms of real and imaginary components (Güneralp, 2006b):

\[
\varepsilon_R = \frac{\partial Re(\lambda_i)}{\partial p} = Re \left( \ell_i^H \frac{\partial A}{\partial p} r_i \right) \frac{p}{Re(\lambda_i)}
\]  

(4.50)

\[
\varepsilon_I = \frac{\partial Im(\lambda_i)}{\partial p} = Im \left( \ell_i^H \frac{\partial A}{\partial p} r_i \right) \frac{p}{Im(\lambda_i)}
\]  

(4.51)

Above definition has two advantages:

- The sign of elasticity is defined relative to the eigenvalue.
- The change in the exponential envelope or the frequency of oscillation is measured relative to the appropriate component of the eigenvalue.

### 4.5.2 Analytical Eigenvector (Weight) Elasticity Analysis

Recall the DDW analysis (Saleh et al., 2006), we examine this approach from another perspective. Equation 4.2 and 4.9 are two solutions to the linear system (they are displayed in Equation 4.52). In DDW, \( w \) is numerically assessed before and after giving a disturbance to a parameter, the weight elasticity is then evaluated accordingly. Whereas we use the formulation decomposing \( w \) into the eigenvector-related factors and system initial condition as the starting point of our analysis. The weight analysis is hence boiled down to be the eigenvector-related component analysis which is promising to be solved in light of our previous study.

\[
x_i(t) = w_1 \ell_1 \mathbf{x}(0) e^{\lambda_1 t} + w_2 \ell_2 \mathbf{x}(0) e^{\lambda_2 t} + \cdots + w_n \ell_n \mathbf{x}(0) e^{\lambda_n t} 
\]  

(4.52)
Consider $w_{ji}$, i.e., the weight associated with the $j^{th}$ mode in state variable $x_i$, when the behaviour mode $j$ is characterized by the real-valued eigenvalue, its elasticity is defined by Equation 4.53.

$$
\varepsilon_{w_{ji}} = \frac{\partial w_{ji}/w_{ji}}{\partial p/p} = \frac{\partial \left( r_{ji} \ell_j^H x(0) \right)}{\partial p} \frac{p}{w_{ji}} \\
= \left( \frac{\partial r_{ji}}{\partial p} \ell_j^H \right) x(0) \frac{p}{w_{ji}} + \left( \frac{\partial \ell_j^H}{\partial p} r_{ji} \right) x(0) \frac{p}{w_{ji}} \\
= \left( \frac{\partial r_{ji}}{\partial p} \ell_j^H + r_{ji} \frac{\partial \ell_j^H}{\partial p} \right) x(0) \frac{p}{w_{ji}} \\
\left( \frac{\partial w_{ij}}{\partial p} \right)
$$

(4.53)

where the term within the bracket is the eigenvector sensitivity to a parameter. It is similar to the eigenvector-related sensitivity with respect to a compact link $A(p, q)$ in Equation 4.30.

$$
\left( \frac{\partial r_{ji}}{\partial p} \ell_j^H + r_{ji} \frac{\partial \ell_j^H}{\partial p} \right) = \left\{ -(A - \lambda_j I)^\# \frac{\partial A}{\partial p} r_{ji} \ell_j^H - r_{ji} \ell_j^H \frac{\partial A}{\partial p} (A - \lambda_j I)^\# \right\}
$$

(4.54)

where the subscript $i$ represents the $i^{th}$ row of $\{ \}$. 

In regard to the weight elasticity associated with the mode characterized by the complex-valued eigenvalues, we have different treatment. For a real-valued square matrix, its complex eigenvalues must occur in complex conjugate pairs. Thus, say, if one eigenvalue is $\lambda = a + bi$, there must exist another eigenvalue equals to $a - bi$. If an eigenvalue is complex, its eigenvectors will be in general vectors with complex entries. This is easy to see from the following equations:

$$
Ar = \lambda r, \quad r \neq 0,
$$

Taking complex conjugates of this equation, we obtain:

$$
A \bar{r} = \bar{\lambda} \bar{r} = \bar{\lambda} \bar{r}
$$

where the equality follows from the fact that $A$ has real entries. This means exactly that $\bar{\lambda}$ is an eigenvalue of $A$, with $\bar{r}$ as an eigenvector. It is not difficult to prove that the weights associated with the conjugated eigenvalues also come in conjugates. Conjugated eigenvalues characterized modes are combined to form an oscillatory behaviour (it is proved in Appendix A). Assume we denote the weights of the conjugated modes as $c + di$ and $c - di$ respectively,
the combined oscillatory behaviour is formulated as \(2\sqrt{c^2 + d^2} e^{at} \sin(\theta - bt)\), where \(\sin\theta = c/\sqrt{c^2 + d^2}\). As time goes, the amplitude will grow or shrink depending on whether \(a > 0\) or \(a < 0\). The frequency is determined by the imaginary part of the eigenvalue, \(b/2\pi\).

The definition of the weight elasticity associated with the exponential behaviour mode is no longer suitable for that of the oscillatory mode. As both the real and imaginary parts affect the amplitude in a form of the root of their square sum. It is advisable to evaluate the change in real and imaginary parts individually, and then compare the absolute value of the changed weight with the original one. Thus the weight elasticity regarding the amplitude in the oscillatory mode is defined as follows:

\[
\begin{align*}
    s &= \frac{\partial w}{\partial p} \\
    \Delta w &= s \cdot \Delta p \\
    w &= \Delta w + w_0 \\
    \frac{\partial w}{\partial p} &= \frac{\Delta w}{|w_0|} \frac{\Delta p}{p} = \frac{abs(w) - abs(w_0)}{abs(w_0)} \frac{\Delta p}{p}
\end{align*}
\]  

(4.55)

where \(w_0\) is the original weight, and \(abs(\cdot)\) denotes the absolute value of \(\cdot\). The first line above analytically calculates the weight sensitivity with respect to a parameter \(p\) (see Equation 4.53 and 4.54). The next line evaluates how much is changed in the weight given a small perturbation of \(p\), \(\Delta p\). The last two lines assesses the weight (in its absolute value) elasticity. This computation is not a solely analytic method but a hybrid method as the numerical element \(\Delta p\) has to be determined relative to the value of \(p\) by the user.

### 4.5.3 The Labour-inventory Model

To illustrate a comprehensive eigen-based analysis methodology, we introduce a linear model: a simplified version of the labor-inventory model and is first used in (Saleh et al., 2006). This model demonstrates the instability and oscillation between labor and stock management in the labor supply chain. The stock and flow diagram of the simplified linear model is portrayed in Figure 4.3 and the model equations are presented in Appendix E.

In order to facilitate the parameter analysis, we simplify the parameter names with their initials in Table 4.5. The system equation is set up in Equation 4.56, where the gain matrix \(A\) is arranged in the order of Inventory, Labor, Vacancy and WIPI. This is an inhomogeneous system
Figure 4.3 – Stock and flow diagram of the labor-inventory model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Initials</th>
<th>Value</th>
<th>Parameter</th>
<th>Initials</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>desired_inventory</td>
<td>DI</td>
<td>40000</td>
<td>customer_order_rate</td>
<td>COR</td>
<td>10000</td>
</tr>
<tr>
<td>manufacturing_cycle_time</td>
<td>MCT</td>
<td>8</td>
<td>wip_adj_time</td>
<td>WIPAT</td>
<td>6</td>
</tr>
<tr>
<td>average_duration_of_employment</td>
<td>PRO</td>
<td>100</td>
<td>inv_adj_time</td>
<td>IAT</td>
<td>12</td>
</tr>
<tr>
<td>productivity</td>
<td></td>
<td></td>
<td>standard_work_week</td>
<td>SWW</td>
<td>40</td>
</tr>
<tr>
<td>labor_adj_time</td>
<td>LAT</td>
<td>0.25</td>
<td>vacancy_adj_time</td>
<td>VAT</td>
<td>4</td>
</tr>
<tr>
<td>average_time_to_fill_vacancy</td>
<td>LAT</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.5 – System parameters with initials and values
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

(b \neq 0) whose solution is outlined in Appendix 4.2.3. As we can see in Table 4.6, the eigenvalues indicate the system behaviour is consisted of two types: exponential decay (negative real eigenvalue) and one damped oscillation mode (the conjugate complex eigenvalues).

\[
\dot{x} = Ax + b
\]

\[
= \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 1/ADOE & -1/ADOE & 1/MCT \\
-1/MCT & 0 & 0 & -COR \\
WIPAT*PRO SWW*LAT & -1/MCT & (1 + ATEFV)(1 + MCT) & 0 \\
\end{pmatrix}
\begin{pmatrix}
x \\
\end{pmatrix}

+ \begin{pmatrix}
1/8 \\
-1/100 \\
-7/2280 \\
10 \\
\end{pmatrix}
\begin{pmatrix}
x \\
\end{pmatrix}

\begin{pmatrix}
-10000 \\
0 \\
28000/57 \\
0 \\
\end{pmatrix}
\]

(4.56)

|  |  |  |  |  |
|---|---|---|---|
| \(\lambda_1\) | \(\lambda_2\) | \(\lambda_3\) | \(\lambda_4\) |
| -0.3531 | -0.1380 | -0.0095+0.0988i | -0.0095-0.0988i |

\[
\begin{pmatrix}
r_1 \\
r_2 \\
r_3 \\
r_4 \\
\end{pmatrix}
\begin{pmatrix}
0.3331 \\
0.0215 \\
-0.0589 \\
-0.9408 \\
\end{pmatrix}
\begin{pmatrix}
-0.6713 \\
-0.0010 \\
0.0010 \\
0.7412 \\
\end{pmatrix}
\begin{pmatrix}
0.7832 \\
0.0068 + 0.0066i \\
-0.0052 - 0.0053i \\
-0.0593 + 0.4624i \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
-0.1484 \\
-2.9926 \\
-17.0633 \\
-0.1156 \\
\end{pmatrix}
\begin{pmatrix}
-0.9161 \\
-78.0759 \\
-41.1800 \\
0.4727 \\
\end{pmatrix}
\begin{pmatrix}
0.2773-0.4624i \\
-32.8259+41.3356i \\
-14.0208-10.3466i \\
0.2272+0.4588i \\
\end{pmatrix}
\]

Table 4.6 – Eigenvalues, right eigenvectors and left eigenvectors of the labor-inventory model

To provide a better way to visualize how the eigenvalues navigate the behaviour, simulation results of four state variables are presented in Figure 4.4. In each sub-graph, the blue curve plots the reference behaviour while the black ones depict the individual decomposed behaviour.
Figure 4.4 – State variables behaviour and the decomposed behaviour
mode, \( e^{t\lambda} \). The curves are accommodated in corresponding coordinates by color. As all eigenvalues are negative, all the modes die down in the end. Among all the behaviour modes, the 1st mode decays the fastest with the biggest absolute value \( \lambda_1 = -0.3531 \), while the oscillatory mode decays the most slowly \( R(\lambda_3) = -0.0095 \). The oscillatory period is about \( 2\pi/\text{Im}(\lambda_3) = 63.6 \) weeks, which is captured by individual graph in Figure 4.4. It is worth noting that, owing to the \( A^{-1}b \), the superposition of three behaviour modes is not equivalent to the aggregated state trajectory as there exists a gap.

### 4.5.3.1 Eigenvalue elasticity to parameters

Eigenvalue elasticity allows us to reveal leverage points to alter the behaviour mode. Moreover, it provides a way to intervene in the behaviour in a large scale, i.e., manipulate each individual behaviour mode and modify each one to different degrees. To check our approach, it is reasonable to produce the elasticity also from numerical computation. As is required by the elasticity computation in Equation 4.49, \( \partial A/\partial p \) has to be first solved as follows:

\[
\frac{\partial A}{\partial \text{MCT}} = \begin{pmatrix}
0 & 0 & 0 & \frac{-1}{\text{MCT}^2}
\end{pmatrix}
= \begin{pmatrix}
-\frac{1}{64}
\end{pmatrix}
\]

\[
\frac{\partial A}{\partial \text{ADOE}} = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
\frac{1}{10000}
\end{pmatrix}
\]

\[
\frac{\partial A}{\partial \text{WIPAT}} = \begin{pmatrix}
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
\frac{1}{3420}
\frac{1}{2280}
\end{pmatrix}
\]
\[
\begin{align*}
\frac{\partial A}{\partial IAT} &= \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
(1 + \frac{ATTFV}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}}) & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
7/27360
\end{pmatrix}
\end{align*}
\]

\[
\frac{\partial A}{\partial PRO} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
(1 + \frac{ATTFV}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}}) & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
7/570 & 1/95
\end{pmatrix}
\]

\[
\frac{\partial A}{\partial SWW} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
(1 + \frac{ATTFV}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}}) & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
7/51200 & 1/15200
\end{pmatrix}
\]

\[
\frac{\partial A}{\partial LAT} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
(1 + \frac{ATTFV}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}}) & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
7/43320 & 3/3361 & 1/7220
\end{pmatrix}
\]

\[
\frac{\partial A}{\partial ATTFV} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
-\frac{(1 + \frac{WIPAT}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}}) + \frac{1}{\text{LAT}^2} + \frac{1}{\text{LAT}}}{\text{LAT}^2 + \text{PRO} + \text{SWW} + \text{LAT}} & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
-7/27360 & 1/3 \left( \frac{1}{100} - \frac{1}{19} \right) & 1/64 & -1/4560
\end{pmatrix}
\]
CHAPTER 4. AN ANALYTICAL EIGENVECTOR APPROACH FOR ANALYSING LINEAR SYSTEMS

\[
\frac{\partial A}{\partial V AT} = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
\frac{ATT FV}{VAT^2 \cdot LAT \cdot PRO \cdot SW \cdot LAT} & \frac{ATT FV}{VAT^2} \left( I - \frac{1}{LAT} \right) & \frac{1}{VAT^2} \\
0 & 0 & 0 \\
\frac{ATT FV}{VAT^2 \cdot WIPAT \cdot PRO \cdot SW \cdot LAT} & 0 & 0
\end{pmatrix}
\]

The eigenvalue elasticities are assessed analytically using Equation 4.49 for the real valued eigenvalue, and Equation 4.50 and 4.51 for the conjugated complex eigenvalues. In performing the numerical analysis, we always set \(\Delta p/p = 0.1\), the eigenvalue is reevaluated accordingly. Table 4.7 reports the results from these two methods. Under every eigenvalue, there are two columns, the former one displays the analytical results while the latter one depicts the numerical assessment. The highest elasticity refers to the highest absolute values as the sign only indicates the direction of the change, while the amount of change is suggested by the absolute value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Eigenvalue elasticity w.r.t. parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\varepsilon(\lambda_1))</td>
</tr>
<tr>
<td>MCT</td>
<td>0.0451</td>
</tr>
<tr>
<td>WIPAT</td>
<td>-0.0914</td>
</tr>
<tr>
<td>ADOE</td>
<td>-0.0293</td>
</tr>
<tr>
<td>IAT</td>
<td>0.0494</td>
</tr>
<tr>
<td>PRO</td>
<td>0</td>
</tr>
<tr>
<td>SWW</td>
<td>0</td>
</tr>
<tr>
<td>LAT</td>
<td>0.0935</td>
</tr>
<tr>
<td>VAT</td>
<td>-0.6699</td>
</tr>
<tr>
<td>ATTFV</td>
<td>-0.3974</td>
</tr>
</tbody>
</table>

Table 4.7 – Analytical and numerical results of eigenvalue elasticity to parameters

Table 4.7 suggests these two methods generate consistent outcomes. This reflects the validation of our proposed analytic solution to the weight elasticity analysis. Moreover, the numerical results will be even closer to the analytical ones if a much smaller perturbation is given around the parameter value, for example, \(\Delta p/p = 0.01\). We can see that the parameters VAT and
**MCT** play the most significant role in the first and second behaviour modes respectively. On the opposite, **ADOE** exerts the least influence on either the them regardless of it sign. One criterion to stabilization is to minimize the real part of the eigenvalue (Forrester, 1982) to achieve a faster convergence. Since the first behaviour mode converges faster than the second mode, the second eigenvalue is aimed to be decreased. Reducing $\lambda_2$ can accelerate the damping and brings the behaviour to a stable state sooner. Table 4.7 indicates **MCT** and **IAT** would be the good points to intervene to increase the stability of the system: raising them can plummet $\lambda_2$ while only a slight growth will appear in $\lambda_1$.

In the event of the complex conjugated eigenvalues, there are also other criteria by which to judge the stability. For example, the imaginary part determines the damping frequency; the damping ratio, i.e., the ratio of the real part to the imaginary part, determines the damping ratio of continuous oscillations. Both the damping frequency and damping ratio can be used as indicators to test the stabilization policies. However, we did not compute the damping ratio elasticity here. **IAT** has the highest elasticity in terms of the real part of the complex eigenvalues. The real part affects the decaying speed. The more negative it is, the more significantly decaying the trajectory is. In regard to the imaginary part, **LAT** and **ATT FV** come first to affect the damped frequency, which manifests themselves to be the leverage points to prolong the oscillation period to stabilize the system.

Finally, it is interesting to note there is one instance that the analytical result contradicts the numerical one: $\lambda_{3,4}^{re}$ with respect to **WIPAT**. Further inspection leads us to perform more numerical tests by changing the value of $\Delta p/p$. Figure 4.5 plots the relationship between the perturbance of **WIPAT** and the real part of eigenvalue. We can see that, as **WIPAT** increases (decreases) at value 6 a little, the eigenvalue increases (decreases) in its absolute value. This indicates the analytical elasticity result should be positive. However, the numerical elasticity result is calculated by increasing 10% of the value, which is too large so that the eigenvalue decreases. This explains why the analytical and numerical results of third eigenvalue elasticity with respect to **WIPAT** contradict in Table 4.7. This investigation reveals the nonlinear relationship between the parameter and the eigenvalue. Moreover, it remind us that the analytical elasticity results could be wrong when the disturbance to the parameter is not small enough. Therefore, we should always be cautious when we make use of the elasticity results in Table 4.7.
4.5.3.2 Weight elasticity to parameters

Besides the eigenvalue elasticity, it is important to consider the weight elasticity as well. As we have discussed in Section 4.2.1, the state behaviour in an inhomogeneous system is slightly different from that in linear systems. It is decomposed into the following format (obtained from Equation 4.12):

\[
x_i = e^{t\lambda_1} r_{i1} H (x(0) + A^{-1}b) + \cdots + e^{t\lambda_4} r_{i4} H (x(0) + A^{-1}b) - (A^{-1}b)_i
\]

\[
x_i = e^{t\lambda_1} w_{i1} + \cdots + e^{t\lambda_4} w_{i4} - (A^{-1}b)_i
\]

We can see the weight is no longer the same as that in the linear system. Consequently, the weight elasticity changes. Similar as the real-valued weight elasticity expression in Equation 4.53, the real-valued weight sensitivity in an inhomogeneous system is expressed as:

\[
\varepsilon_{w_{ji}}^R = \frac{\partial r_{ji} H (x(0) + A^{-1}b)}{\partial p} \frac{p}{w_{ji}}
\]

\[
= \left( \frac{\partial r_{ji} H (x(0) + A^{-1}b)}{\partial p} \right) \left( A^{-1} \frac{\partial b}{\partial p} - A^{-1} \frac{\partial A}{\partial p} A^{-1} b \right) \right) \frac{p}{w_{ji}}
\]

\[
= S_{w_{ji}}^R
\]
The first derivative term in the inner box is the eigenvector-related component sensitivity to the parameter $p$, and it can be solved by Equation 4.54. The weight of each decomposed behaviour mode is outlined in Table 4.8. The weight associated with the oscillatory behaviour is always shown as a positive value which corresponds to the amplitude. An equivalent negative weight can be achieved through shifting the phase by $\pi$ in $\sin(\ )$ representation. This table also shows the $2^{nd}$ and $3^{rd}$ mode weights of *Inventory* and *WILI* are much greater than others, we hence will place the focus on the weight analysis of these specific modes.

$$x = w_1 e^{\lambda_1 t} + w_2 e^{\lambda_2 t} + w_3 e^{Re(\lambda_3) t} \sin(\theta - bt)$$

<table>
<thead>
<tr>
<th>Variable</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory</td>
<td>-122.22</td>
<td>14432</td>
<td>7384</td>
</tr>
<tr>
<td>Labor</td>
<td>-7.87</td>
<td>20.72</td>
<td>89.09</td>
</tr>
<tr>
<td>Vacancy</td>
<td>21.61</td>
<td>-21.22</td>
<td>70.40</td>
</tr>
<tr>
<td>WIPI</td>
<td>345.24</td>
<td>-15934</td>
<td>5861</td>
</tr>
</tbody>
</table>

Table 4.8 – Behaviour mode weights for four state variables

In addition, we list the computation of the constant vector with respect to each parameter ($\partial b / \partial p$) in Equation 4.58 step by step for the convenience of examining the eigenvector-related elasticity analysis.
\[ \frac{\partial b}{\partial \text{ATT}FV} = \begin{pmatrix} 0 \\ 0 \\ \frac{(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW})}{VAT} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 7000 \\ 171 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{VAT}} = \begin{pmatrix} 0 \\ 0 \\ -ATTFV(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -14000 \\ 171 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{MCT}} = \begin{pmatrix} 0 \\ 0 \\ -(1 + ATTTFV)(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2000 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{WIPAT}} = \begin{pmatrix} 0 \\ 0 \\ -(1 + ATTTFV)(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{IAT}} = \begin{pmatrix} 0 \\ 0 \\ -(1 + ATTTFV)(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{LAT}} = \begin{pmatrix} 0 \\ 0 \\ -(1 + ATTTFV)(1 + MCT + DI + \text{LAT} + \text{PRO} + \text{SWW}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{PRO}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{SWW}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \]

\[ \frac{\partial b}{\partial \text{ADOE}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \]

(4.58)
<table>
<thead>
<tr>
<th></th>
<th>Inventory</th>
<th>Labor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( w_1 )</td>
<td>( w_2 )</td>
</tr>
<tr>
<td>IAT</td>
<td>-8.762</td>
<td>0.424</td>
</tr>
<tr>
<td>WIPAT</td>
<td>6.298</td>
<td>0.019</td>
</tr>
<tr>
<td>MCT</td>
<td>-23.697</td>
<td>1.599</td>
</tr>
<tr>
<td>SWW</td>
<td>15.13</td>
<td>3.919</td>
</tr>
<tr>
<td>PRO</td>
<td>15.13</td>
<td>3.919</td>
</tr>
<tr>
<td>VAT</td>
<td>2.992</td>
<td>0.062</td>
</tr>
<tr>
<td>LAT</td>
<td>2.009</td>
<td>0.342</td>
</tr>
<tr>
<td>ATTFV</td>
<td>-4.565</td>
<td>-0.701</td>
</tr>
<tr>
<td>ADOE</td>
<td>3.786</td>
<td>0.147</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Vacancy</th>
<th>WIPI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( w_1 )</td>
<td>( w_2 )</td>
</tr>
<tr>
<td>IAT</td>
<td>-8.585</td>
<td>-7.387</td>
</tr>
<tr>
<td>WIPAT</td>
<td>5.971</td>
<td>2.946</td>
</tr>
<tr>
<td>MCT</td>
<td>-20.988</td>
<td>4.954</td>
</tr>
<tr>
<td>SWW</td>
<td>14.13</td>
<td>2.919</td>
</tr>
<tr>
<td>PRO</td>
<td>14.13</td>
<td>2.919</td>
</tr>
<tr>
<td>VAT</td>
<td>0.596</td>
<td>0.406</td>
</tr>
<tr>
<td>LAT</td>
<td>2.344</td>
<td>0.653</td>
</tr>
<tr>
<td>ATTFV</td>
<td>-5.987</td>
<td>0.357</td>
</tr>
<tr>
<td>ADOE</td>
<td>3.710</td>
<td>0.265</td>
</tr>
</tbody>
</table>

Table 4.9 – Analytical and numerical results of weight elasticity to parameters associated with 1st and 2nd mode. The leverage points with highest elasticity (regardless of the sign) are marked with a box.

Table 4.9 shows the weight elasticity associated with the exponential behaviour modes. The pure numerical method is utilized to verify the outcomes as before. MCT (manufacturing cycle time) is the highest leverage point in terms of \( w_1 \) for all variables. Considering \( w_1 \), either IAT or SWW/PRO has the greatest value of elasticity among these four state variables. It is interesting to observe the elasticity to MCT is always negative in \( w_1 \) but positive in \( w_2 \). Consequently, a reduction in MCT results in an “increase” in \( w_1 \) and a “decrease” in \( w_2 \). “ ” means the change is an increase if it is in the same direction of the weight, and an decrease if it is in the opposite direction. Through comparisons in Figure 4.4, we know the first behaviour mode decays much faster than the second one. Find a leverage point to “shrink” \( w_2 \) is a good strategy to reduce its slow convergence to the steady state in the aggregated trajectory. For example, SWW/PRO and MCT are most significant parameters that affect \( w_2 \) among all state variables. Therefore, they are likely to be the effective points to intervene the second behaviour mode.
Regarding the weight elasticity associated with the oscillatory behaviour mode, its analytical computation has to follow the definition in Equation 4.55. The weight sensitivity \( \frac{\partial w_{ji}}{\partial p} \) is first solved using Equation 4.54, the rest is to measure the change in the absolute value of the weight according to the change in the parameter. The term \( \Delta p/p \) is set to be 0.1 in both analytical and numerical methods. While in the numerical assessment, the weight has to be reevaluated to compare the absolute value between the reference weight and the new one.

Table 4.10 presents the weight elasticity in four state variables in the model. The left part of each column shows the results from the analytical method while the right part displays the numerical results.

<table>
<thead>
<tr>
<th></th>
<th>Inventory</th>
<th>Labor</th>
<th>Vacancy</th>
<th>WIPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAT</td>
<td>0.988</td>
<td>1.020</td>
<td>0.521</td>
<td>0.277</td>
</tr>
<tr>
<td>WIPAT</td>
<td>-0.063</td>
<td>-0.060</td>
<td>-0.569</td>
<td>-0.829</td>
</tr>
<tr>
<td>SWW</td>
<td>3.283</td>
<td>3.283</td>
<td>2.789</td>
<td>2.075</td>
</tr>
<tr>
<td>PRO</td>
<td>3.283</td>
<td>3.283</td>
<td>2.789</td>
<td>2.075</td>
</tr>
<tr>
<td>VAT</td>
<td>-0.076</td>
<td>-0.072</td>
<td>-0.048</td>
<td>-0.046</td>
</tr>
<tr>
<td>LAT</td>
<td>0.233</td>
<td>0.203</td>
<td>-0.446</td>
<td>-0.417</td>
</tr>
<tr>
<td>ATTFV</td>
<td>0.848</td>
<td>0.722</td>
<td>1.528</td>
<td>1.456</td>
</tr>
<tr>
<td>ADOE</td>
<td>-0.008</td>
<td>-0.008</td>
<td>0.010</td>
<td>0.010</td>
</tr>
</tbody>
</table>

Table 4.10 – Analytical and numerical results of weight elasticity to parameters associated with 3rd oscillatory behaviour mode. The leverage points with highest elasticity (regardless of the sign) are marked with a box.

This table also suggests the results from different sources are consistent with each other. Negative elasticity indicates a 10% “increment” to the parameter will cast an “decrease” on the weight, which in turn reduces the oscillatory amplitude. On the contrary, positive elasticity suggests the weight will “increase” given the parameter “increases”. Weight elasticity provides a novel perspective to intervene on individual state variable. From this example, we also find that for some particular parameters (e.g., SWW, PRO), it is possible to separate the eigenvalue and eigenvector in the model analysis.

The oscillation can be relieved by reducing its amplitude, while the absolute value of the weight determines the amplitude. The fluctuations can be eased if we reduce \( |w_3| \). Observing from Table 4.9 and Table 4.10 in terms of Inventory and WIPI, we find that MCT has the
highest leverage influence in the $3^{rd}$ behaviour mode, while SWW and PRO come second. Further scrutiny on Table 4.7 finds PRO and SWW have no impact on eigenvalues while MCT has. Moreover, these two parameters have identical impact to the system, we do not distinguish them in the following analysis.

Summarizing the above analysis, we suggest two possible policies to reduce the oscillation:

1. Decrease the manufacturing cycle time (MCT). This would simultaneously reduce the $2^{nd}$ mode weight as the weight elasticity is positive among all state variables.
2. Reduce the standard work week / productivity (SWW/PRO). This policy seems to be quite counter-intuitive, and we will find more details in the following investigation.

### 4.5.3.3 Examination on the policy recommendations

From the above analysis, we know that MCT is the leverage point to effectively reduce the amplitude of the oscillatory behaviour in the system. As the elasticity is positive, we reduce the parameter by 10% from its original value 8 to 7.2. Figure 4.6 shows the new behaviour in contrast with the reference behaviour. It is evidently observable that the amplitude of the oscillation is significantly diminished throughout all four state variables. The oscillations are much smooth but the damping frequency (i.e., the observed frequency in the decaying cycle) is increased a little bit. This is also verified by the negative eigenvalue (in terms of its imaginary part) elasticity. The reduction of the parameter raises the imaginary part of the eigenvalue, and in turn lessens the oscillatory period $2\pi/Im(\lambda)$. Overall, this is considered as an effective policy. In addition, we can analyze this policy by simple reasoning as follows. Reducing the MCT has a direct impact to boost the production rate in a short term, which catches up with the production start rate more quickly. Therefore, the work in process inventory (WIP I) has less fluctuations. Meanwhile, decreasing the MCT reduces the desired work in process inventory (Desired wip), which further cut the desired labour. As a result, the vacancy and the actual labour are diminished. The production start rate will be lessened as the shrink in labour. Because the production rate is always trying to catch up with the production start rate, its final value is also reduced. As a result, the inventory, which is an accumulation between the production rate and the shipment rate, therefore will experience smaller oscillations.
Figure 4.6 – Comparisons of state trajectories on the change of $MCT$
The second policy is conducted by dropping $SWW$ by 10% to 36. Its effect across four state variables is examined, and is reported in Figure 4.7. However, instead of a relief to the oscillation, all trajectories fluctuate even more intensively in the system. To reassure the result, we assess the weight with the decreased $SWW$. Table 4.11 confirms the phenomenon by showing $w_3$ actually increases in each state. This contradicts our anticipations as we think a positive elasticity suggests a change in the same direction of the dependent variable in response to the independent variable. However, it seems that either an increase or a decrease the parameter (by 10%) results in an expansion in the oscillatory mode weights. This demonstrates an identical case that we find in the EEA analysis (the complex eigenvalue elasticity to $WIPAT$ in Table 4.7). Like eigenvalue, eigenvector has a nonlinear relationship with the parameters, the same is true for the weight (the weight is comprised of the eigenvector-related components). As a result, the weight alternated by a perturbance (not small enough) in the parameter may not achieve our expectation.

<table>
<thead>
<tr>
<th>Variable</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory</td>
<td>-122.22</td>
<td>62.704</td>
<td>14432</td>
</tr>
<tr>
<td>Labor</td>
<td>-7.87</td>
<td>4.49</td>
<td>20.72</td>
</tr>
<tr>
<td>WIPI</td>
<td>345.24</td>
<td>-177.11</td>
<td>-15934</td>
</tr>
</tbody>
</table>

Table 4.11 – Behaviour mode weights for four state variables after modifying $SWW$. The left column under each weight shows the reference weight while the right column shows the modified weight.

Finally, after probing the two policy generated from the eigenvector analysis, recall that $IAT$ is the most influential parameter in affecting the real part of the conjugated eigenvalue in the EEA. We, therefore, add another experiment to test the efficiency of this policy. An amount of 10% is added to its reference value to run the simulation. Because its elasticity is positive with a negative real part of the eigenvalue, enlarging the parameter will increase the convergence. Figure 4.8 captures the modified system behaviours. It shows damping occurs across over all four state variables, however, the diminish of the amplitude is not obviously observable. The effect of adopting this policy is hardly satisfactory compared with the change on $MCT$. This can be attributed to the small real part value of the conjugated eigenvalues, i.e., $Re(\lambda_{3,4}) = -0.0095$, the change in even the highest leverage parameter does not incur much alternation in the oscillatory behaviour. Therefore, one needs a large value to change and a large elasticity.
Figure 4.7 – Comparisons of state trajectories on the change of $SWW$
Figure 4.8 – Comparisons of state trajectories on the change of $IAT$
in that value to change parameter. This further shows that the elasticity can be weighted by the current weights/eigenvalues to evaluate how effectively a value modification in a parameter may cause a change among them. The weighting of elasticity is particularly worthy when the values of weights/eigenvalues are close to each other. One similar idea is discussed in (Güneralp, 2006b).

The above experiments shows reducing $MCT$ is the best policy to significantly improve the system behaviour in all aspects. Moreover, our results show focusing solely on the EEA may be insufficient to reveal the most effective leverage points. It is thus advisable to utilize the weight/eigenvector-related component analysis which would shed an alternative light on the structure analysis, and even generate more efficient policy recommendations.

### 4.6 Summary and Discussion

In this chapter, we conduct a series of studies concentrating on the analytic eigenvector analysis in terms of various structure components. While the EEA reveals the dominant structure that influences the divergence or convergence rate of the behaviour mode, the eigenvector-related component analysis identifies which part of the structure that impacts the weight of the behaviour mode most. One strength of the eigenvector analysis is it is directly related to a specific variable (eigenvectors vary for each state variable) rather than the system level as the EEA (eigenvalues remain the same for all the state variables). An alternative representation of the state trajectory is proposed where the left eigenvector is introduced. As a result, the eigenvector-related sensitivity is part of the comprehensive state sensitivity (see Equation 4.14). An important conclusion derived from our eigenvector analysis is: **the right and left eigenvector sensitivities have to be assessed together and simultaneously as the eigenpair are dependent of one another.** Furthermore, analytic formulas to compute the eigenvector-related (eigenpair) sensitivity with respect to various structure components, such as compact links, pathways, causal links, and parameters, are developed throughout this chapter. Finally, a case study of the labour-inventory model is presented to examine the eigen-based analysis in terms of system parameters. A number of numerical tests are conducted to verify the analytical results in parallel. In this model, both EEA and weight analysis are conducted to examine their efficiency in leverage point identification and the weight analysis even performs better than the
In summary, the eigenvector analysis can be utilized to identify high leverage points from a new perspective. Our analysis shows a complete eigen-based analysis produces more insightful implications than the EEA alone. In addition, the analytical eigenvector analysis method is capable of saving time from repetitive simulations in numerical analysis and enables a much more efficient search for high leverage structures.

One drawback is that the eigen-based analysis is not directly applicable to the nonlinear models. The effectiveness and robustness of the analytic eigenvector analysis method to the nonlinear models is left for further tests and on-going developments. Another issue associated with the eigenvector-related component analysis is that perturbation to the substructure will affect both the eigenvalues and eigenvectors. It is more complicated than solely working with the EEA as the selected leverage points have to satisfy both criteria, and the leverage points may have conflicting effects in optimizing the system behaviour. We do suggest one should examine the structure elasticity values in both the eigenvalue and eigenvector analysis before making policy decisions. Despite of these limitations, we are optimistic with the eigenvector analysis which can assist one to see an overall picture and make comprehensive policy strategies.
Chapter 5

An Extension of Loop Deactivation in the Behavioural Method

The aim of this chapter is to refine a computational dominant loop analysis method, the behavioural method, relating to its deactivating process. As is reviewed in an earlier chapter 2, this simulation based formal analysis method is quite different from the analytical methods such as EEA and the eigenvector analysis in previous chapter. The behavioural method is in line with the classic methods that rely on hypothesis testing as individual candidate feedback loop will be deactivated in order to test their respective effect on the behaviour of interest. Dominant loops are identified by checking whether the atomic behaviour pattern (see Section 2.2.1.3 for reference) differs from the reference ABP. A crucial step in the behavioural method is to deactivate the candidate loop. Ford proposed to deactivate a feedback loop by freezing the value of its control variable within this particular feedback loop. By doing this, the feedback loop can be thought of as inactive and lose its impact on the system. Nevertheless, one limitation of this proposition is there is no guarantee that every loop will have a control variable to be manipulated to deactivate the loop.

Recently, Phaff revisited the behavioural method and proposed possible improvements from two aspects (Phaff, 2008):

- Instead of a control variable, a unique edge, which is an edge that uniquely belongs to a loop, is selected as an intervention point to deactivate the feedback loop. It is a looser constraint since the existence of a control variable indicates the existence of two unique
edges whereas a unique edge can exist without having a control variable;

- The shortest independent loop set (Oliva, 2004) should be adopted as the candidate loop set. In the original version, there is no specific loop selection method suggested, it is, therefore, necessary to specify a loop selection method to unify the analysis, and further to facilitate the automation.

However, these improvements do not completely solve the loop deactivation problem. Because there are chances that the loops in the SILS have no unique edges. The work of this chapter studies this particular circumstance when the candidate feedback loop has neither a control variable nor a unique edge, and proposes a new loop deactivation method by modifying the unique two consecutive edges which distinguish this loop from others. If not a single unique edge can be found in a candidate feedback loop, a pair of consecutive edges that uniquely belongs to one loop is utilized to undertake the deactivation task.

This chapter is organized as follows: First a brief review of the behavioural method is presented together with the illustration of the yeast model. We then clarify the scenario when the loop deactivation by the unique edge fails. It will be demonstrated by a hypothetic example. Subsequently, a three-step approach of deactivating a feedback loop by manipulating a unique pair of consecutive edges is developed and illustrated with the same hypothetical example. In order to verify this proposed method, we further apply it to the Long Wave model (Sterman, 1985) and compare the analysis result with other dominant loop analysis methods, including the eigenvalue elasticity analysis and the original version of the behavioural method. Finally this chapter ends up with conclusions and recommendations for further research. The research outlined in this chapter has been disseminated in (Huang et al., 2010b).

5.1 A Brief Revisit to the Behavioural Method

Before we introduce the new loop deactivation method, a brief revision to the behavioural method is first presented using yeast model (see Table 5.1 for its equations). The yeast model is a second order non-linear model and a classical example of overshoot-and-collapse dynamics. There are four feedback loops marked by the open circles with arrow and Cells is selected as the variable of interest (see Figure 5.1). For L1 and L2, there is one unique edge each, i.e.,
timestep=0.01 ; simulation length=90
Stocks:
Cells=INTEG(births-deaths, 1)
Alcohol=INTEG(alcohol generation, 0)
Flows:
births=Cells/division time* effect of alcohol on birth
deaths=Cells/life time* effect of alcohol on death
alcoholgeneration=Cells*alcohol per cell generation
Auxiliaries:
effectofalcoholonbirths=(-0.1*Alcohol)+1.1
effectofalcoholondeaths=EXP(Alcohol-11);
life time=30 ; division time=15
alcohol per cell generation=0.01

Table 5.1 – Equations for the yeast model

In its original version, the reference behaviour of Cells is first partitioned into four phased based on the ABP in Figure 5.1(b). Within each phase, the candidate loop is to be deactivated to identify the dominant feedback loop. The control variable and unique edge for each individual loop are outlined in Table 5.2. Furthermore, the specific modification to the equation is depicted to enable the loop deactivation. Four simulations have to be performed after the deactivation operation is done in each phase to assess the ABPs of the modified behaviour. If the

(a) The stock and flow diagram of the yeast model
(b) Behaviour of Cells

Figure 5.1 – The yeast model
CHAPTER 5. AN EXTENSION OF LOOP DEACTIVATION IN THE BEHAVIOURAL METHOD

<table>
<thead>
<tr>
<th>Control variable/unique edge</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cells→births</td>
<td>eab</td>
<td>ead</td>
<td>ead</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Specific modification</th>
<th>births=</th>
<th>deaths=</th>
<th>births=</th>
<th>deaths=</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c/division time*eab</td>
<td>c/life time*ead</td>
<td>Cells/division time*c</td>
<td>Cells/life time*c</td>
</tr>
</tbody>
</table>

Table 5.2 – Information for the loop deactivation in the yeast model. \( c \) refers to the constant value of the control variable or the starting point of the unique edge at the starting time of a given phase.

A new ABP of \( Cells \) is different from the reference one, the loop deactivated is considered as a dominant loop. This process is then repeated in the other three phases to identify all dominant feedback loops throughout the entire simulation period.

This example sets the scene for the work that we are to present in regard to the loop deactivation procedure. The behavioural method is applicable to the system where the individual feedback loop has at least one unique edge. In the following sections, we are to examine the scenario that no unique edges exists in a candidate loop and find a solution for this.

## 5.2 A New Loop Deactivation Method

In our previous discussion of the behavioural method, we know the SILS is adopted as a method to select the candidate loops for analysis. However, there is one problem when attempting to deactivate a loop, the independent loop set may contain loops that have no unique edges. The definition of ILS states: “An independent loop set of a digraph is a maximal set of loops whose incidence vectors are linearly independent, i.e., every other loop is linearly dependent upon the loops in it” (Kampmann, 1996). In the procedure of constructing the SILS, every newly added loop has the least number of new edges which have not been included in the loop set. When adding a new loop to the set, there are chances that the existing unique edges in the previously introduced independent loops are turned to be non-unique now.

In order to clarify under what circumstance this problem occurs, we use a hypothetical example which is shown in Figure 5.2. It is known that in a strongly connected graph the relationship between the total number of independent loops \( u \), nodes \( n \) and edges \( e \) are:

\[
u = e - n + 1
\]
As this system is strongly connected, there are $8 - 6 + 1 = 3$ independent loops. Table 5.3 lists the loop matrix describing the relationship between the loops and their edges. We observe that the incidence vectors of each loop (in rows) are linearly independent whereas $L_2$ does not have a unique edge. This is a simple scenario which shows the incapability of both Ford’s behavioural method and its improved version. We can imagine a complex system will contain more loops that have no unique edge. Therefore, it is necessary to find a way to deactivate the loop under such circumstances. In the following section, we will introduce a method to overcome this problem and use the same example to demonstrate it step by step.

![Figure 5.2 – A hypothetical model](image)

Table 5.3 – The loop matrix for the hypothetical example

<table>
<thead>
<tr>
<th></th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$e_3$</th>
<th>$e_4$</th>
<th>$e_5$</th>
<th>$e_6$</th>
<th>$e_7$</th>
<th>$e_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_1$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$L_2$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$L_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### 5.2.1 Deactivating a Loop by Its Unique Consecutive Two Edges

The idea of deactivating a loop with no unique edge is an extension of the work from (Phaff, 2008). We consider using the unique consecutive two edges to deactivate the candidate feedback loop. The unique consecutive two edges refer to two consecutive edges which belong to a loop exclusively, and no any other loops in SILS contain them. We present the procedure of how to deactivate a loop with no unique edge in the behavioural method as follows. Three
steps of deactivating a loop with no unique edge are described as follows:

1. Identify all the feedback loops using the SILS algorithm, and let them be our candidate loops.

2. Filter out the loops that do not have a unique edge and identify a pair of unique consecutive two edges which do not lie in any other loops in each selected loops. In order to deactivate these loops, we follow the steps below to modify the model:

   (a) Create two nodes which serve as the copies of the two variables of the first edge in pair (i.e., the starting and middle nodes in the edge pair). Link these two new nodes with the same order as their counterparts in the reference model.

   (b) Drop the second edge in the unique consecutive edges, and then add a link from the copy of the middle variable to the ending variable of the unique edge pair.

   (c) Copy all the edges who end with the middle variable in the unique consecutive edges (except the edge which is one of the unique two edges) to the variable of its copy.

3. Set the new variable who is the copy of the origin of the unique consecutive edges to be a fixed value. The choice of this fixed value is consistent with the Ford’s behavioural method, i.e., the value at its deactivation.

To clarify the methodology stated above, we use the model in Figure 5.2 to demonstrate the approach.

1. The loops in SILS are depicted in Table 5.3.

2. Table 5.3 shows that $L_2$ shares all its constituent edges with the other two loops and does not have a unique edge. Besides, only the combination of $e_3$ and $e_4$, the unique edge pair, can determine $L_2$. Other combinations, e.g., $(e_2, e_3) \in \{L_1, L_2\}$, $(e_4, e_2) \in \{L_2, L_3\}$, do not exclusively belong to one loop. Therefore, the unique consecutive two edges are identified as $e_3, e_4$ or denoted by nodes in sequence: $b, c, a$.

   (a) Create two nodes $b', c'$ as the copies of $b, c$. We link these two new nodes $b' \rightarrow c'$ in the same direction in the original model $b \rightarrow c$. This is illustrated in Figure 5.3.
Drop the second edge in the unique consecutive edges, $e_4$, and then add a link from the copy of the middle variable to the ending variable of the unique edge pair, i.e., add a link from $c'$ to $a$. Figure 5.4 shows that $e_4$ is removed and replaced by a dash line $e_{c'a}$.

Copy all the edges where the variable $c$ is a tail to $c'$ except $e_{bc}$. There is only one edge satisfying this condition: $f \rightarrow c$. In this case, the variable $c'$ is initialized. It is constructed in the same way as its counterpart $c$, except for one edge, $e_3$. The final modified model for deactivating $L_2$ is depicted in Figure 5.5.

3. Set node $b'$ to be a constant at the beginning of each phase. Subsequently, the behavioural method can be performed to analyze the influence of this particular loop to the behaviour of the variable of interest.
Compare the model after deactivating (Figure 5.5) with the reference model (Figure 5.2), we can see the removal of $e4$ impacts $L2$ and $L3$ simultaneously. What we do to recover $L3$ and maintain $L2$ deactivated is as follows:

1. We identify that $c$ is a joint point of $L2$ and $L3$. It is a function of two “inputs”, say, $c = G(b, f)$, nodes $b$ and $f$ lie in $L2$ and $L3$ respectively. In order not to impact $L3$, we retain all the inputs to $c$. However, we have to deactivate $L2$, so a node $b'$ is created to be a constant value of $b$. Another variable $c'$ which is a copy of $c$ is created for the purpose of propagating the information flow passed through the constant $b'$ when deactivating $L2$.

2. $c'$ is connected to $a$ while $e4$ is deleted. Therefore $L2$ is deactivated and its modified version $L2'$ is: $b' \rightarrow c' \rightarrow a$. At the same time, $L3$ is cut by removing $e4$.

3. In order to recover $L3$, we add the edge $f \rightarrow c'$ which makes $c' = G(b', f)$. In comparison with the reference model where $c = G(b, f)$, we find $c'$ takes the role of $c$ and flows back to $a$ with $L2$ deactivated but $f$ in $L3$ unchanged. A new loop is formed: $a \rightarrow b \rightarrow f \rightarrow c' \rightarrow a$. This is $L3'$ which we consider as an equivalent of $L3$ in the modified model after deactivating $L2$.

In summary, Figure 5.5 shows the final modified model with the new loop deactivation method. Compared with the reference model, $L1$ remains the same. Though $L3$ becomes a new loop $L3'$, it is still regarded as “not deactivated” judging by the definition of the original way of deactivating a loop (Ford, 1999), whereas $L2$ is indeed deactivated by setting $b'$ to be a constant. Therefore, we believe deactivating the unique consecutive two edges is a reasonable and
reliable approach to deactivate a loop.

### 5.3 A Case Study of the Long Wave Model

To demonstrate the proposed methodology with a more complex model, we test our approach on the simple long-wave model. This is a nonlinear economic model developed by Sterman (1985) to explain the long term economic cycles caused by capital self-ordering in the simplest possible terms. The model has three state variables (Capital, Supply and Backlog), yet it is highly interconnected and contains 16 independent feedback loops. Its stock and flow diagram is shown in Figure 5.6. Appendix F contains the model equations.

![Figure 5.6 – Stock and flow diagram – the simple long wave model](image)

#### 5.3.1 Why This Model?

The Simple Long Wave model is chosen because: (1) Both Kampmann (1996) and Ford (1999) had studied this model and reached the identical general conclusion with regard to which loops dominate the behaviour of variable *Desired Production*. We can hence take their result as a
benchmark to test the validity of our loop deactivation approach; (2) Though the behavioural method was applied to the long wave model by Ford, it was not a complete analysis. As Ford only selected the dominant loops identified by EEA to test and verify the behavioural method. A problem with the loop analysis on the subset of the complete loop set is this would affect the validity of the choice of the control variable. As the chance of a variable becoming a control variable diminishes when the number of loop grows. (3) Under the assumption that the SILS is selected as the complete candidate loop set, some loops including the dominant loops (identified by the behavioural method from Ford and EEA from Kampmann) do not have a unique edge, thus we cannot use either Ford’s behavioural method or Phaff’s analysis. Therefore, it is a good opportunity to test the new loop deactivation approach which is developed to enable the behavioural method under such a particular circumstance.

5.3.2 The Loop Deactivation Method

Based on the behavioural method, the first step is to select the variable of interest. In order the compare our result with EEA, we cannot randomly pick a variable of interest. It is known that EEA is variable-independent, i.e., the dominant loops are the same for any state variables in the model. However, the behavioural method is variable-dependent. Different variables of interest give rise to different phase partitions. In terms of the similarities of the behaviour pattern in each phase and the durations of the corresponding phases (indicated by Table 5.6), we find those in the variable of Desired Production coincide with EEA (Kampmann, 1996) more than any other variables. For the same reason, Ford also chose Desired Production as the variable of interest (Ford, 1999). Desired Production is therefore chosen as the variable of interest in our analysis to compare the results with EEA and the original behavioural method.

The next step is to set up the candidate loop set and identify the unique edges or the unique pair of consecutive two edges. We use SILS algorithm to generate the candidate loop set (see Table 5.4 and Table 5.5). Followed by this, the unique edge for each loop is to be identified. Within the SILS of the simple long wave model, we find that four loops do not have unique edges. Table 5.4 and Table 5.5 list these two loop categories with their unique edges and the unique pair of the consecutive two edges respectively.

We partition the time intervals (phases) in the third step. The behaviour of Desired Production
<table>
<thead>
<tr>
<th>Loop no.</th>
<th>Variable sequence (Loop name)</th>
<th>Unique edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Acquisitions, S</td>
<td>Acquisition, S</td>
</tr>
<tr>
<td></td>
<td>(Supply line 1st-order control)</td>
<td></td>
</tr>
<tr>
<td>L2</td>
<td>Depreciation, C</td>
<td>Depreciation, C</td>
</tr>
<tr>
<td></td>
<td>(Capital decay)</td>
<td></td>
</tr>
<tr>
<td>L4</td>
<td>RelativeOrders, CapitalOrders, S, Acquisitions, C, Depreciation</td>
<td>Depreciation, RelativeOrders</td>
</tr>
<tr>
<td></td>
<td>(Steady state capital)</td>
<td></td>
</tr>
<tr>
<td>L5</td>
<td>DesiredOrders, RelativeOrders, CapitalOrders, S, Acquisitions, C, Depreciation</td>
<td>Depreciation, DesiredOrders</td>
</tr>
<tr>
<td></td>
<td>(Capital replenishment)</td>
<td></td>
</tr>
<tr>
<td>L6</td>
<td>RelativeOrders, CapitalOrders, S, SupplyAdjustment, DesiredOrders</td>
<td>S, SupplyAdjustment</td>
</tr>
<tr>
<td></td>
<td>(Supply line adjustment)</td>
<td></td>
</tr>
<tr>
<td>L7</td>
<td>RelativeOrders, CapitalOrders, S, Acquisitions, C, CapitalAdjustment, DesiredOrders</td>
<td>C, CapitalAdjustment</td>
</tr>
<tr>
<td></td>
<td>(Capital adjustment)</td>
<td></td>
</tr>
<tr>
<td>L8</td>
<td>Acquisitions, Capital, Depreciation, DesiredSupplyLine, SupplyAdjustment, DesiredOrders, Relativeorders, CapitalOrders, S</td>
<td>Depreciation, DesiredSupplyLine</td>
</tr>
<tr>
<td></td>
<td>(Steady-state Supply line)</td>
<td></td>
</tr>
<tr>
<td>L10</td>
<td>DesiredOrders, RelativeOrders, CapitalOrders, S, Acquisitions, C, Capacity, Production, DesiredSupplyLine, SupplyAdjustment</td>
<td>Production, DesiredSupplyLine</td>
</tr>
<tr>
<td></td>
<td>(Supply line adjustment via Production)</td>
<td></td>
</tr>
<tr>
<td>L12</td>
<td>Acquisitions, C, Capacity, CapacityUtilization, Production</td>
<td>Capacity, CapacityUtilization</td>
</tr>
<tr>
<td></td>
<td>(Demand balancing)</td>
<td></td>
</tr>
<tr>
<td>L13</td>
<td>DesiredProduction, CapacityUtilization, Production, B</td>
<td>DesiredProduction, CapacityUtilization</td>
</tr>
<tr>
<td></td>
<td>(Production scheduling)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Hoarding)</td>
<td></td>
</tr>
<tr>
<td>L16</td>
<td>DesiredProduction, DesiredCapital, CapitalAdjustment, DesiredOrders, RelativeOrders, CapitalOrders, CapitalOrdersBacklog, B</td>
<td>DesiredProduction, DesiredCapital</td>
</tr>
<tr>
<td></td>
<td>(Capital Self-ordering)</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: B-Backlog, C-Capital, S-Supply

*Table 5.4 – Feedback loops in the SILS – the simple long wave model (a)*
CHAPTER 5. AN EXTENSION OF LOOP DEACTIVATION IN THE BEHAVIOURAL METHOD

<table>
<thead>
<tr>
<th>Loop no.</th>
<th>Variable sequence</th>
<th>Unique consecutive edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3</td>
<td>Acquisitions, C, Depreciation, CapitalOrders, S</td>
<td>Depreciation, CapitalOrders, S</td>
</tr>
<tr>
<td></td>
<td>(Capital expansion)</td>
<td></td>
</tr>
<tr>
<td>L9</td>
<td>Acquisitions, C, Capacity, Production</td>
<td>Capacity, Production, Acquisitions</td>
</tr>
<tr>
<td></td>
<td>(Economic growth)</td>
<td></td>
</tr>
<tr>
<td>L11</td>
<td>Acquisitions, C, Capacity, Production, B</td>
<td>Production, B, Acquisitions</td>
</tr>
<tr>
<td></td>
<td>(Order fulfillment)</td>
<td></td>
</tr>
<tr>
<td>L14</td>
<td>Acquisitions, C, Depreciation, CapitalOrders, CapitalOrdersBacklog, B</td>
<td>CapitalOrdersBacklog, B, Acquisitions</td>
</tr>
<tr>
<td></td>
<td>(Backlog expansion)</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5 – Feedback loops in SILS – the simple long wave model (b)

Figure 5.7 – The behaviour of Desired Production
and its ABP over the entire simulation period are presented in Figure 5.7. We can see it is periodic and the cycle is approximately 45 years. The cycle in the middle is chosen for its clarity and completeness. We divide its behaviour into five phases based on the ABP. Figure 5.8 plots the behaviour in that cycle and the value of the ABP. Moreover, the time intervals are outlined in Table 5.6 to compare with the phase information in EEA from Kampmann’s analysis. Though the timing of shifts in dominance are different\(^1\), the behaviour patterns concur with each other in each phase and the durations of the corresponding phases are close. One discrepancy occurs in the end of the cycle, we identify a fifth linear phase while Kampmann attributed it to be convergent. This can be attributed to their different definitions of the phase.

<table>
<thead>
<tr>
<th>Phase name</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>behaviour pattern</td>
<td>exponential</td>
<td>logarithmic</td>
<td>exponential</td>
<td>logarithmic</td>
<td>linear</td>
</tr>
<tr>
<td>Time interval (\text{(ours)})</td>
<td>(46.5-49.25)</td>
<td>(49.50-58.5)</td>
<td>(58.75-60)</td>
<td>(60.25-66.75)</td>
<td>(67-90)</td>
</tr>
<tr>
<td>Time interval (\text{(EEA)})</td>
<td>(50-53)</td>
<td>(54-62)</td>
<td>(63-66)</td>
<td>(67-90)</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.6 – Phases and atomic behaviour patterns of Desired Production

The loop deactivation starts afterwards. We will highlight the analysis on the loops that have no unique edges and the other loops can be analyzed by the current behavioural method. Let us focus solely on the loops in Table 5.5 and demonstrate the new deactivation procedure step

\(^1\) The intervals in our analysis are approximately 4 years earlier than those in EEA, we can adjust our start time of the simulation 4 years behind the current start time to make the intervals match
CHAPTER 5. AN EXTENSION OF LOOP DEACTIVATION IN THE BEHAVIOURAL METHOD

by step. L3 is first selected as the candidate loop. The consecutive unique edges are Depreciation → CapitalOrder and CapitalOrder → Supply. Its deactivation is carried out as follows:

1. Create two nodes which serve as the copies of the two variables of the first edge in the unique consecutive edges. Two nodes are named DepreciationConstant and CapitalOrderTemp and correspond with Depreciation and CapitalOrder respectively. Link the new nodes with the same direction as their counterparts in the reference model. The screenshot of the added edge is shown by Figure 5.9(b).

2. Switch the tail of the edge from the middle variable in the unique edge pair to its copy while keeping the head of the edge unchanged. In this model, we delete CapitalOrders → Supply and add an edge CapitalOrdersTemp → Supply (see Figure 5.9(c)).

3. Add new edges starting from the variables which have links to variable in the middle of the unique edge pair (except the variable in the unique edge pair), and ending to the copy of that middle variable. We identify only one edge pointing to CapitalOrders except Depreciation → CapitalOrders, i.e., RelativeOrders → CapitalOrders. Therefore, we add a new edge RelativeOrders → CapitalOrdersTemp. Figure 5.9(c) shows the finally modified model after deactivating L3.

The structure of L3 has been modified and the feedback from Capital to Depreciation is cut and represented by a new loop: DepreciationConstant → CapitalOrdersTemp → S → Acquisitions → C. Then we have to set DepreciationConstant which is the copy of the origin in the unique edge pair to be a fixed value and modify equations when simulating the model with L3 deactivated:

\[ \text{CapitalOrdersTemp} = \text{DepreciationConstant} \times \text{RelativeOrders} \]
\[ \text{Supply} = \text{INTEG(CapitalOrdersTemp} - \text{Acquisitions, Supply)} \]

where DepreciationConstant refers to the value of Depreciation at the deactivation time.

Repeated procedures are applied to deactivate L9, L11 and L14. The resulting model is illustrated in a set of graphs (Figure 5.11, 5.13, and 5.15 respectively). Meanwhile, the reference model which highlights the unique consecutive two edges is also displayed for the purpose of clarification. We will outline the deactivation procedure on these loops.
Figure 5.9 – Loop capital expansion ($L3$) is deactivated
The unique consecutive two edges in the economic growth loop $L9$ are $\text{Capacity} \rightarrow \text{Production}$ and $\text{Production} \rightarrow \text{Acquisitions}$. Create two nodes ($\text{CapacityConstant}$ and $\text{ProductionTemp}$) and link them in the same direction as their counterparts in the reference model (Figure 5.11(a)). We then remove the edge $\text{Production} \rightarrow \text{Acquisitions}$ (represented by the dash line) and replace it by adding the edge $\text{ProductionTemp} \rightarrow \text{Acquisitions}$ for deactivating $L9$ (Figure 5.11(b)). Finally, all the variables which have links to $\text{Production}$ in the reference model are relinked to $\text{ProductionTemp}$ except for the variable $\text{Capacity}$ to recover other loops. Hence, one edge from $\text{CapacityUtilitization}$ to $\text{ProductionTemp}$ is added.
The changed equations in deactivating $L9$ are outlined below:

\[
\begin{align*}
    \text{ProductionTemp} & = \text{CapacityConstant} \times \text{CapacityUtilization} \\
    \text{Acquisitions} & = \text{Supply} \times \frac{\text{ProductionTemp}}{\text{Backlog}}
\end{align*}
\]

where $\text{CapacityConstant}$ is the value of $\text{Capacity}$ at the shifting time of each phase.

The third loop order fulfillment $L11$ is selected to be deactivated. $\text{Production} \rightarrow \text{Backlog}$ and $\text{Backlog} \rightarrow \text{Acquisitions}$ are the unique consecutive edges which distinguish $L11$ from other
loops in SILS. To avoid the deactivation of edge Production → Backlog affecting other loops, we maintain the original edge. However, in order to deactivate L11, we create a new edge ProductionConstant → BacklogTemp (Figure 5.13(a)) to work as a copy of Production → Backlog. Then, we have to add “inputs” (i.e., the terms on the right hand side of equation BacklogTemp) to the new node BacklogTemp. On account of the fact that BacklogTemp’s counterpart is Backlog, we add the same “inputs” to BacklogTemp (except Production as ProductionConstant serves as an “input”). Finally, we link BacklogTemp to Acquisitions and delete the edge Backlog → Acquisitions.

Based on the above analysis, we need to change equations as:

\[
\text{BacklogTemp} = \text{INTEG} (\text{CapitalOrdersBacklog} + \text{GoodsOrders} - \text{ProductionConstant}, \text{BacklogTemp})
\]

\[
\text{Acquisitions} = \text{Supply} \times \frac{\text{Production}}{\text{BacklogTemp}}
\]

where DepreciationConstant and the initial value of BacklogTemp refer to the values Depreciation and Backlog have at the deactivating time.

![Image](image.png)

**Figure 5.14** – Highlight of loop backlog expansion (L14) in the reference model

The final loop is the backlog expansion loop (L14). Its unique consecutive two edges are CapitalOrdersBacklog → Backlog and Backlog → Acquisitions. A new constant CapitalOrdersBacklogConstant is created to be a fixed inflow of BacklogTemp (Figure 5.15(a)). We add the other “inputs” of Backlog to BacklogTemp, i.e., Production and GoodOrders. Finally, this new substructure replaces Backlog in the input of Acquisitions in Figure 5.15(b). The deactivation of L14 is implemented by passing the constant inflow of BacklogTemp to Acquisitions without affecting the edge CapitalOrdersBacklog → Backlog in the reference model (Figure 5.14). Modifications are made according to the model in Figure 5.15(b) by the following
equations:

\[
\text{BacklogTemp} = \text{INTEG}\left(\text{CapitalOrdersBacklogConstant} + \text{GoodsOrders} - \text{Production}, \text{BacklogTemp}\right)
\]

\[
\text{Acquisitions} = \frac{(\text{Supply} \times \text{Production})}{\text{BacklogTemp}}
\]

where \(\text{CapitalOrdersBacklogConstant}\) refers to the value of \(\text{CapitalOrdersBacklog}\) at its deactivating time and the initial value of \(\text{BacklogTemp}\) corresponds to \(\text{Backlog}\) at that same time point.

5.3.3 Experiment result

Simulations on the modified model with the above four loops deactivated respectively are performed in phase I. We observe that none of their atomic behaviour patterns changes, as shown in Figure 5.16. This indicates that they are not dominant loops in this phase. Furthermore, deactivating the self-ordering loop \((L16)\) gives rise to the change of behaviour pattern from exponential to linear. Therefore, \(L16\) is considered to be the dominant loop in phase I. Examining the result from Kampmann’s eigenvalue analysis and Ford’s behavioural method in the same phase, we find our dominant loop test concurs with their conclusion, i.e., we all agree that, in phase I, the self-ordering loop \(L16\) contributes most to generating the exponential growth of \(\text{Desired Production}\).

In phase II, EEA concludes the economic growth loop \((L9)\) and capital expansion loop \((L3)\)
have the most significant influence on Desired Production growing increasingly slowly (49.50 - 58.5 year). Thus, we initially test L9. The modified model reduces the convergence of Desired Production, but still maintains a weak logarithmic growth. Then, the capital expansion loop (L3) is chosen to be tested alone. The result is as same as L9. The atomic behaviour pattern sustains convergent over phase II, although its value is close to zero. Taking into account the possibility of forming a shadow loop pair, we deactivate L3 and L9 simultaneously. The behaviour pattern now changes from logarithmic to roughly linear and we can see the change of the behaviour pattern values in Figure 5.17. As it is a numerical simulation and the data has precision limitation, it is difficult to observe exactly linear behaviour, so our conclusion is derived based on the comparison with others, it is the most close to zero. Since the dominance test when both loops are inactive is different from when deactivated individually, we conclude that that the capital expansion loop dominates but has a shadow feedback loop (Ford, 1999): the economic growth loop. This result is not identical with Ford’s analysis, which considers the economic growth is the only dominant loop. This is attributable to the way of deactivating the loop that differs. Specifically, deactivating L9 by its unique consecutive edges is weaker than by a unique edge. However, this is not a contradiction.

The precipitous decrease of Desired Production in phase III is 3 years in eigenvalue analysis while we identify it to be much smaller, i.e., 1.25 years. Ford obtained a similar result as ours where phase III consists of two phases, exponential and logarithmic. For the exponential
decay portion, the self-eroding loop \((L_{16})\) is the dominant loop, which is consistent with the eigenvalue analysis. We perform the dominance tests over all the candidate loops, and find the behaviour patterns remain the same (exponential) for all except the capital self-ordering loop \((L_{16})\). Deactivating \(L_{16}\) results the ABP in linear and this agrees with the result from Ford. Particularly, Figure 5.18 presents the individual ABP test result for the four loops in Table 5.5.

The gradual recovery of Desired Production characterizes phase IV. An identical analysis is conducted and the ABPs in Figure 5.19 shows that they are identical to that in the base run simulation, which suggests none of these four loops is dominant. In addition, as the found in Ford’s analysis, the capital decay loop \((L_2)\) is also a dominant loop. However, we identify that the loop of supply line 1-st order control \((L_1)\) is also a dominant loop. Figure 5.20 shows deactivating \(L_1\) generates the exponential behaviour in the middle of the phase.
Figure 5.19 – Atomic behaviour patterns in phase IV

Figure 5.20 – Atomic behaviour pattern of L1 deactivated in phase IV
Likewise, in the final phase, which is the late period of phase IV in EEA, we repeat the ABP tests. For the loops with unique edge pairs, the deactivation of each of them does not change the behaviour patterns at all. This is demonstrated in Figure 5.21 where corresponding four ABP lines lie at the horizontal zero axis. For the rest feedback loops, the ABP suggests the loop capital decay ($L_2$) dominates as the ABP changes from linear to the exponential in the later stage of phase V (see Figure 5.21). The result concurs with the conclusions from both Ford and Kampmann.

In order to clearly present the dominant loops identified by different method, we depict them in Table 5.7. In summary, our method of deactivating a loop with no unique edge by the unique consecutive two edges concurs with the previous analysis using the same model. This helps us to establish confidence in the new loop deactivation approach.

<table>
<thead>
<tr>
<th>Dominant loops</th>
<th>EEA</th>
<th>behavioural method (Ford(1999))</th>
<th>behavioural method (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$L_{16}$</td>
<td>$L_{16}$</td>
<td>$L_{16}$</td>
</tr>
<tr>
<td>II</td>
<td>$L_{3}, L_{9}$</td>
<td>$L_{9}$</td>
<td>$L_{3}, L_{9}$</td>
</tr>
<tr>
<td>III</td>
<td>$L_{16}$</td>
<td>$L_{16}$</td>
<td>$L_{16}$</td>
</tr>
<tr>
<td>IV (V)</td>
<td>$L_{1}, L_{2}$</td>
<td>$L_{2}$</td>
<td>$L_{1}, L_{2}$</td>
</tr>
</tbody>
</table>

Table 5.7 – Dominant loop analysis by different approaches
5.4 Conclusions and Future Work

Based on the work from (Ford, 1999) and (Phaff, 2008), we proposed a loop deactivation method by modifying its unique consecutive two edges. The simple long wave model is chosen to demonstrate this method and test its validity. A main contribution of this chapter is that we propose a new loop deactivation method by manipulating the unique pair of the two consecutive edges when neither the control variable nor the unique edge is available within a feedback loop in the behavioural method (assume the SILS is adopted as the candidate loop set). This improvement makes the behavioural method more robust and extends its applicability, as well as develop a systematic formal analysis software.

Nevertheless there are some limitations in the proposed approach: 1) When more than one pair of the unique consecutive edges are identified, it does not specify if using different pairs to deactivate the same loop would vary the atomic behaviour pattern. 2) It does not solve the loop deactivation thoroughly. A failure case is that the candidate loop cannot find even a single pair of unique consecutive two edge. Though it is unlikely in most cases, it is still a potential problem.

Another drawback associated with the behavioural method is using the atomic behaviour pattern to determine whether it is a dominant loop. This works when the behaviour is logarithmic or exponential, but it becomes vulnerable when the behaviour is linear. It is because the simulation result contains some approximations and its accuracy depends on many factors, such as the simulation timestep, the decimal digits that the result keeps. Therefore, it is unlikely to get a series of continuous zeros for the values of the behaviour pattern. A possible solution is to set a range for the linear behaviour pattern according to the specific application, within such range we consider it is linear. In this chapter, only one example is given to compare the analysis result with other methods. It is in need of abundant models and test examples to build confidence on this alternative loop deactivation approach.
In this chapter we will propose a variant of the behavioural method, a simulation based sensitivity analysis. As has been thoroughly discussed in both literature review chapter and previous chapter, the behavioural method is a procedure that identifies changes in the atomic behavior patterns in the presence, and absence, of a feedback loop (or a combination of feedback loops), in order to identify loop dominance within a specific time interval. Dominance is asserted provided the ABP changes in the absence of the candidate feedback loop.

The new variant method to be presented in this chapter combines both the behavioural method and the sensitivity analysis. Sensitivity analysis is used to determine how “sensitive” a model is to changes in the value of the parameters of the model and to changes in the structure of the model (Breierova and Choudhari, 1996). As the variant is a hybrid method, contributions from each method can be pinpointed at two different aspects: while the behavioural method provides a means to modifying the feedback structure, the sensitivity analysis offers a way to evaluate and compare the response based on the structure change. This approach has been disseminated in the publication (Huang et al., 2009).

In the following section, a detailed description of the proposed variant method is to be presented. Particularly, a reasonable selection of testing values is introduced as to ensure fair comparisons of the sensitivity between different feedback loops. In addition, the evaluation of various responses of the variable of interest due to manipulating different feedback loops is
suggested. Subsequently, a case study of the yeast model will be elaborated as to illustrate this approach in Section 6.2. A summary of this piece of work and the future research opportunities will be outlined in Section 6.3.

6.1 A Variant of the Behavioural Method: A Sensitivity Approach

The sensitivity analysis is usually performed with setting different values to the parameter to examine changes of a model’s behavior. By showing how the model responds to the change of the parameter, we could find the sensitive parameter to the model under the given set of conditions. However, in this method, the “parameter” refers to a variable which is selected from the candidate feedback loop. In addition, due to the deactivation of the corresponding loop in the model, the resulted responses can be utilized to investigate the dominant feedback loops.

The procedure of this hybrid approach is described as follows:

1. Select variable of interest in the model.

2. Run the simulation of the reference model, and perform the phase partition by calculating the ABPs at each time step.

3. Set up the a candidate loop pool with independent feedback loops, and identify all unique edges for each feedback loop.

4. Pick the control variable or if there is no control variable, pick the starting vertex of the unique edge as the control variable to deactivate the loop. Identify possible testing values for each control variable.

5. For each phase, apply the sensitivity analysis over the deactivated model. The loop deactivation is same as in the behavioural method, i.e., setting the control variable to a constant in the unique edge. The value of the control value at the deactivating point is termed as the “default value” henceforth. However, this time, the default value is replaced with the testing values, and instead of one simulation, the loop deactivation has to repeat until all the testing values run out.
6. Display the results from the sensitivity analysis and integrate these findings into a single graphical representation. An insight of which loop casts most significant influence will be developed after seeing graphs from all feedback loops. However, it is necessary to find a measurement to quantify the extent of the sensitivity.

In the above statements, there are two unsolved issues highlighted with italics: identify possible testing values for control variables, and find a measurement to quantify the extent of the sensitivity of the variable of interest to the feedback loop.

### 6.1.1 A Selection of Testing Values for Control Variables

We know that the selection of testing values determines the response of the variable of interest. In general, the broader range the testing values cover, the bigger variation of the resulted behaviour will be. Since different control variables are often in different scales, it is impractical to set same deviation (the range that the control variable varies) for all variables. The selection of testing values among these control variables then becomes a factor to be taken into consideration so that a fair comparison is secured.

The core of this issue is to identify a range for the testing values to vary. Since each deactivated model simulation is run over a partitioned phase. We suggest that a reasonable boundary for each control variable is determined by checking its range in the relevant phase from the reference simulation. The testing values can then be from a uniform distribution across this region. The number of testing values can be depending. We believe it is a fair representation of a variable’s boundary for sensitivity analysis to a specified phase and model, and makes the sensitivity comparison rational.

### 6.1.2 A Measurement to Quantify the Sensitivity

The second issue is to introduce a measurement to quantify the sensitivity based on the simulation results. As we know Ford uses the change of atomic behavior pattern to determine the dominant feedback loop. If the deactivated behavior pattern is consistent with that in the reference model, it would not be considered as a dominant loop. It then only provides a binary answer to the analyst: yes or no. We have poor information on how dominant that loop is and
how the other less significant loops perform throughout the phase. Nevertheless, this variant approach is able to visually demonstrate the degree of the sensitivity and it is also possible to quantify this extent over time.

In this approach, a close idea of residual sum of squares is adopted as the quantitative measurement. For a given model, we assume there are $T$ unique edges for $N$ candidate feedback loops, and $T \geq N$. The size of the testing values for a control variable is $d$. We use $v$ to denote the variable of interest. Its value in the deactivated model of loop $i$ at time $t$ is denoted as $v_i^t$. The behavior of the variable of interest $v$ in the deactivated model of loop $i$ with $j^{th}$ testing value at time step $t$ is then denoted by $v_{i,j}^t$. The behaviour in the deactivated model with the default value will be taken as the benchmark, and denoted as $v_{i,0}^t$, where the subscript $0$ refers to the default value. All other behaviours are to compare with this benchmark, and their difference at time $t$ is formulated as:

$$\Delta v_{i,j}^t = v_{i,j}^t - v_{i,0}^t \quad (6.1)$$

The variance $S$ between these two behaviour at $t$ yields:

$$S_{i,j}^t = (\Delta v_{i,j}^t)^2 \quad (6.2)$$

Subsequently, the calculation of the overall variance for $d$ sensitivity simulations regarding the feedback loop $i$ deactivated is shown in Equation 6.3, and the average value is obtained by dividing the overall variance with $d$ in Equation 6.4. This represent the average variance of loop $i$ sensitivity with respect to the variable of interest $v$.

$$S_{i}^t = \sum_{j=1}^{d} S_{i,j}^t \quad (6.3)$$

$$\bar{S}_i^t = \frac{S_i^t}{d} \quad (6.4)$$

Finally, we adopt the average standard variance $\sigma$ as the loop sensitivity measurement to identify the dominant feedback loops. This standard variance is presented in Equation 6.5. The bigger the standard variance is, the more sensitive the variable is to this loop, which means it is a dominant loop.

$$\sigma_i^t = \sqrt{\bar{S}_i^t} \quad (6.5)$$
This sensitivity measurement shows that the loop dominance can be assessed and compared on a scale evolving with time, thereby avoiding any binary classifications which can sometimes mask the underlying behaviors. A loop which is considered not dominant under the original criteria (ABP) can now be shown to have some degree of dominance relative to the other loops in a model. Conversely a loop classified as dominant can also be compared with its peers at each time step. We are able to make use of the visualized result to identify the shadow loops as well. The loops with relatively larger dominance would be selected as the dominant loops. Our measurement of the sensitivity shows more descriptive information and affords to reveal the fluctuation of loop dominance over time.

6.2 A Case Study of the Yeast Model

To illustrate the sensitivity approach, we choose the familiar yeast model as our study case. It is a classical example of overshoot-and-collapse dynamics. Due to the discussion of this model in previous chapter, we assume readers have some knowledge on it. The stock-flow diagram and the system equations are presented in Figure 6.1 and Table 6.1 respectively.

![Stock-flow diagram for the yeast model](image)

**Figure 6.1** – The stock-flow diagram for the yeast model

We carry out our approach step by step as follows:

1. Choose the variable of interest: Cells.
timestep=0.01 ; simulation length=90

Stocks:
Cells=INTEG(births-deaths, 1)
Alcohol=INTEG(alcohol generation, 0)

Flows:
births=Cells/division time* effect of alcohol on birth
deaths=Cells/life time* effect of alcohol on death
alcohol generation=Cells*alcohol per cell generation

Auxiliaries:
effectofalcoholonbirths=(-0.1*Alcohol)+1.1
effectofalcoholondeaths=EXP(Alcohol-11);
lifetime=30 ; division time=15
alcohol per cell generation=0.01

Table 6.1 – Equations for the yeast model

2. Run the simulation of the reference model, and obtain the behavior of the variable of interest. This is used to partition the phase based on the atomic behavior pattern. Figure 6.2 plotted the behavior of cell over a length of 90 units. Table 6.2 lists the phase partitions and its corresponding behavior patterns.

<table>
<thead>
<tr>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>exponential growth</td>
<td>logarithmic growth</td>
<td>exponential decay</td>
<td>logarithmic decay</td>
</tr>
<tr>
<td>0, 50.84</td>
<td>50.84, 65.49</td>
<td>65.49, 74.52</td>
<td>74.52, 90</td>
</tr>
</tbody>
</table>

Table 6.2 – Time intervals over simulation length and behavioral patterns

3. Identify the independent feedback loop set as its candidate loop set. Four feedback loops and their polarities are labeled in Figure 6.1, L1, the cell birth loop, is positive while the other three are all negative, including the cell death loop and the interactions loops with the alcohol.

Subsequently, the control variables and unique edges for individual candidate feedback loop are to be identified. Our observation shows there are two unique edges in L1 and L2 respectively, and two control variables eab and ead lie in L3 and L4 respectively. We outline them in Table 6.3 for clarity.

4. Identify the testing values for the sensitivity analysis. As described in the sensitivity approach, a boundary of each control variable has to be identified as to select the testing values. Figure 6.3(a) and 6.3(b) display the reference behavior of eab and ead while the
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

Figure 6.2 – Behavior of Cells over the simulation time

Table 6.3 – Candidate unique edges or control variables in each loop

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cells → births</td>
<td>Cells → deaths</td>
<td>eab</td>
<td>ead</td>
</tr>
</tbody>
</table>

behaviour of Cells is plotted earlier. In light of these graphs, we can locate the boundary for individual control variable, and the specific information is presented in Table 6.4. We use the notation “B” to refer to the boundary, and \( d \) to represent the size of testing values. The testing series is generated using the uniform distribution in Equation 6.6. In this case, we select 100 testing values across its boundary. These testing values are used to substitute for the default value in the deactivated model in turns.

Table 6.4 – Boundary for control variables in the yeast model

\[
\text{Min}(B) + \frac{\text{Max}(B) - \text{Min}(B)}{d} \times j \quad (j = 0 \ldots d - 1)
\] (6.6)

5. Apply sensitivity analysis on each deactivated model (associated with a certain feedback
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

6. Display the simulation results from the sensitivity analysis and use a measurement to quantify the extent of sensitivity. In each phase we plot a set of four figures to show the behavior of Cells under the sequential deactivation of four candidate loops. Figure 6.4(a) plots the behavior of Cells under the deactivation of edge Cells—births with a certain of different values on the control variable Cells in phase \( I \). This graph shows only 10 curves for clarity. The “default” in the legend refers to the “default value”. Likewise, we can identify an almost identical experiments performed involving loop \( L2 \), \( L3 \) and \( L4 \), and the results are shown in Figure 6.4(b), 6.4(c) and 6.4(d) respectively.

**Figure 6.3** – Behavior of variables in the reference model

(a) The behavior of effect of alcohol on birth

(b) The behavior of effect of alcohol on death
Figure 6.4 – Sensitivity analysis in phase I
Take Figure 6.4(a) for example, the plotted curves demonstrate how the variable of interest responds to the change of values on the control variable throughout phase I. Compared it with the other three peers in Figure 6.4, it is obvious that Cells is the most sensitive to the change of the control variable via the edge Cells→birth. In this scenario, we cannot simply address the variable of interest is sensitive to the control variable, because the $L1$ and the $L2$ both share the same control variable (Cells), and they exhibit completely different sensitivity. This suggests it is the feedback loop not the variable that gives rise to the different landscapes in Figure 6.4(a) and 6.4(b). Meanwhile, it is straightforward to see from Figure 6.4(b) and 6.4(d) that there is almost no change on the behavior of Cells no matter how the control variable varies across their boundaries in $L2$ and $L4$. Considering the control variable $eab$ in $L3$, a number of variance varies the behavior of Cells but it is not as significant as the $L1$.

Figure 6.5 is plotted under the proposed the sensitivity measurement. This measure is an indicator of the degree of sensitivity. This quantitative indicator presents how significant the edge (or loop) influences the variable of interest. In a summary, Cells is most sensitive to $L1$ and insensitive to $L2$ and $L4$ while Cells gradually increased its sensitivity to $L3$. In light of this knowledge, we conclude loop $L1$ is a dominant loop in phase I.

Likewise, the outcomes of the the behavior with loops deactivated in phase II are illustrated in Figure 6.6, and the standard variance measure of the edge sensitivity is depicted in 6.7. Edge Cells→death in $L2$ remains almost no impact on Cells as in phase I. No
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

Sensitivity analysis on L1 in Phase II

(b)

Sensitivity analysis on L2 in Phase II

(c)

Sensitivity analysis on L3 in Phase II

(d)

Sensitivity analysis on L4 in Phase II

Figure 6.6 – Sensitivity analysis in phase II

Figure 6.7 – Sensitivity analysis – standard variance in phase II
matter how the control variable changes across its boundary, the variable of interest exhibits almost the same trajectory. Loop $L_2$ thereby would not be a dominant loop. At the same time, $L_1$ now loses its dominance to $L_3$. Compared with $L_2$, $L_3$ becomes the most sensitive loop to the variable of interest in phase II as Figure 6.6(c) has the a bunch of most radial curves. Furthermore, the standard variance is calculated to quantify the sensitivity and shown in Figure 6.7. It reflects that the difference of sensitivity from different loops is much smaller than in phase I.

Figure 6.8 demonstrates the sensitivity analysis results on phase III. It is evident comparing with these graphs, the change in loop $L_4$ give rise to the most diverse behaviors of Cells. Consequently, $L_4$ is regarded as the dominant feedback loop. In contrast, we can see the variable of interest is insensitivity to the change made in $L_1$. Figure 6.8(b) and 6.8(c) show similar influences are casted to the variable of interest from $L_2$ and $L_3$. However, a more radial tail in Figure 6.8(b) suggests $L_2$ affect the behaviour more than $L_3$ in the long run. Figure 6.9 integrates the sensitivity analysis from four loops into one
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

Figure 6.9 – Sensitivity analysis in phase III–standard variance

Graph, the standard variance of sensitivity confirms the above analysis.

In final phase IV, the curves plotted in Figure 6.10(a) and 6.10(c) seem to be bundled in to a thick one, which indicates that the variable of interest is not sensitive to L1 and L3. However, the curves in the peers subfigures of 6.10(b) and 6.10(d) radiate to a much broader area. In addition with the standard variance calculated in Figure 6.11, we come to a conclusion that L2 is the loop that dominates the behaviour of Cells in phase IV.

In the end, it is necessary to compare the analysis results and verify this sensitivity approach. The behavioural method is thereby conducted over the yeast model for us to build confidence on it. The simulation results on the deactivated model in individual phase are displayed in Figure 6.12. The dominant feedback loops are identified by comparing the ABPs of the coloring curves with that of the black curve. Finally, the dominant feedback loops from both methods are outline in Table 6.5 for comparison. Despite slight disagreement in phase III, these two methods generate consistent results across other phases.

<table>
<thead>
<tr>
<th>Dominant loops</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity analysis</td>
<td>L1</td>
<td>L3</td>
<td>L4</td>
<td>L2</td>
</tr>
<tr>
<td>The behavioural method</td>
<td>L1</td>
<td>L3</td>
<td>L3 &amp; L4</td>
<td>L2</td>
</tr>
</tbody>
</table>

Table 6.5 – Dominant feedback loops comparison with different approaches
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

Figure 6.10 – Sensitivity analysis on edges in phase IV

Figure 6.11 – Sensitivity analysis in phase IV–standard variance
CHAPTER 6. A SENSITIVITY ANALYSIS EXTENSION TO THE BEHAVIOURAL METHOD

6.3 Conclusions and Future Research

The behavioural method is an important formal analysis method which offers a way to investigate the dynamic system. We propose a variant of this method by integrating a sensitivity analysis approach in order to enhance our understanding of the dominant loops in a model. In this method, a means of selecting testing values for the control variable is proposed. Moreover, the sensitivity of various feedback loops is taken as an indicator of dominance. A sensitivity measurement is proposed by computing the average standard variance of the deactivated behaviours. Using sensitivity analysis, we generate a broader set of results around a range of possible values for the control variable, and thereby assess the impact of these changes to the behavior of the variable of interest. The outcome of this analysis is a richer set of loop dominance analysis for each atomic behavior pattern in the model. It provides an ideal basis to identify the dominant loop by evaluating the sensitivity over various loops.

One potential drawback of this approach is it does not address the problem of selecting the...
unique edge to deactivate the feedback loop if there is more than one unique edge in the loop. Will deactivating the loop using different unique edges give rise to different behaviour? Would the selection of the unique edges change the conclusion of the dominant feedback loops? If both answers are yes, how can we identify the “right” one to represent the feedback loop to be deactivated. These problems are worth investigating in the future work.
Chapter 7

Software Design for the Dominant Loop Analysis Methods

Previous chapters discuss various dominant feedback loop analysis methods which are outlined in Figure 7.1. The computational methods comprise the behavioural method, and the behavioural method based sensitivity analysis while the analytical methods consist of the eigenvalue and eigenvector method. Sterman (2000) summarizes a list of future simulation software features which include “the automated identification of dominant loops and feedback structures”. In this chapter we will provide an overview of our design and architecture for the computational dominant feedback loop analysis methods for SD models. While the behavioural method has been implemented within a simulator and related aspects are to be discussed in the subsequent section, the conceptual modular design of the sensitivity analysis will be presented in a later section. This conceptual design is promising to be fully implemented as it is developed as an extension of the behavioural method. For the readers who are interested in the implementation of EEA, find (AbdelGawad et al., 2005) for reference.

7.1 The Design of the Computational Analysis Methods

The behavioural method of the computational analysis can be part of the a simulation tool through which it can obtain the necessary information (e.g., the model structure and the behaviour of each variable) to function appropriately. Therefore, it is accommodated within a
multi-method simulation tool for modelling complex systems named “iSimPlus”\(^1\) developed by the system dynamics research group in National University of Ireland, Galway.

### 7.1.1 iSimPlus

The iSimPlus provides a means to formalize, simulate and optimize models. The iSimPlus focuses on the simulations in terms of two aspects: system dynamics, centers the feedback perspective, and the agent-based computational modelling, which uses individual and their interactions as the basic building block. An overall architecture of this software tool is shown in Figure 7.2. The input is a text model file containing the model setup, e.g., start/finish time, timestep, the numerical method for solving the ODEs and network type, as well as the equations for all the variables used in the model. The functionality provided by each component is briefly introduced as follows:

- The parser reads the text file, checks the syntax errors and retrieve the variable information from the equations. More sophisticated actions involves sorting out the simulation orders for all variables, and finding the dependencies for any given variable. All the information is to be stored in appropriate data structures for later use in the other components.

---

\(^1\) iSimPlus stands for individual simulator Plus.
• The network models the interaction structures for collections of agents. The available network types are small world, scale free and random at present. This component is crucial in the agent-based model simulation.

• The solver mainly generates the model the numerical values of all variables of the system at each time step based on the network type and the specified numerical method (e.g., the Euler integration).

• The optimizer provides both the particle swarm optimization and genetic algorithm to find the best combination of parameter values in an objective function.

• The analyser implements the behavioural method to perform the dominant loop analysis. It interacts with the controller which contains all reference simulation results to perform the loop deactivation and resimulation every time after modification made to the model. Moreover, an interface to visualize the outcome of the analyser has been developed and integrated within iSimPlus. The dominant feedback loop analysis methods are constrained to work with the aggregate models currently. The analyser component is to be discussed in the following context.

Interested readers can find more information on the design of iSimPlus in (Duggan, 2008, 2011).

7.1.2 Analyser: Implementation of the Behavioural Method

In this section we will outline the functionality provided by the Analyser. The analyser implements the behavioural method improved by Phaff (2008), where the feedback loop is deactivated by its unique edge. The feedback loop cannot be analyzed if it does not have a unique edge in this version. The key elements in the Analyser are shown in Figure 7.3. In this graph, there are two streams going down marked with two different types of arrows. The dotted lined red arrow refers to the flow in the reference model while the solid lined blue arrow represents the flow in deactivated model. The analyser covers the following important elements:

1. The structural partition: including the loop partition and pathway partition.

---

2 The conventional SD models with integral equations and do not have agent information.
Figure 7.2 – The architecture design of iSimPlus.
2. Behaviour evaluator: evaluate the atomic behaviour pattern for a variable of interest at a given time interval.

3. Phase partition: partition the time interval based on the result from the behaviour evaluator.

4. Loop deactivation: fixing the control variable to be a constant (within a certain formula) for the candidate feedback loop.

5. Simulation: simulate the model for a given time interval.

6. ABP comparison: check if the ABPs from different behaviour are consistent.

An elaboration on each outlined element is presented below.

Figure 7.3 – The structured key components of the Analyser.
7.1.2.1 Structural Partition

The loop partition is responsible for systematically selecting feedback loops for any given model. The algorithm described in Algorithm 3 is used to extract the independent loop set. In addition, the unique edges are calculated in this routine.

The pathway partition is an extra functionality to the behavioural method but can be utilized by the eigenvalue elasticity analysis.

7.1.2.2 Behaviour Evaluator

This component utilizes the simulation behaviour to compute its second derivative, by which the ABPs are obtained over time (the definition of ABP can be found in Chapter 2.2.1). The output of the behaviour evaluator is used for the phase partition and ABP comparison for the identification of dominant feedback loops.

7.1.2.3 Phase Partition

The main task of the phase partition is to divide the entire simulation time into different phases based on the ABP of the reference behaviour variable of interest. The partitioned phase information is used to control the loop deactivation procedure, which is captured in Figure 7.3. Moreover, we provide a functionality to ignore the noise which refers to a scenario where a different behaviour pattern (from its two neighbouring phases) occurs and lasts for only a short time period. The minimum time threshold of a noise phase can be set from the user-interaction interface. Once it is set, any phase of a shorter duration will be considered as a noise and integrated into a subsequent phase.

7.1.2.4 Loop Deactivation

In case of more than one unique edges, the deactivation of this loop is conducted once (in each phase) by modifying one randomly selected unique edge. In this routine, the deactivation is achieved by setting the targeted variable to be constant within the particular equation (corresponds to the unique edge). This structural modification functionality is provided by the
CHAPTER 7. SOFTWARE DESIGN FOR THE DOMINANT LOOP ANALYSIS METHODS

parser. Subsequently, the system has to be restored to the state at a specific time (starting time point of a phase) before a simulation can be run correctly.

7.1.2.5 Simulation

The simulation for a deactivated model is achieved by getting access to the specially designed method provided in solver through the controller. This method requires the starting and finishing times to run a simulation. After each simulation, the behaviour evaluator is called to calculate the ABP and the new ABPs are recorded for the use of identifying dominant loops.

7.1.2.6 ABP Comparison

In each phase, on the completion of loop deactivation for all the feedback loops, the procedure of identifying the dominant feedback loops starts. The reference ABPs are compared with new ABPs resulted from a certain deactivated model. Once a mismatch between the ABPs is identified, the associated loop is considered as a dominant loop (it does not require the ABPs are different throughout the entire phase). There can be more than one feedback loops that dominate the behaviour. Whereas it is also likely that not a single loop dominates over a certain phase. In this case, a shadow loop analysis has to be conducted. In the latest version of iSimPlus, we have added the shadow loop analysis by which two feedback loops can be deactivated simultaneously. However, the simultaneous deactivation of more feedback loops is not available now.

7.2 A Conceptual Design of the Behavioural Method Based Sensitivity Analysis

The behavioural method based sensitivity analysis was discussed in chapter 6, and it shares some similarities with the behavioural method. As a result, some components in the Analyser can be reused in its implementation, and it is promising to implement this sensitivity analysis in iSimPlus. Figure 7.4 presents the conceptual design of the sensitivity analysis. Several components in this graph are outlined below.
1. Loop partition
2. Phase partition
3. Simulation
4. Loop deactivation: fixing the control variable with a number of testing values for each candidate feedback loop.
5. Sensitivity evaluation: quantify the sensitivity of each feedback loop based on the simulation results from multiple loop deactivations.
6. Sensitivity comparison: compare the sensitivity from various feedback loops and sort the loops with the descending order.

**Figure 7.4** – The conceptual design of the behavioural method based sensitivity analysis.

Compared with designed architecture of the behavioural method in Figure 7.3, we can see that the first three components are exactly same whereas the distinctions occur in the loop deactivation and the means of evaluating the dominant feedback loops, including the variance.
CHAPTER 7. SOFTWARE DESIGN FOR THE DOMINANT LOOP ANALYSIS METHODS

evaluation and its comparison. The following descriptions over the newly added components are presented according to Figure 7.4.

7.2.0.7 Loop Deactivation

Once the loop partition and phase partition are done, the loop deactivation begins. The means to deactivate a candidate feedback loop remains unchanged as in the behavioural method. However, this deactivation repeats \( d \) times (will be specified in the method) with varied values assigned to the targeted variable (the starting vertex of the unique edge) within each phase. The way of how to select testing values for a targeted variable was discussed in the earlier chapter 6.2. In Figure 7.4, the balls in filter represent deactivated models. The statement \( P_1 = V_{1i} \) appearing on the ball denotes that the targeted variable \( P_1 \) (representing loop 1) is set to be a constant \( V_{1i} \), similar denotations for others. Likewise, the overlapped files refers to different loops and the blank balls represent their deactivated models going through the filter for simulation. However this part is not exactly presented in graph and one filter is used for illustration purpose only.

7.2.0.8 Sensitivity Calculator

The sensitivity calculator evaluates the standard variance of the deactivated behaviour of the variable of interest. It intakes \( d \) rounds simulation results from for each candidate feedback loop including the the reference behaviour (deactivated with the default value\(^3\)), then calculates the average standard variance. In Figure 7.4, the default value is implicitly embedded in the testing value series. The average standard variance shows the average degree of deviation from a given feedback loop deactivated with a series of values to the reference behaviour. The steps for the average standard variance calculation can be found in Equation 6.1 - 6.5.

---

\(^3\) A value of the targeted variable at a the starting timestep of a phase in such a simulation that no loop is deactivated.
7.2.0.9 Sensitivity Comparison

The counterpart of this component in the behavioural method is ABP comparison in Figure 7.3. Through comparing the standard variance between individual loop, the dominant feedback loop is identified by seeking for a loop with biggest standard variance. Nevertheless, if two or more loop has close average variance value, it indicates there may be more than one dominant loops. This more sophisticated function should be added to the variance comparison component to enhance a more comprehensive analysis.

7.3 Summary

In this chapter, we have given an overview of the software design architecture of the computational dominant loop analysis methods. Initially, the simulator “iSimPlus” is introduced whereby the “Analyser” which implements the behavioural method can be implanted and follows a discussion of its architecture and key elements. Furthermore, we present a conceptual modular design on the behavioural method based sensitivity analysis. This design is a further development on the basis of the Analyser. Finally, the reusable components are highlighted while the new components that need to be added to accomplish the sensitivity analysis are discussed.
Chapter 8

Conclusions

The research presented in this dissertation demonstrates a number of contributions in the domain of feedback loop analysis. We have examined both analytical and computational methods in identifying dominant feedback loops and key elements. Each results chapter has attempted to investigate a certain existing method to propose improvements from a specific relevant aspect. These studies provide alternative ways and broaden our perspective to understand the underlying structural cause to the resulted behaviour.

Our research has focused on two feedback loop analysis methods, the analytical eigen-based analysis and the computational behavioural method. One closely linked topic relating to these methods is how to select candidate feedback loops to be analyzed. This dissertation has discussed a broad range of these topics, and shown through a couple of models the validation of the proposed new approaches, enhanced applicability and new insights shedding light on the policy making.

In the following sections we will outline a series of summaries stemming from each of our results chapters.

8.1 Loop Partition Algorithm

A loop selection algorithm is useful, particularly, in a medium or large sized model. It has significant implications in both the analytical and computational loop analysis. One well-known
and widely used loop selection algorithm is the SILS algorithm which is a special implementation of the independent loop set concept proposed by (Kampmann, 1996). Its origin and pivotal role in the eigenvalue elasticity analysis have been discussed and clarified in Chapter 3.

Close scrutiny shows the SILS is necessary to be a maximal independent loop set so as to generate correct loop elasticity. However, we identify the possible failure of SILS to deliver this task when we attempted to apply EEA to an agent-based goal diffusion model. In this example, the SILS algorithm identifies a smaller sized independent loop set. The cause of this failure is due to the SILS constrains itself to choose loop from only geodetic cycles. Therefore, we propose an algorithm that guarantees to identify a maximal set of independent loop. It is further validated by a series of models.

8.2 Eigenvector Analysis Method

Eigen-based analysis consists of eigenvalue elasticity analysis and eigenvector analysis. While extensive research has been focused on EEA, there has been a lack of study on eigenvector analysis. Our study has concentrated on the role of the eigenvector in the behaviour. A new representation of the state variable’s trajectory in the linear system is initially introduced, where the weight of each behaviour mode can be decomposed into the right, left eigenvector components and the system initial conditions. This reflects a fact that the eigenvalue determines the behaviour mode while the eigenvector affects its scale (or amplitude for oscillatory behaviour). Importantly, analytical solutions to compute the eigenvector-related (i.e., the mode weight) sensitivity in terms of various structure components, including the compact link, pathway and parameter are developed.

Through showing the labour-inventory model, the analytical eigenvector sensitivity results are validated by corresponding numerical tests. Moreover, different intervention points yielded by eigenvector analysis and EEA are compared. In this case study, both methods attempt to identify high leverage system parameters. By calculating parameter elasticities in both methods, it is possible to identify high leverage parameter that influences the mode weight and behaviour mode respectively. In the result comparison analysis, we find the eigenvector analysis produces better intervention points than EEA. The eigenvector analysis is therefore
efficient and potent in identifying high leverage point to improve the behaviour. We advise that using both methods will produce more comprehensive implications for policy designs.

8.3 Research Work Related to the Behavioural Method

This dissertation has conducted a series of studies regarding the behavioural method. Initially, one specific problem we tackle is to enable the loop deactivation when no unique edge exists in a feedback loop. As we know, a unique edge is least required to deactivate a loop, its influence over the variable of interest can then be assessed and compared with others. If no unique edge is available, the analysis is incomplete. We have proposed a solution to deactivate a loop without a unique edge in Chapter 5. The key idea is to identify two consecutive edges which exclusively belong to the candidate feedback loop, and a series modifications have to be done to make the loop deactivated as if it has a unique edge. The long wave model is examined using this extended method. We compare the dominant feedback loops with those from EEA, our method identifies all dominant loops in EEA, with one more loop that is considered to be dominant.

A proposal of a variant of the behavioural method in conjunction with the sensitivity analysis follows. In this approach, the trajectory in the traditional simulation over a loop deactivated model is regarded as the reference behaviour. Afterwards, a series of testing values are used to replace the control variable’s default value to run simulations on the same deactivated model. These resulted behaviours are utilized to calculate the average standard variance from the reference behaviour at each timestep. This serves as a measure of how sensitive that the variable of interest is to individual feedback loop. The loop with greatest value indicates a dominant feedback loop. Our experiment on the yeast model shows the analysis results from this variant approach are consistent with those in the behavioural method. We hope this variant method adds additional values to the behavioural method.

Finally, a software design to automate the behavioural method is presented in Chapter 7. The behavioural method has been implemented as an analysis package within a multi-method simulation tool for modelling complex systems named “iSimPlus”. An architecture of this simulation tool is outlined while the key components and their functionalities in the analysis package are elaborated. In addition, a conceptual design of the behavioural method based sensitivity
analysis is discussed in the context of the iSimPlus simulation software in the end.

8.4 Future work

A number of open research questions stem from the work outlined in this dissertation. Firstly, the validity of our proposed methods is concluded based on limited number of study cases. There are still many case models which should be studied to examine their validity. Some drawbacks and limitations are elusive, and can only be identified through applying these methods on more examples. Further refinements and improvements are possibly built on accordingly. For example, more models should be studied to exploit the merits and weaknesses of the variant of the behavioural method.

Secondly, a remaining task in the eigenvector analysis is to develop an analytic solution to the eigenvector sensitivity (or elasticity) with respect to the feedback loops. While formulations of eigenvector sensitivity with respect to other structural components, including the compact link, pathway, and parameters are proposed, the eigenvector sensitivity to loop has not been solved. It is interesting to examine how the feedback loop influences the scales of the interested behaviour mode. It allows one to explore the potential policy in a much broader space and provides practical implications from an alternative perspective.

Finally, an issue related to the dominant feedback loop analysis for practitioners, is how to make use of the identified dominant feedback loops and modify the system structure to obtain the desired behaviour. Such analysis helps one to understand the underlying structural cause of the focused behaviour and verify the model structure. However, it is equally important to modify the dominant feedback structure to adjust the problematic behaviour. Modifications on high leverage feedback loops are not easy to implement. In general, we change the loop structure by modifying its constituent links or the relevant parameter values. For the loops with unique edges or parameters (i.e., exclusively belong to one loop), it is understandable that the change takes place within the loop is independent of others. Nevertheless, for the loops that have no unique components, any change made to its components simultaneously affects other feedback loop(s), and the aggregated influences from them may cause unexpected behaviour. It is therefore difficult to find proper intervention points with the knowledge of the dominant feedback loops, and sometimes adding or removing structural components will be necessary.
We believe it is worth great efforts to accelerate the transfer from the dominant feedback loop analysis result to the implementations in real world.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


Saleh, M., Davidsen, P., 2007. Extending eigenvalue analysis to nonlinear models via incorporating higher order terms of taylor series expansion, The 25th International Conference of the System Dynamics Society, Boston, MA, USA.


Appendix A

Conjugated eigenvalues and the oscillatory behaviour

The conjugated complex eigenvalues are always associated with the oscillatory behaviour. A mathematical explanation is provided in the following context.

In a real-valued square matrix, if there exists complex-valued eigenvalues, they must appear in conjugates, \( \lambda_i = \bar{\lambda}_{i+1} \), so do the weights associated with these two behaviour modes \((\alpha_i^0 r_{ij} \text{ and } \alpha_{i+1}^0 r_{(i+1)j} \text{ for } x_j)\).

Let us assume \( \lambda_i = a + bi, \lambda_{i+1} = a - bi \), and their associated weights to be \(c + di\) and \(c - di\) respectively. These two behaviour modes are added up in the state trajectory to be:

\[
(c + di)e^{t(a+bi)} + (c - di)e^{t(a-bi)} = e^{at} \left\{ (c + di)e^{bt}i + (c - di)e^{-bt}i \right\}
\]

Substitute Euler’s formula: \(e^{ix} = \cos x + i \sin x\) for above equation, we obtain:

\[
e^{at} \left\{ (c + di)(\cos bt + i \sin bt) + (c - di)(\cos bt - i \sin bt) \right\} = e^{at} \left\{ 2c \cos(bt) - 2d \sin(bt) \right\} = 2\sqrt{c^2 + d^2}e^{at} \sin(\theta - bt) \text{ note: } \sin\theta = \frac{c}{\sqrt{c^2 + d^2}}
\]

Finally, the behaviour associated with a pair of conjugated eigenvalues is oscillation whose amplitude is \(2\sqrt{c^2 + d^2}e^{at}\) and the frequency is determined by the imaginary part of the eigenvalues, \(b/2\pi\).
Appendix B

Computation of eigenvalue sensitivity with respect to the compact link gain

For any square matrix $A$ $(n \times n)$ has a distinct eigenvalue $\lambda_i$, the sensitivity of the eigenvalue with respect to any entry of the matrix $A$, $A_{pq}$, is equal to the product of the $p$th component in the left eigenvector and the $q$th component in the right eigenvector (both are associated with $\lambda_i$):

$$S^\lambda_{pq} = \ell_i^H \times r_{iq}$$

**Proof:**

For the eigenvalue $\lambda_i$, we have:

$$Ar_i = \lambda_i r_i$$  \hspace{1cm} (B.1)

$$\ell_i^H A = \lambda_i \ell_i^H$$

where the eigenvectors are normalized:

$$\ell_i^H r_i = 1$$

Differentiating Equation B.1 with respect to the entry $A_{pq}$ gives:

$$\frac{\partial A}{\partial A_{pq}} r_i + A \frac{\partial r_i}{\partial A_{pq}} - \frac{\partial \lambda_i}{\partial A_{pq}} r_i - \lambda_i \frac{\partial r_i}{\partial A_{pq}} = 0$$

$$\left( \frac{\partial A}{\partial A_{pq}} - \frac{\partial \lambda_i}{\partial A_{pq}} I \right) r_i + (A - \lambda_i I) \frac{\partial r_i}{\partial A_{pq}} = 0$$  \hspace{1cm} (B.2)

Equation B.2 is now pre-multiplied by $\ell_i^H$. The second term becomes zero and it can be rewritten as:

$$\ell_i^H \frac{\partial \lambda_i}{\partial A_{pq}} r_i = \ell_i^H \frac{\partial A}{\partial A_{pq}} r_i$$

$$\frac{\partial \lambda_i}{\partial A_{pq}} \ell_i^H r_i = \ell_i^H \frac{\partial A}{\partial A_{pq}} r_i$$

$$S^\lambda_{pq} = \ell_i^H r_{iq}$$

195
Appendix C

Computation of Eigenvalue Elasticity with respect to a Loop

Let the identified pathways be $P_{ij_1}, P_{ij_2}, \ldots$ and $P_{ij_N}$ where $N$ is the total number of pathways that starts from $x_i$ and ends at $x_j$:

$$ P_{ij} = \{P_{ij_1}, P_{ij_2}, \ldots, P_{ij_N}\} $$

Let the gains of those pathways be $g_{ij_1}, g_{ij_2}, \ldots$, and $g_{ij_N}$ respectively. The compact link gain ($A_{ij}$) equals to the sum of the pathways gain which pass through the two states, $x_i$ and $x_j$.

$$ A_{ij} = \sum_{P_s \in P_{ij}} g_{p_s} \quad (C.1) $$

Suppose the dominant eigenvalue is $\lambda_k$, so the eigenvalue elasticity with respect to a certain compact link, e.g., $A_{ij}$, $\varepsilon_{kij}$ can be obtained from the equation below Saleh and Davidsen (2000):

$$ \varepsilon_{ij} = \frac{\partial \lambda_k}{\partial A_{ij}} * \frac{A_{ij}}{\lambda_k} = L_k(i) * R_k(j) $$

where $L_k(i)$ refers to the $i^{th}$ value of $k^{th}$ left eigenvector, and $R_k(i)$ refers to the $i^{th}$ value of $k^{th}$ right eigenvector (both eigenvector correspond to the eigenvalue $\lambda_k$) and these can be computed from the gain matrix (constituted by compact link gains). The eigenvector matrices satisfy $LR = I$.

Regarding the elasticity with respect to a certain pathway passing from $x_i$ to $x_j$, e.g., $P_s$, we calculate it by the following equation:

$$ \varepsilon_{ik} = \frac{\partial \lambda_k}{\partial g_{ps}} * \frac{g_{ps}}{\lambda_k} = \frac{\partial \lambda_k}{\partial A_{ij}} * \frac{\partial A_{ij}}{\partial g_{ps}} * \frac{g_{ps}}{\lambda_k} $$

$$ = \frac{\partial \lambda_k}{\partial A_{ij}} * 1 * \frac{g_{ps}}{\lambda_k} = \frac{\partial \lambda_k}{\partial A_{ij}} * \frac{A_{ij}}{\lambda_k} * \frac{g_{ps}}{A_{ij}} $$

$$ = \varepsilon_{ij} * \frac{g_{ps}}{A_{ij}} $$
APPENDIX C. COMPUTATION OF EIGENVALUE ELASTICITY WITH RESPECT TO A LOOP

Note that one pathway belongs to only one compact link. In light of Equation C.1, we know that the partial derivative of $A_{ij}$ with respect to a pathway $P_s$ equals to 1 if $P_s \in P_{ij}$, otherwise 0. The above equation states the pathway elasticity equals to the corresponding compact link elasticity multiply by their gain ratio (pathway gain over the compact link gain).

The gain of each individual pathway i.e. the values of $g_{p_{ij1}}$, $g_{p_{ij2}}$, ... and $g_{p_{ijN}}$ could easily be computed by multiplying the gains of the elements $g_{lr}$ (gain of causal links).

$$g_{p_{ij}} = \prod_{l_r \in P_{ij}} g_{lr} \quad (C.2)$$

So we have:

$$\frac{\partial g_{p_{ij}}}{\partial g_{lr}} = \frac{g_{p_{ij}}}{g_{lr}}$$

when the link $l_r$ lies in the pathway; and

$$\frac{\partial g_{p_{ij}}}{\partial g_{lr}} = 0$$

when the link $l_r$ does not lie in the pathway. Also note that one causal link could be a member of more than one pathway. Therefore, the causal link elasticity can be written as:

$$\varepsilon_{k l_r}^l = \frac{\partial \lambda_k}{\partial g_{lr}} \frac{g_{lr}}{\lambda_k} = \sum_{P_s \subset \text{all pathways}} \frac{\partial \lambda_k}{\partial g_{ps}} \frac{g_{ps}}{\lambda_k} \frac{\partial g_{ps}}{\partial g_{lr}} \frac{g_{lr}}{\lambda_k}$$

$$= \sum_{P_s \supset P_l} \frac{\partial \lambda_k}{\partial g_{ps}} \frac{g_{ps}}{g_{lr}} \frac{g_{lr}}{\lambda_k}$$

$$= \sum_{P_s \supset P_l} \varepsilon_{k p_s}^l$$

(C.3)

The equation indicates the causal link elasticity can be obtained by adding up all the pathway elasticities which pass through the causal link.

Let the loop denoted by $C_1$, $C_2$, ..., $C_M$ and the gain of those loop be $g_{c_1}$, $g_{c_2}$, ..., $g_{c_M}$ respectively, where M is the total number of loops. The loop gain is equal to the product of gains of causal links in the loop:

$$\frac{\partial g_m}{\partial g_{lr}} = 0 \quad (C.4)$$

when the causal link $g_{lr}$ lies in loop $m$; and

$$\frac{\partial g_m}{\partial g_{lr}} = \frac{g_m}{g_{lr}} \quad (C.5)$$

when causal link $g_{lr}$ does not lie in loop $m$. In light of Equation C.4 and Equation C.5, loop
APPENDIX C. COMPUTATION OF EIGENVALUE ELASTICITY WITH RESPECT TO A LOOP

Elasticity can be derived from causal link elasticity as follows:

$$
\varepsilon_{l_r}^k = \frac{\partial \lambda_k}{\partial g_{l_r}} \frac{g_{l_r}}{\lambda_k} = \left\{ \frac{\partial \lambda_k}{\partial g_{c_1}} \frac{g_{c_1}}{\lambda_k} \ast \frac{\partial g_{c_1}}{\partial g_{l_r}} \ast \ldots \ast \frac{\partial \lambda_k}{\partial g_{c_M}} \frac{g_{c_M}}{\lambda_k} \frac{g_{l_r}}{\lambda_k} \right\} 
\ast \frac{g_{l_r}}{\lambda_k}
\sum_{C_m \supset l_r} \varepsilon_{c_m}^k
$$

The equation states that the elasticity of an eigenvalue with respect to the gain of a causal link is equal to the sum of the elasticities of the eigenvalue with respect to the gains of all the loops that pass through this link.
Appendix D

Results on the Fully-connected Goal Diffusion Model (Three Agents)

The equations of the fully-connected three-agent goal diffusion are made up as follows:

\[
\text{GoalAgent}_i = \text{INTEGRAL}(\text{ChangeInGoal}_i, \text{GoalAgent}_i^0)
\]

\[
\text{ChangeInGoal}_i = \frac{\text{TargetGoal}_i - \text{GoalAgent}_i}{\text{AT}_i}
\]

\[
\text{TargetGoal}_i = \frac{\text{GoalAgent}_1 + \text{GoalAgent}_2 + \text{GoalAgent}_3}{3}
\]

\[
\text{AT}_1 = 10 \quad \text{AT}_2 = 5 \quad \text{AT}_3 = 1
\]

where \(i = 1, 2, 3\). The loop elasticity computations are implemented in MATLAB. The gain matrix of this system is:

\[
A = \begin{pmatrix}
\frac{-2}{3 AT_1} & \frac{1}{3 AT_1} & \frac{1}{3 AT_1} \\
\frac{1}{3 AT_2} & \frac{-2}{3 AT_2} & \frac{1}{3 AT_2} \\
\frac{1}{3 AT_3} & \frac{1}{3 AT_3} & \frac{-2}{3 AT_3}
\end{pmatrix}
= \begin{pmatrix}
\frac{-1}{15} & \frac{1}{15} & \frac{1}{15} \\
\frac{1}{15} & \frac{-2}{15} & \frac{1}{15} \\
\frac{1}{3} & \frac{1}{3} & \frac{-2}{3}
\end{pmatrix}
\]

and its eigenvalues are:

\[
\lambda_1 = -0.7181; \quad \lambda_2 = 0; \quad \lambda_3 = -0.1485.
\]

The intermediate results of the compact link elasticity, pathway elasticity and causal link elasticity are not shown here. Table D.1 reports the loop elasticity for the goal diffusion model calculated with two algorithms SILS and ILS (\(\lambda_2 = 0\) cannot be computed). The loop IDs are consistent with the loop markings in Figure 3.2 and Table 5.3, and the blank cells correspond to elasticity of the missing loop in the SILS.

Though it does not make sense to calculate the loop elasticity using the SILS, we still outline its results for the purpose of comparing with their counterparts in the ILS. Observation of this table shows some of the results are close, while some are of opposite signs, e.g., Loop 1, 5, 8 in terms of \(\lambda_1\). Elasticities with opposite signs generate counter policies. It is thus critical to ensure the independent loop set is complete, which can finally generate correct loop elasticity.
### Table D.1 – Loop elasticity in the 3-agent goal diffusion model.

<table>
<thead>
<tr>
<th>Loop ID</th>
<th>$\lambda_3$</th>
<th>$\lambda_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ILS</td>
<td>SILS</td>
</tr>
<tr>
<td>1</td>
<td>0.0132</td>
<td>-0.0053</td>
</tr>
<tr>
<td>2</td>
<td>-0.0790</td>
<td>-0.0323</td>
</tr>
<tr>
<td>3</td>
<td>-0.0941</td>
<td>-0.0464</td>
</tr>
<tr>
<td>4</td>
<td>0.1099</td>
<td>0.1461</td>
</tr>
<tr>
<td>5</td>
<td>0.0237</td>
<td>-0.0038</td>
</tr>
<tr>
<td>6</td>
<td>0.0379</td>
<td>0.0372</td>
</tr>
<tr>
<td>7</td>
<td>0.1218</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.0082</td>
<td>0.0281</td>
</tr>
<tr>
<td>9</td>
<td>0.2899</td>
<td>0.2685</td>
</tr>
<tr>
<td>10</td>
<td>-0.5135</td>
<td>0.1165</td>
</tr>
</tbody>
</table>
Appendix E

Equations of the Labor-inventory Model

Stocks:
INVENTORY=(Production_Rate-Shipmen rate, 50000)
LABOR=(Hiring_Rate-Quit_Rate, 1000)
VACANCY=(Vacancy_Creation_Rate-Vacancy_Closure_Rate, 150)
WORK_IN_PROCESS_INVENTORY=(Production_Start_Rate-Production_Rate, 60000)

Flows:
Production_Rate=WORK_IN_PROCESS_INVENTORY/manufacturing_cycle_time
Shipment_rate=customer_order_rate
Hiring_Rate=VACANCY/average_time_to_fill_vacancy
Quit_Rate=LABOR/average_duration_of_employment
Vacancy_Creation_Rate=Adjust_for_vac+Desired_hiring_rate
Vacancy_Closure_Rate=Hiring_Rate
Production_Start_Rate=LABOR*productivity*standard_work_week

Auxillaries
Adjust_for_vac=(Desired_vacancy-VACANCY)/vacancy_adj_time
Desired_vacancy=Desired_hiring_rate*average_time_to_fill_vacancy
Desired_hiring_rate=Adjust_for_labor+Quit_Rate
Adjust_for_labor=(Desired_labor-LABOR)/labor_adj_time
Desired_labor=Desired_production_start_rate/(productivity*standard_work_week)
Desired_production_start_rate=Adj_for_wip+Desired_production
Adj_for_wip=(Desired_wip-WORK_IN_PROCESS_INVENTORY)/wip_adj_time
Desired_wip=Desired_production*manufacturing_cycle_time
Desired_production=customer_order_rate+Production_adj_from_inv
Production_adj_from_inv=(desired_inventory-INVENTORY)/inv_adj_time

Constants
APPENDIX E. EQUATIONS OF THE LABOR-INVENTORY MODEL

vacancy_adj_time=4
manufacturing_cycle_time=8
customer_order_rate=10000
average_time_to_fill_vacancy=8
average_duration_of_employment=100
productivity=0.25
standard_work_week=40
labor_adj_time=19
wip_adj_time=6
inv_adj_time=12
desired_inventory=40000
Appendix F

Equations for the Simple Long Wave model

timestep dt=0.25 years; start time=0.0; Euler integration method.

Stocks
Backlog = INT(CapitalOrdersBacklog+GoodsOrders-Production, NormalDeliveryDelay)
Supply = INT(CapitalOrders-Acquisitions,(Backlog/Production)*Depreciation)
Capital = INT(Acquisitions-Depreciations,
CapitalOutputRatio*AvgLifeOfCapital/(AvgLifeOfCapital-CapitalOutputRatio))

Flows
CapitalOrders = Depreciation*RelativeOrders
Acquisitions = Supply*Production/Backlog
Depreciation = Capital/AvgLifeOfCapital
CapitalOrdersBacklog = CapitalOrders
Production = Capacity*CapacityUtilization

Auxiliaries
RelativeOrders = RelativeOrderfnc(DesiredOrders/Depreciation)
DesiredOrders = CapitalAdjustment+Depreciation+SupplyAdjustment
CapitalAdjustment = (DesiredCapital-Capital)/CapitalAdjustTime
Capacity = Capital/CapitalOutputRatio
DesiredCapital = CapitalOutputRatio* DesiredProduction
CapacityUtilization = CapacityUtifnc(DesiredProduction/Capacity)
SupplyAdjustment = (DesiredSupplyLine-Supply)/SupplyAdjustTime
DesiredProduction = Backlog/NormalDeliveryDelay
DesiredSupplyLine = Depreciation*Backlog/Production
AvgLifeOfCapital = 20
CapitalOutputRatio = 3
CapitalAdjustTime = 1.5
SupplyAdjustTime = 1.5
NormalDeliveryDelay = 1.5
GoodOrders = 1
RelativeOrderfnc = {(-1,0),(-0.5,0),(0,0.2),(0.5,0.5),(1,1),(1.5,1.5),(2,2),(2.5,2.5),
(3,3),(3.5,3.5),(4,4),(4.5,4.4), (5,4.8),(5.5,5.2),(6,5.5),(6.5,5.65),(7,5.7),(7.5,5.75),(8,5.8),(40,6)}
CapacityUtifnc = {0,0.2,0.3),(0.4,0.6),(0.6,0.8),(0.8,0.9),(1,1),(1.2,1.03),(1.4,1.05),
(1.6,1.07),(1.8,1.09),(2,1.1)}