In spite of the energy crisis, population and environment degradation issues, the use of automobiles has been going up. This call for continuing the efforts towards developing more efficient, environmentally friendly, safer and more controllable vehicles. This often translates into developing better models and increasing the use of onboard computers. The use of computers for control invariably requires models which execute faster and are reliable even in extreme conditions. Bond graph based techniques allow the development of continuously extensible models and easier integration with control systems.The present work deals with the development of the so called half car models using Bond graph based approaches to study the response of the vehicle while passing over a ramp or uneven surface.A successful compilation of the Bond graph on the Bond graph package Symbol Shakti shows that the model has been created with logical correctness.More extensive validation may be needed before it can be taken up for testing its utility for online control.

Manoj Kumar Singh

He received B.Tech,M.Tech and Ph.d (Pursuing), from U.P. Technical University, Lucknow and member of ASME,ISTE and MIE . He is Assistant Professor-III , Mechanical Engineering Department, School of Management Sciences, Technical Campus, Lucknow . He has published 54 National /International research papers and Guided 27 B.Tech Student Projects.

978-3-659-61086-8

MSD Half Car Using Bond Graph

Singh, Singh, Faruqi

Manoj Kumar Singh
Bharat Raj Singh
M.A. Faruqi

Modeling And Simulation Of Dynamic Half Car Using Bond Graph

Automobiles Continue to be Central Present Day Human Activities and Literature Exists on Their Evolution and Adaptation

LAP LAMBERT Academic Publishing

Manoj Kumar Singh
Bharat Raj Singh
M.A. Faruqi

Modeling And Simulation Of Dynamic Half Car Using Bond Graph

Manoj Kumar Singh
Bharat Raj Singh
M.A. Faruqi

Modeling And Simulation Of Dynamic Half Car Using Bond Graph

Automobiles Continue to be Central Present Day Human Activities and Literature Exists on Their Evolution and Adaptation

LAP LAMBERT Academic Publishing

Impressum / Imprint
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:
LAP LAMBERT Academic Publishing
ist ein Imprint der / is a trademark of
OmniScriptum GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Germany
Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /
Printed at: see last page
ISBN: 978-3-659-61086-8

ACKNOWLEDGEMENT

The author deeply express his heartfelt respect and gratitude to the Prof. (Dr.) M. A. Faruqi, Ex- Professor, Mechanical Engineering Department, Indian Institute of Technology, Kharagpur, who has always been a source of encouragement.

The author would like to highly appreciate and express thanks to the Dr. M.P.Singh, Executive Secretary and Shri Sharad Singh, Secretary and CEO, School of Management Sciences,Technical Campus Lucknow, who have always been helpful to the author directly and indirectly for getting the present work carried out.

The author cannot forget to mention the moral support provided by his parents, parents of his wife, wife and the daughter, without their enlightening support this book could have not taken shape.

The author extremely grateful to Prof.(Dr.) Amalendu Mukherjee, Professor, Mechanical Engineering Departmet, Indian Institute of Technology Kharagpur, who always believed that i am devoted to a meaningful pursuit.

The author also would like to thank to the Prof.(Dr.) Manoj Mehrotra, Director, SMS-IMS,Lucknow, Prof.(Dr.) Bharat Raj Singh, Director, SMS-IT, Lucknow, Prof.(Dr.) Dharmendra Singh, Dean Academics, SMS-IT, Lucknow, Dr. P.K.Singh,COE,SMS-IT, Lucknow for their encouragement and support.

Manoj Kumar Singh

DEDICATION

This book is dedicated to my beautiful wife Sabita Singh and my daughter Kasturi Kumari . My wife has always been the closest person during my studies.

Her constant support in every possible way has always helped me to move forward to reach my goals. I would not be able to achieve my success without her support.

I also want to dedicate this work to my parents and parents of my wife. Thank you all for all your advice and support.

TABLE OF CONTENTS

Details			Page No.
Acknowledgement			i
Dedication			iii
Table of contents			v-vii
List of tables			viii
List of figures			ix
Nomenclature			x-xi
Abbreviations			xii
Chapter1	**-**	**Introduction**	**1-3**
Chapter 2	**--**	**Literature Review**	**4- 30**
	2.1	Scope of Modeling and Simulation	4
	2.2	The development of vehicle stability control	4-5
	2.3	Dynamic model using bond graph	5-6
	2.4	Trailing Arm Suspension for Heavy Trucks	6-7
	2.5	Target cascading	7-8
	2.6	Hydraulic system design and system modeling	8
	2.7	Modeling of an Automobile Suspension	9
	2.8	Systematic identification of dynamic system	10
	2.9	Vehicle dynamic and design	10-11
	2.10	Modeling and simulation of vehicular power systems	12-13
	2.11	Four wheel vehicle with passive suspension	14

	2.12	A full car model with passive suspension	15
	2.13	Modeling and simulation of hybrid electric vehicle	15-17
	2.14	Dynamic modeling and simulation of electric vehicles	17-18
	2.15	Development of a full car model	18-19
	2.16	A vehicle for delivering laboratory experience	19-20
	2.17	A half car active suspension model	20-21
	2.18	Bond graph modeling and simulation software	21-30
Chapter 3	**-**	**Present work**	**31-54**
	3.1	Introduction	31-32
	3.2	Development of the model	32-33
	3.3	Modeling the dynamics of an automobile with road excitation	33-53
	3.4	Results and conclusions	54
Chapter 4	**-**	**Results and discussions**	**55-56**
	4.1	Modeling	55-56
	4.2	Discussions and possibilities of future works	56
References			**57-58**
Appendices			**59-84**
	I	**Bond graph**	**59-61**
	I.1	Introduction	59-61
	II	**Concept of Bond graph elements**	**62-70**
	II.1	Introduction	62-63
	II.2	Source of effort	63
	II.3	Source of flow	63-64
	II.4	The Inertia element	64
	II.5	The Compliant element	65-66
	II.6	The resistive element	66-67

II.7	Transformer	67
II.8	Creation of system equation	67-68
II.9	Generation of system equation	68-69
II.10	Creation of system bond graph	69-70
III	**Main feature of Symbolshakti software**	**71-78**
III.1	Symbols bond pad	71
III.2	Symbols simulator module	72-73
III.3	Symbols compilation	73-75
III.4	Symbols simulator module circuit	76-77
III.5	Symbols onlineplot	77-78
IV	**C++ code to a machine language excutable code**	**79-84**

LIST OF TABLES

Table No.	Captions	Page No.
Table 3.1	The description of the elements of the Bondgraph	35
Table3.2	Parameter of half car model with fixed suspension	36
Table I.1	Efforts and Flow Variables	61

LIST OF FIGURES

Figure Nos.	Captions	Page Nos.
Fig 2.1	Driver/system/vehicle interaction	5
Fig.2.2	VOAS-2 suspension on test vehicle	7
Fig.2.3	Half-car model	8
Fig.2.4	Spring mass- damper model	9
Fig.2.5	Vehicle systems models	11
Fig.2.6	Vehicle longitudinal forces	13
Fig.2.7	Full car suspension model	14
Fig.2.8	Full car model	19
Fig.3.1	Half car model with a fixed Suspension	35
Fig.3.2	1 Junction element models	37
Fig.3.3	0 Junction element models	39
Fig.3.4	Bondgraph model of half car model with a fixed Suspension	41
Fig.3.5	Generation of equation of half car model with fixed suspension	42
Fig.3.6	Rocking motion of the half car model at front suspension with vertical angular displacement C.G	44
Fig.3.7	Heaving motion of the half car model at rear suspension with vertical displacement C.G.	44
Fig.3.8	Rocking motion of the half car model at front suspension with vertical angular displacement C.G	45
Fig.3.9	Heaving motion of the half car model at rear suspension with vertical displacement C.G.,	45
Fig.3.10	Half car model with hinged suspension	49
Fig.3.11	Bondgraph model of half car model with a hinged suspension	51
Fig.3.12	Generation of equation of half car model with hinged suspension	53
Fig.3.13	Heaving motion of the half car model at rear suspension	53
FigIII.1	Symbols bond pad	72
FigIII.2	Symbols simulator module	72
FigIII.3	Symbols compilation window	74
FigIII.4	Symbols compilation window .	75
FigIII.5	Symbols simulator module- Circuit.ext	76
FigIII.6	Onlineplot	78

NOMENCLATURE

e	Effort
f	Flow
F	Force
τ	Torque
V	Voltage
P	Pressure
μ	Chemical potential
h	Enthalpy
e_m	Magneto-motive force
v	Velocity
ω	Angular velocity
i	Current
dQ/dt	Volume flow rate
ds/dt	Entropy change rate
dV/d	Volume change rate
dN/dt	Mole flow rate
dm/dt	Mass flow rate
φ	Magnetic flux
I	Inertial element
C	Complement element
R	Resistive element
0	Effort equation junction
1	Flow equation junction
TF	Transformer
SE	Source of effort
SF	Source of flow
[X]	State vector
[U]	Input vector
[A]	System matrix
[B]	Input matrix
v	Velocity of the car
h	Height of ground excitation
l	Length of ground excitation
a	Distance of rear wheel from C.G
b	Distance of front wheel from C.G
R_{2f}	Front damper
K_{2r}	Rear stiffness
K_{2f}	Front stiffness
Mc	Mass of the half car

R_{2r} Rear damper
Mcg Weight of the half car
R_{1r} Rear damper
R_{1f} Front damper
K_{1r} Rear stiffness
K_{1f} Front stiffness
Jc Moment of inertia of the half car
Mbfg Weight of the hinged suspension arm of the half car
Mbrg Weight of the rear suspension arm of the half car
Mbf Mass of the hinged suspension arm of the half car
Mbr Mass of the real suspension arm of the half car
Jbr Moment of inertia of the rear suspension arm of the half car
Jbf Moment of inertia of the hinged suspension arm of the half car
K_{3f} Front stiffness
K_{3r} Rear stiffness
R_{3f} Front damper
R_{3r} Rear damper
Vr(t) Velocity of the car at rear suspension arm
Vf(t) Velocity of the car at hinged suspension arm

ABBREVIATIONS

VOAS-2	Volvo Optimized Air Suspension-2
DOF	Degree Of Freedom
ARR	Analytical Reduction Relationships
FDI	Falt Detection and Isolation
ICE	Internal Combustion Engine
ESL	Enginering System Laboratory
LQR	Linear Quadratic Regulator
GUI	Graphical User Interface
BG	Bond Graph
MB	Model Builder
UDMT	User Define Model Type
MDI	Multiple Doucument Interfaces
ODE	Ordinary Differantial Equation
CSS	Complex System Simulator
DDL	Dynamic Link Library

CHAPTER 1

INTRODUCTION

Improvements in automobile technologies are continuing at ever increasing pace with the possibilities of putting more and more intelligence in the vehicles. The demand of making the automobiles safer and environmentally friendly is a universal one. Towards this research in the suspension systems is focused around improving the ride comfort and safety while the vehicle takes a turn, is braked or accelerated or passes over irregular surfaces.

Much of the improvements are being obtained through the use of sophisticated computer based controls along with more elaborate use of sensors. These invariably require better models for the automobiles and often models that can be executed in real time on computer for control purposes.

For evaluating ride quality and control of the vehicles quarter car models are often enough but generally half car models are used for better results, as the heave and rocking motion can be kept in focus.

There has been a trend to model the vehicle dynamics using Bondgraph based techniques as it allows easy up – gradation and merger with the development of control systems for the vehicles.

The present work deals with the modeling of a half car using Bondgraph techniques for heave and rocking motions as it goes over a ramp. Two different models have been developed. The first one is based on conventional suspension, well reported in literature. The results obtained are compared with the published literature. This validates the basic modeling approach adopted.

Taking the ideas from this a second model is developed based on a hinged arm suspension, which could be suitable for active control systems. For an active control system the additional spring and dashpot may be replaced with the actuators.

Chapter 2 deals with the literature on the topic. A variety of papers have been reviewed discussing the issues of independent suspension, full and half car models, used of Bondgraphs for modeling the suspension systems etc.

Chapter 3 deals with the actual development of the models using a Bondgraph package- SYMBOL: SHAKTI. The first model deals with conventional half car model, with a linearised proposition for the tire stiffness and assumption that the vehicle frame is a rigid body. The Bondgraph created are loaded with data from available sources and the results are verified. Additional results are also obtained from the model.

The second model deals with a half car suspension through a hinged arm with a provision for providing active controllers. However, in the simulations another set of springs and dashpots have been used. This method is developed as in improvement

over the first model with the additional of the hinged arm, with additional springs and dashpots.
Some results have been obtained for the vehicle performance as the vehicle goes over a ramp, using the vehicle data available in literature. Essentially it shows that how models can be easily evolved once the base model is prepared.

Chapter 4 deals with the Results and Discussions and shows the usefulness of the techniques adopted.

It also discusses the possible areas of future work related to the present one.

CHAPTER 2

LITERATURE REVIEW

The literature review has been studied essentially from the point of view modeling, Bond graph based modeling of vehicles.

2.1 SCOPE OF MODELING AND SIMULATION

An overview of simulation modeling and analysis has been provided by Andradottir et al [1] many critical questions are answered in the paper. What is modeling? What is simulation? What is simulation modeling and analysis? What types of problems are suitable for simulation? How to select simulation software? What are the benefits and pitfalls in modeling and simulation? This has been adapted as a model for studies.

2.2 THE DEVELOPMENT OF VEHICLE STABILITYCONTROL

Tseng and Ashrafi [2] established what are realistic subjects encountered in the challenge of achieving technology improvement in a vehicle stability control systems. Various approaches to design and development of automotive systems have been reviewed. They include driver intent recognition, vehicle status measurement and estimation, control target generation, system actuation efficiency and smoothness, road bank angle detection, system development and evaluation, and fault detection. The cycle of the operations is shown in Fig. 2.1,

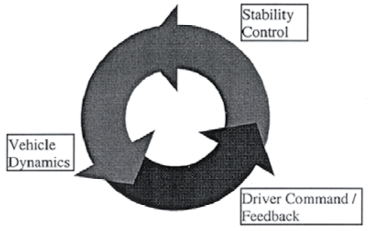

Fig. 2.1 Driver/system/vehicle interaction

2.3 DYNAMIC MODEL USING BOND GRAPGHS

The need for having a comprehensive model with the components of an automobile properly represented has been established by Louca et al [3]. These include the engine, the drive train and vehicle drive systems. The model is developed with the assumptions that it will be used for a wide range of excitations and therefore, all possible complexity's included in the model, e.g. Drive train flexibility and the large rigid body motions. The Bond graph formulation is used for model development because it facilitates the integration of component/subsystem models, provides the user with physical insight, and allows easy manipulation of models. The Engine model is of a steady state torque generator. The drive train consists of the torque converter, transmission and driveline. A nonlinear planar model of the vehicle is used to predict the dynamics in the longitudinal, heave and pitch degrees of freedom. An international delivery truck is implemented in the modeling and simulation environment. The integrated vehicle simulation is validated against transient data measured on the proving ground. An energy based model reduction methodology is applied in order to produce proper vehicle models that provide more design insight. This provides a systematic approach to address the modeling assumptions and generate reduced models that are valid under specific scenarios. The reduced model e.g. Fig.2.3, produces results very similar to the full (baseline) model. In addition to its predictive quality, the utility of the reduced model to study tradeoffs involved in redesigning components and control strategies for improved performance of the vehicle system is demonstrated.

2.4 TRAILING ARM SUSPENSION FOR HEAVY TRUCKS

Glass [4] has dealt an experimental evaluation of a prototype trailing-arm suspension for heavy trucks. Volvo Optimized Air Suspension-2 (VOAS-2), the prototype trailing-arm suspension shown in Fig.2.2, The vehicle was assembled this way to allow kinematics testing of both possible designs (especially roll stiffness) without removing the vehicle from the test facility. The trailing-arm spring is constrained by a bushing that has very low rotational friction and reduces the suspension’s ability to dissipate energy without an additional viscous damper. This aspect of VOAS-2 is quite desirable from the standpoint of controlling suspension damping, and for providing better ride and harshness characteristics due to the suspension.

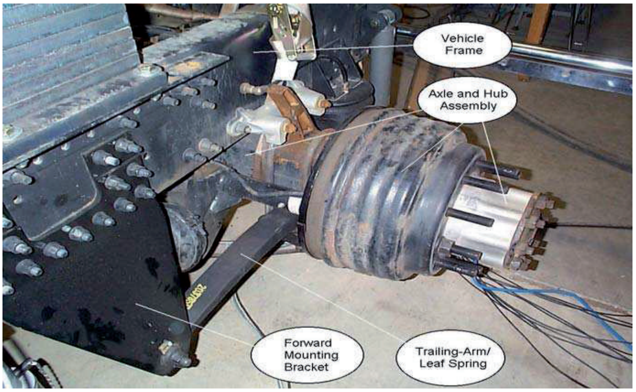

Fig. 2.2 VOAS-2 Suspension on test vehicle.

2.5 TARGET CASCADING

Kim et al [5] established that Target cascading in product development is a systematic effort to propagate the desired top-level system design targets to appropriate specifications for subsystems and components in a consistent and efficient manner. If analysis models are available to represent the consequences of the relevant design decisions, analytical target cascading can be formalized as a hierarchical multilevel optimization problem. The article demonstrates this complex modeling and solution process in the chassis design of a sport-utility vehicle. Ride quality and handling targets are cascaded down to systems and subsystems utilizing suspension, tire, and spring analysis models. Potential incompatibilities among targets and constraints throughout the entire system can be uncovered and the trade-offs involved in achieving system targets under different design scenarios can be quantified.

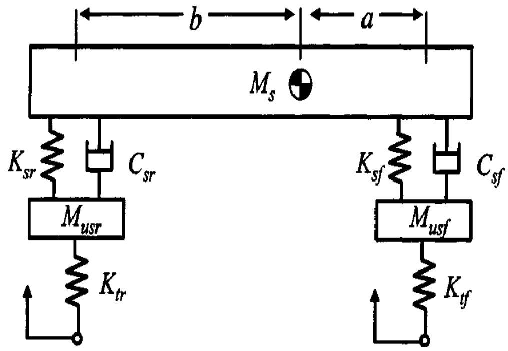

Fig.2.3 Half-car model

2.6 HYDRAULIC SYSTEM DESIGN AND DYNAMIC MODELING

Kim et al [6] describe a hydraulic system design and vehicle dynamic modeling for development of tire roller traction, an essential aspect in the system analysis of tire rollers. Generally, tire rollers are one of the most useful types of machines employed in road construction, technically applied to many construction fields. First to be conceptualized is the new hydraulic driving system and the motion equations for dynamic and hydraulic analysis. First, we design the hydraulic circuit is designed for the steering control and the driving machine system; It can be employed to advance the performance of the lateral control and creating a prototype of construction equipment. Second the hydraulic steering system model and hydraulic driving system model through tire roller system are formulated. Finally, the acquired performance results in actual tire roller equipment using the data acquisition system are validated. These results may perhaps facilitate the establishment of priorities and design strategies for incremental introduction of tire roller technology into the vehicle and construction field.

2.7 MODELING OF AN AUTOMOBILE SUSPENSION

Maxim and Nguyen [7] have dealt the Modeling suspension of an automobile is of interest for many automotive and vibrations engineers. Of importance for these engineers are the ride qualities of the vehicle traversing over broken roads and control of body motion. When traveling over rough terrain, the vehicle exhibits bounce (up and down), pitch (rotation about the center of gravity along the vehicle's length) and roll (rotation about the center of gravity along the vehicle's width) motions. For the modeling, it will be assumed that the vehicle is a rigid body with a suspension that will be modeled as a two-degree-of-freedom (DOF) system. The suspension will consist of equivalent springs in which the stiffness of the tire and the spring are combined and equivalent dampers that account for the shock absorber and the damping of the tire. Modeling of an automobile suspension spring- mass- damper model shown in Fig. 2.4,

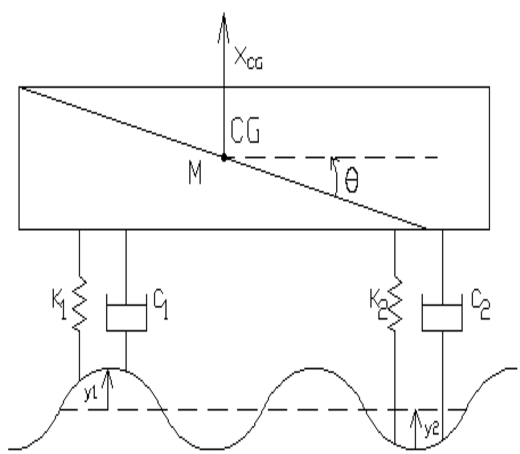

Fig. 2.4 Spring-mass-damper model

2.8 SYSTEMATIC IDENTIFICATION OF DYNAMIC SYSTEM

Rideout et al [8] propose a technique to quantitatively and systematically search for decoupling among elements of a dynamic system model, and to partition models in which decoupling is found. The method can validate simplifying assumptions based on decoupling, determine when decoupling breaks down due to changes in system parameters or inputs, and indicate required model changes. A high-fidelity model is first generated using the bond graph formalism. The relative contributions of the terms of the generalized Kirchoff loop and node equations are computed by calculating and comparing a measure of their power flow. Negligible aggregate bond power at a constraint equation node indicates an unnecessary term, which is then removed from the model by replacing the associated bond by a modulated source of generalized effort or flow. If replacement of all low-power bonds creates separate bond graphs that are joined by modulating signals, then the model can be partitioned into driving and driven subsystems. The partitions are smaller than the original model, have lower-dimension design variable vectors, and can be simulated separately or in parallel. The partitioning algorithm can be employed alongside existing automated modeling techniques to facilitate efficient, accurate simulation-based design of dynamic systems.

2.9 VEHICLE DYNAMICS AND DESIGN

Granda [9] established the basic theoretical principles in vehicle dynamics and design combined with a practical approach-using computer aided techniques, that allows students to build and analyze vehicle dynamics and mechatronics systems used in vehicles using computer models for analysis and design. The article covers a variety of topics such as study of tires, drive train and gear boxes in ground vehicles, kinematics of linkages for analysis of position, velocity and acceleration in two and three dimensions with applications to mechanisms, suspensions and steering mechanisms, vehicle dynamics using multibody systems in three dimensions and computer models of vehicles using solid models and dynamic models. A typical vehicle system model used is shown in Fig. 2.5,

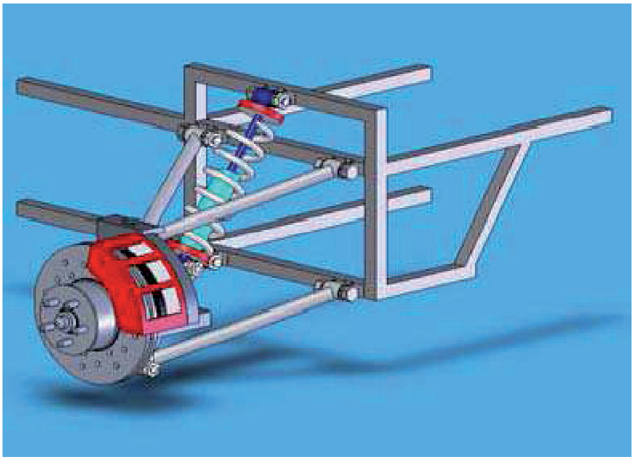

Fig. 2.5 Vehicle systems models

The objective is to provide the student with analytical and computer skills that will allow him to analyze and design two and three dimensional components and entire working Assemblies, and to train students so that they can acquire the ability to perform in kinematic and kinetic dynamic analysis, Finite Element Analysis, time and frequency domain analysis etc.

2.10 MODELING AND SIMULATION OF VEHICULAR POWER SYSTEMS

Zoroofi [10] established that due to the limitations in the availability of fossil fuels and the high consumption rate of this energy for transportation, inclination of vehicle industry toward other sources of energy is inevitable. Electric vehicles and hybrid vehicles could be a good solution. Thanks to the state of art electric motors, power electronics, embedded power train controllers, energy storage systems like batteries and ultra capacitors, the performance of the vehicle could become more and more energy efficient. Since the integrating of all these components in a drive train configuration could be a challenge for the manufacturer, computer simulation and modeling before prototyping could be really beneficial in terms of cost, Safety and design performance.

It is shown that modeling and simulation could be really helpful in design process. The battery model was verified with measurements and is proved to have a good accuracy. An example of modeling longitudinal forces is shown in the Fig. 2.6,

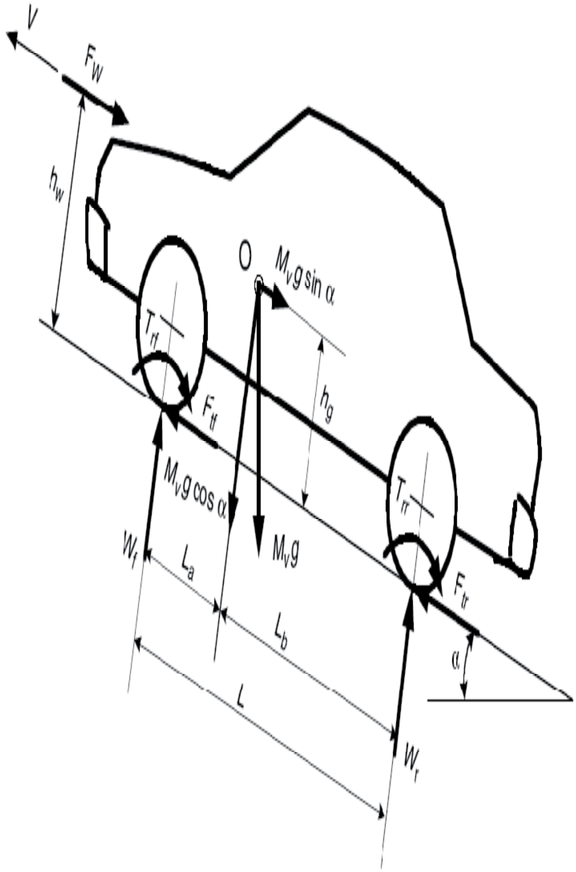

Fig. 2.6 Vehicle longitudinal forces

2.11 FOUR WHEEL VEHICLE WITH PASSIVE SUSPENSION

The paper by Silva et al [11] applies the model-based analytical reduction relationships (ARR) technique, implemented on the so-called Diagnostic Bond Graph, to the problem of detecting and isolating faults in vehicle suspensions. Fourteen degrees of freedom (DOF), four-wheeled vehicle Bond Graph model adapted from models available in the literature is used as starting point. The main contribution of the paper is the proposition of a simplified Diagnostic Bond Graph that, on the ground of the chosen measurements, allows solving the fault detection and isolation (FDI) problem on a reduced subsystem decoupled from the wheel dynamics. This renders unnecessary using the complex and uncertain ground-tire interaction model. The simulation results presented illustrate the methods ability of monitoring and isolating all the possible suspension faults considered. A full car suspension model used is shown in Fig. 2.7,

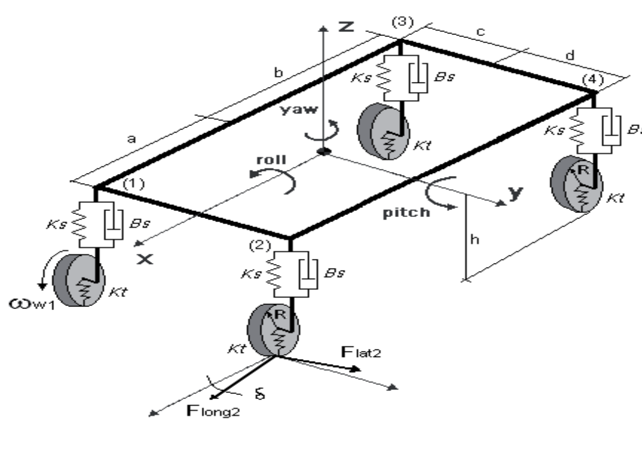

Fig. 2.7 Full car suspension model

2.12 A FULL CAR MODEL WITH ACTIVE SUSPENSION

Active suspension systems e.g. due to Adibi and Rideout [12] have dealt at the subject of significant study over the last two decades. Currently, many active suspension systems can be found on commercially-available automobiles. This paper shows the benefits of the bond graph modeling method to simulate the ability of an active suspension system to improve ride and handling. The active versus passive suspension system is evaluated based on a linearized full car model with seven degrees of freedom. The results show a significant reduction in bounce and pitch acceleration as well as some improvement in roll acceleration of the body for both deterministic and random road profiles. Simulations are performed using commercial software that allows hybrid bond graph and block diagram models. Model construction, simulation, control design and evaluation can therefore be done easily in one software environment

2.13 MODELING AND SIMULATION OF HYBRID ELECTRIC VEHICLE

Milner et al [13] have presented The U.S. Army (TACOM-TARDEC) developed and validated high-fidelity six-degree-of-freedom model to be used in a study for the development of a prototype autonomous vehicle. The model captures realistic dynamics of the six-wheeled, skid-steered vehicle along with the electrical, thermal, and mechanical response of a detailed series hybrid-electric power system with in-hub drive motors, lithium-ion battery, and generator linked to a diesel engine. These components were modeled and integrated via extensive power and energy component libraries developed for use with a high-fidelity software tool for dynamics modeling. Further, the vehicle model's entire complement of components was integrated in a flexible configuration that allowed them to be readily adjusted or swapped out so the user could use the model to ascertain the relative effects of modifying the vehicle's structural or power system components on specific vehicle evaluation criteria. Such criteria include the vehicle's performance with high-speed stability, skid steering stability, body pitch/roll/dive/squat characteristics, braking capability, road/soft-soil traversal, and steering maneuverability.

The model captures both the on- and off-road mobility for the vehicle via use of an extensive library of various terrains including hard surface, sand, sandy loam, clay soil, and snow. Further, detailed waypoint-based path navigation routines automate the vehicle's traversal over a number of user-selectable courses including some established military courses such as Churchville-B, Perryman 1, 3, and A, and Munson with user-defined vehicle velocities. The model functions as an executable file run independent of any proprietary or close-source software; the user utilizes a simplified interface to vary any of the variables associated with the vehicle's geometry, power system, course and speed to navigate, and terrain type applied to the course. The graphical view for the vehicle traversing the selected terrain is shown with an open source 3D graphics tool. The vehicle model was designed primarily for the trade study for the design of a specific vehicle, but was created with sufficient flexibility and capability for modeling future vehicles as well.

The interchangeability of the vehicle models' components and environments allow a user to modify or replace the vehicle's power system components, chassis masses, tires, transmission, duty cycles, courses to traverse, and many other aspects of the vehicle. Thus the user can essentially model any vehicle with similar types of components or structures and use that model to determine the impact of those elements upon many vehicle design considerations such as mass requirements, volume constraints, power system requirements, wheels design, suspension characteristics, and controls. Several new vehicle models are already being developed using this model's flexibility and capability.

2.14. DYNAMIC MODELING AND SIMULATION OF ELECTRIC VEHICLES

Gauchia and Sanz [14] observed that the current energy scenery is continued to be dominated by fossil fuels, especially oil. This dependency is turning critical due to the reducing reserves, uncertain oil resources, and political and economical ramifications of a concentration of fossil fuel reserves in a limited number of regions. The transportation sector is especially affected by this situation and needs to develop new energy vectors and systems to reduce the oil dependency whilst attending to environmental issues. Therefore, vehicle manufacturers are turning to hybrid and electric vehicles. Hybrid vehicles combine an internal combustion engine (ICE) with energy storage systems, which allows reducing the installed power of the ICE, and consequently the fuel consumption and pollutant emissions. With this power train, the user is capable of driving in a pure electric mode, through the energy storage system (normally batteries), or in a hybrid mode with both ICE and storage for more challenging driving cycles.

Electric vehicles are especially interesting due to the exclusion of the ICE, which reduces to zero the emissions on the road and presents a higher efficiency of the power train and environmentally friendly operation. However, even if these reasons are activating its interest, there are several drawbacks which should be solved before reaching a mass production scale. Some of these issues include the development of energy technologies able to guarantee an adequate vehicle range, attractive power ratings and safe, simple and fast recharge. Nowadays there is no electric energy storage technology which can exhibit both high energy and power densities, necessary to meet range and accelerating requirements. Therefore, there is an intensive research to develop new materials for electrochemical energy devices and to hybridize electrochemical energy systems to reach the necessary power and energy specifications. The most popular technologies are Ni-Mn and Li-based batteries, which present higher energy densities than classic Pb-acid batteries. However, these technologies cannot achieve the range obtained with fossil fuels. Therefore, other energy systems, such as fuel cells or flow batteries are being studied as part of a hybrid electric vehicle power train. Finally, this energy system research should be done taking into account the particular situation of transportation, where the weight, volume and cost of the systems included are relevant for a successful and massive use of the electric vehicle. To carry out this research in the final application stage of electrochemical systems, it is necessary to be able to test, model and simulate this system in real operating conditions.

2.15 DEVELOPMENT OF A FULL CAR MODEL

Creed et al [15] discuss the creation of a full car model shown in Fig. 2.8, for a standard road going vehicle. This model has been equipped with suspension force actuators to allow for the future development of an active suspension control system to improve the vehicle's ride comfort. These types of systems are becoming increasingly common on both passenger and commercial vehicles. The flexibility of these systems offer allows them to be specifically tuned for performance or comfort, making them optimum for many applications. Active suspension is concerned with controlling the vertical movements of the vehicle in response to the road inputs to each of the wheels. This is accomplished by actively applying vertical forces in the suspension to counteract some of the effects of the road surface. As a result, these systems can be used to minimize vehicle body roll, vertical accelerations experienced by the passengers, and improve overall vehicle handling.

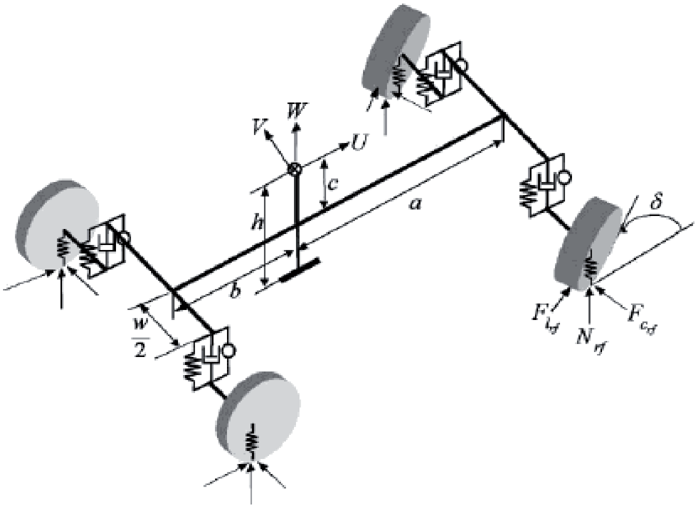

Fig. 2.8 Full car model

2.16 A VEHICLE FOR DELIVERING LABORATORY EXPERIENCE

Lyons et al [16] established that the practice of mechanical engineering requires the ability to investigate and analyze complex thermal and mechanical systems. An effective way for the students to develop their understanding of mechanical engineering systems is for them to get hands-on experience by working in small groups in a laboratory environment. This paper describes a plan to develop a unique capstone laboratory course that provides this experience. The course, Engineering System Laboratory (ESL) will be based upon an integrated sequence of laboratory experiments on an automobile and its subsystems. The automobile is chosen as the system to study because it is compact, relatively inexpensive and is in the direct realm of experience of most students. More importantly, its many complex subsystems provide opportunities for the students to apply the spectrum of their mechanical engineering knowledge, including the principles of mechanics, dynamics, thermodynamics, heat transfer, and controls.

2.17 A HALF CAR ACTIVE SUSPENSION MODEL

Wakeham and Rideout [17] investigate the appropriate level of model complexity when designing optimal vehicle active suspension controllers using the Linear Quadratic Regulator (LQR) method. The LQR method requires the formulation of a performance index with weighting factors to penalize the three competing objectives in suspension design: suspension travel (rattle space), sprung mass acceleration (ride quality) and tire deflection (road holding). The optimal control gains are determined from the solution of a matrix Riccati equation with dimension equal to the number of state variables in the model. A quarter car model with four states thus poses a far less onerous formulation problem than a half or full car model with eight or more states. However, half and full car models are often assumed to be more accurate than quarter car models, and necessary for capturing and controlling degrees of freedom such as pitch and roll motion which are not directly available from a quarter car. The vertical acceleration, pitch acceleration and road holding of a pitch plane vehicle are controlled in this paper using both quarter and half car-based controllers. First, optimal gains are calculated for each of the front and rear actuators assuming that the front and rear of the vehicle can be separately modeled as quarter cars with four states each. Then, half car-based optimal gains, based on feedback of eight states for the entire vehicle, are computed. Using quarter car-based controllers at the front and rear of a half car gives superior performance in reducing sprung mass inertial acceleration, and can effectively control pitch motion even when interactions between front and rear suspensions are not decoupled. Minimizing vertical motion of the front and rear ends indirectly regulates pitch motion. Improvements resulting from the additional complexity of the half car-based controller are seen only when the weighting factor for pitch suppression is very high in the performance index.

2.18 BOND GRAPH MODELING AND SIMULATION SOFTWARE

2.18.1 Symbolshakti

The review has of various software packages is based on the publication Mukherjee and Samantray (2006). Symbol Shakti is an object oriented hierarchical hybrid modeling, simulation and control analysis software. It allows users to create models using bond graph, block-diagram and equation models. Large number of advanced sub-models called Capsules is available for different engineering and modeling domains. It automatically derives completely reduced system equations. Differential causalities and algebraic loops are solved using its powerful symbolic solution engine. It creates high-level C language code and allows any external code embedding. The simulator module has both online and post-display facilities. Event handlers and online event notification and variation of parameters are key features of this advanced simulator. It has a well-developed controls module that automatically transforms state-space modules from BG or block diagram models and converts them to analog or digital transfer functions. Most control and high-level control analysis can be performed.

This software uses a contemporary GUI. It's symbolic and numeric solution capabilities are very advanced. The iconic modeling facility allows system-morphic model layout. The event handlers are a bonus. It also has many post-processing facilities over the simulated results. This software requires pre-installed Microsoft Developer Studio (version 5.0 or above). The direct C++ compilation allows easy integration of external code. The controls module incorporates advanced state space, analog, digital routines for various conversions, filters, feedback systems. The controls module handles matrices, transfer functions, quadruples and numeric data equally. This software is recommended by the developers for use in research and industrial modeling of large systems etc.

2.18.2 Enport

This is the first bond graph modeling and simulation software written in the early seventies by Hales and Rosenberg (2001). This software did not request causalities to be specified, and it transformed the topological input description into a branch admittance matrix which could then be solved. Consequently, Enport is able to handle structurally singular problems. The current version of the code, Enport-7 offers an alphanumerical topological input language and a menu-driven graphical input. Enport-7 runs on various mainframe computers, but a slightly reduced version, Enport/PC, exists for IBM PC's and compatibles.

There are a lot of sentimental values attached with this product. This software was instrumental in growth of bond graph theory and applications. The new release model builder (MB) supports hierarchical model structuring by defining subsystem components that can contain other components; and it has display properties, such as an icon, that can be used in a graphical modeling environment. It structures equations for solution with MATLAB. User-defined model types (UDMTs), which are generalized model definition based on multiport templates that can be specialized for particular purposes, are the latest improvements to this product. The multiple document interfaces (MDI) architecture makes it a suitable editing and development environment. However, the product is not available in a commercial scale and doesn't confirm to cutting-edge software development practices.

2.18.3 Archer

Figueiredo et al (2008) has dealt it is a product of The "Laboratories d'Automatique et d'Informatique Industrielle de Lille" (L.A.I.L.) at the Ecole Centrale de Lille. The bond graph group (BG-group) of LAIL is a research structure devoted to the development and application of the bond graph theory to create physically consistent modeling of a wide class of engineering and life-sciences systems. The BG-Group has developed the modeling software (Archer) allowing a structured and graphical development of engineering models. This group has also developed advanced methods for fault diagnosis and structural analysis using bond graphs.

The software is written in VB and C++, is object oriented and structured. It is yet not commercially available. The software allows determination of structural controllability, observability and invertibily of linear models. It is a high quality academic work based on the research at the "Ecole Centrale de Lille" catering mostly to automatic control theory. The user interface lacks modern features. Numerical simulation and control systems analysis are not within the scope of this product.

2.18.4 Camp-G

Camp-G software package adopted by Granda (1997) that helps engineers and scientists design Mechatronics and Dynamic Systems using as input physical models described by the method of Bond Graphs. Mechanical, electrical, hydraulic, thermal and control systems can be modeled together using computer graphics, Camp-G is a model generating tool that interfaces with Languages such as MATLAB / SIMULINK, ACSL® and others to perform computer simulations of physical and control systems.

A Preprocessor, based on a good GUI, doesn't support object based modeling. Equations derived are neither completely reduced nor sorted properly. It heavily depends on external software to perform post-processing. In the post-processing domain, relation to the base bond graph model is lost amidst mathematical abstractions.

2.18.5 20-Sim

Broenink (1995) has dealt 20-Sim is a modeling and simulation program that runs under windows. It is an advanced modeling and simulation package for dynamic systems that supports iconic diagrams, bond graphs, block diagrams, equation models or any combination of these. With it one can simulate the behavior of dynamic systems, such as electrical, mechanical and hydraulic systems or any combination of these. The latest release is enabled and allows interaction with SIMULINK/MATLAB.

The product is a time-tested modeling tool evolved after the famous TutSim software. The sub-model facilities are big bonus. It supports hierarchical modeling, but sticks to an out-dated PDMT (Pre-Defined Model Type) object implementation. Doesn't require external compilers or any other post-processing software. Control systems analysis module is present only in form of simulation and some basic frequency domain charts. The object property and equation description language doesn’t confirm to any current day programming languages like Pascal, FORTRAN, C, or C++. Overall, it is a good product recommended for modeling of small to medium sized systems. The graphics and hard copy output quality is poor. Use of non-standard menu and toolbar systems, difficult to access library windows and SDI (Single document interface) architecture make overall model creation very tedious.

2.18.6 Pasion 32

An object-oriented simulation tool for discrete, continuous and combined models has been extensively investigated by Raczynski (2000). It supports ODE, signal-flow graphs, bond graphs, queuing models and animation. It is low-cost simulation software for discrete event and continuous system simulation, queuing models, bond graphs, signal flow graphs, animation in 3D scenarios, training and more. The Bond Graph model is created on the screen using a menu-driven easy-to-use graphical editor BONDW. No causalities are needed. BGS can verify the user causalities or impose its own causalities on the user model. Non-linear dependencies can be used describing source nodes or special user-defined non-linear graphs. BGSW (Bond Graph Simulator) generates a set of differential equations for a given model. These equations are used to automatically generate the corresponding Pasion code. It then invokes the solver module DIFEQ for simulation.

A product from Stanislaw Raczynski who is an editor of open directory project on scientific – simulation software, this product lives up to reputations of its author. It uses an object-oriented, Pascal-related simulation language. The language has a clear process/event structure. Its translator generates Pascal source code that can be run using a Pascal compiler. The software computes transient process simulation as well as frequency response simulation. State events, discrete and continuous objects can run concurrently. Various frequently used processes are available in library form. Also includes a post mortem analyzer for stochastic (discrete and continuous) models. The complex system simulator (CSS) is very useful for combined systems, permits model coupling. Sub models of different types (queuing, continuous, etc.) can run concurrently in the same simulation program. Hierarchical model building features are not very advanced in bond graph domain. The GUI features are rather rudimentary. Recommended for class-room and tutorial purpose use by students and researchers.

2.18.7 Bondlab

BondLab is a design environment which has the aim to facilitate and optimize the design cycle of mechatronics products by Minten et al (1997). BondLab is developed as a platform independent MATLAB toolbox. It integrates seamlessly the behavioral modeling with other tasks in a mechatronic design cycle. It has an easy to use model entry graphical interface and a unified treatment of both linear and nonlinear models. Several smooth transitions between the behavioral and causal model descriptions are available non linear parametric ODE description in MATLAB function format, Transfer Function (symbolic and numerical), non linear state space (symbolic and numerical), parametric Simulink2® .mdl block diagram format, contains direct simulation, visualization and animation facilities.

The GUI confirms to standard software practices. The editing tools are ergonomically placed. It is based on single document interface (SDI) architecture in Win 32, which makes model building a bit difficult. A bond graph animation support is available. Overall, this product has been rated as average software.

2.18.8 Cambas

In this automated modeling software using BGs has been heen adopted in literature by Moore and Frumkin (2012) from University of Michigan, System components are represented by icons called templates, which have a fixed number of ports by which they can be interconnected. Expandable (variable complexity) bond graph models are used to represent the detailed model contained in each component template. Cambas allows the design engineer to simply select and arrange the icons (templates) containing the expandable model to build a 'word bond graph like' representation that matches the configuration of the system to be modeled (designed). Cambas then automatically deduces the Proper Model (global system bond graph) by searching through the component templates based on the eigen value structure of the system. The software includes four major components, namely, bond graph processor, system synthesizer, equation generator, and eigen value solver. The parameters (mass, stiffness, damping, diameter, etc.) of each component are entered using the Parameters tool. The bond graph of any component can be displayed using the Expand tool.

Cambas is developed using the C programming language and the OSF/Motif graphics commands for risc workstations. This software facilitates the development of proper models (simplest model with physically meaningful parameters and variables) using a two-level, Template-Based Modeling Approach. The proper model is detected when all the system eigen values within a user specified frequency range of interest are found. Additional accuracy criteria, which require all of the critical system eigen value to converge to a user specified tolerance, can also be used. This product aims at generating proper models during the early stages of the design, shortening the design cycle and boosting the dynamic performance. Applicable to only linear systems, simulation and control analysis (besides eigen value inspection) are out of its scope. It is a freeware available from Automotive Research center, University of Michigan.

2.18.9 Dymola

Dymola provides an object oriented modeling using Modelica language by Dynasim (1992). Dymola allows for graphical model composition from library models, continuous/discrete simulation and 3D animation. Bond graph methodology can be used. Model details are given by ordinary differential and algebraic equations, also in a matrix form. Dymola converts the differential-algebraic system of equations symbolically to state-space form if possible. Graph-theoretical algorithms are used to determine which variable to solve for in each equation and to find minimal systems of equations (optionally using tearing) that have to be solved simultaneously (algebraic loops). The equations are then, if possible, solved symbolically. Linear systems of equations can be solved symbolically or numerically. Dymola also supports instantaneous and discontinuous equations. Ready to use model libraries are available in many engineering domains.

Handles large, complex multi-domain models, faster modeling by graphical model composition, symbolic pre-processing, allows user defined model components, 3D Animation and real-time simulation. Huge library modules make it an ideal platform for easy and quick model creation. Uses object oriented modeling language Modelica to support hierarchical structuring, reuse and evolution of large and complex models independent from the application domain. Acausal sub-model creation based on differential and algebraic equations gives flexibility of implementation. Learning Modelica language seems to be the only hitch. There are no major provisions for advanced frequency domain and control system analysis. It is rated a very high as a modeling language, but fails to impress in its bond graph processing capabilities.

2.18.10 Hybridsim

Hybridsim is an implementation of a hybrid (mixed continuous/discrete behavior) bond graph modeling and simulation software has been adopted in literature by Mosterman (2000). It embodies a set of physical principles that govern discontinuous changes in physical system models which violate the continuity of power constraint. It is an experimental modeling and simulation environment to establish a formal framework and serves as a precursor to an object-oriented implementation as part of the Modelica modeling language. This software was designed using IBM Visual Age for Java. It consists of a model editor and two toolboxes, one for bond graph elements and one for block diagram elements. The simulator has an animation facility to study the power distribution over time.

Hybrid bond graphs extend traditional bond graphs by an ideal switching element, the controlled junction. Simulation is based on graph propagation. No explicit system of equations is derived. It supports only ideal bond graph elements and a few block-diagram components for analysis of small linear systems. The software is still in development stage. Free java source code can be downloaded.

2.18.11 Modelica

Modelica is a language designed for multi domain modeling developed by Broenink (2003) a non-profit organization with seat in Linköping, Sweden. Modelica is an object-oriented modeling and simulation tool, influenced by many of the ideas from Dymola. It is suited for multi-domain modeling, for example, mechatronic models in robotics, automotive and aerospace applications involving mechanical, electrical, hydraulic and control subsystems, process oriented applications and generation and distribution of electric power. Models in Modelica are mathematically described by differential, algebraic and discrete equations. Modelica is designed such that available, specialized algorithms can be utilized to enable efficient handling of large models having more than hundred thousand equations. Modelica is suited and used for hardware-in-the-loop simulations and for embedded control systems.

The simulation and graphical editing interface of many software products like Dymola generate and use Modelica code. The last release of Modelica language Version 1.4 was on December15, 2000, by the Modelica Association. Modelica can be used for mixed continuous and discrete models (Hybrid models) as well as for Discrete Event and Discrete Time Models. It can deal with conditional equations with causality changes and generally adopts to a formal a causal (non-causal) modeling scheme at the front-end. It is suitable for modeling of large systems using hierarchical modeling scheme embedded as reusable sub-model classes. The lack of proper GUI drivers and difficulty in linking with other high-level programming languages like C and C++ code.

CHAPTER 3

PRESENT WORK

3.1 INTRODUCTION

The review in the previous chapter shows that modeling of an automobile is an important field of study. Further it shows that new modeling techniques, like the one based on the use of Bond graph is becoming popular, as it helps in several ways, like flexibility and extensibility of models and automatic generation and solution of the system equations etc.

Further, study of literature shows Louca et al [3] that for various purposes like quick evaluation of specific features and configurations and real time running of the models for controls. Simple models are desirable and are adequate, as compared to full models.

Four wheeled automobiles have been modeled in a variety of ways, to study stability controllability, of environmental friendliness and self navigation etc.

Many of these studies show Kim et al [5] that the most common requirement is to have a relatively simple but responsive model at hand which may be run in real- time during the vehicle operation.

Recent studies particularly Granda [9] and Silva et al [11] show that while the full blown vehicle models may be needed to be used in some application a plane two wheeled half car like model and even a single wheel quarter model have a role in evaluating the active ride control system of a vehicles. Further, the controlling system itself may need to run a vehicle model to provide more advanced type of vehicle control,These models may also form a basis of crisis control in the event of any damage of the vehicle.

Keeping the above observation in mind the present work has concentrated on building a half car vehicle model and its validation and evaluation for road un – eveness.

Further, a Bondgraph based approach has been adopted as it easily permits evaluation of variety of parameters and also in principle run time versions if needed can be derived.

3.2 DEVELOPMENT OF THE MODEL

For the proposed Bondgraph based study University developed software called Symbolshakti has been used. Main feature of this software is outlined in Appendices-III and its operation for solving the problem is also shown there.

The concept of Bondgraphs based on power flow in various elements of a system (I,C, R, TF, SE, SF) and the concept of junctions (0,1) is shown in appendices II. If the roll of the vehicle is not important (often the case) the normal 4 wheeled vehicle as mentioned earlier can be reduced to a 2 wheeled model and various parameters can be identified for a front wheel or rear wheel drive vehicles.

Since the role of modeling and simulation in engineering design continues to gain competitive advantage, it is desirable to reduce the time required to move from concept to the final product. All the standard assumptions requiring linearity in tire stiffness rigidity of the body and lack of lateral motion of the tire etc have been adopted. As a model complexity increases in step with advances in computer software and hardware, the engineer remains well served to use "proper models simulation". Proper modeling can be defined as the systematic determination of the model with minimum complexity.

3.3 MODELING THE DYNAMICS OF AN AUTOMOBILE WITH ROAD EXCITATION

3.3.1 Outline of the Plan of Studies

The studies here have been divided into two parts

(a). The first one deals with the modeling of half car with simple spring dashpot suspension on the front end rear wheels as it goes over a road bump, Various responses are studies through a bond graph model.

(b). The second study deals with development of a more elaborate suspension model with a hinged arm between the wheels and the chassis with spring and dash pots on both sides of the pivot.

The first model is relatively standard one and has been studied widely with the simplifying assumptions of rigidity of the chassis and linearity of tire stiffness etc.

The bondgraph developed for it is to be verified with the one reported in Mukherjee et al [18] by selecting the parameters reported there. The rocking and heaving motions of the vehicle have to be obtained and then validated with the reported one.

The half car model is to be used further to study the performance of the vehicle in terms of other parameters as it passes over the bump.

The model of a hinged arm suspension (often used for active control) is to be developed next. The performance of this suspension is to be modeled as the vehicle goes over a bump on the road.

3.3.2 Half Car Models

3.3.2.1 Half Car Model with a Fixed Suspension

The distance between the front and back suspension from C.G. were selected as 'b' and 'a' respectively, The vertical motion of the car is modeled by spring and dashpots suspensions at the front and the back wheels (Fig. 3.1), The vehicle model was given a velocity step input of 1 m/s, lasting for 10 seconds for simulation.

The model allows heave and pitching motion of the vehicle to be studied.

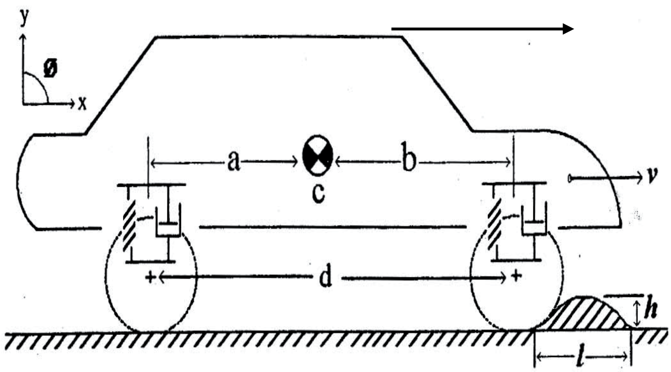

Fig. 3.1 Half car model with a fixed suspension

3.3.2.2 Description of the Elements of the Bondgraph

The description of the elements of the Bondgraph shown in Table 3.1,

Table 3.1 the description of the elements of the Bondgraph

Name of the element	Symbol
Flow equalizing junction	1
Effort equalizing junction	0
Inertial element	I
Compliant element	C
Resistive element	R
Source of effort	SE
Source of flow	SF
Transformer	TF

3.3.2.3 Parameters of a Half Car Model with Fixed Suspension

Parameters of a half car model with fixed suspension as given in Table 3.2,

Table 3.2 Parameters of a half car model with fixed suspension

Description	Parameter name	Values used
Velocity of the half car	v	1 m/s
Height of ground excitation	h	0.1 m
Length of ground excitation	l	0.3 m
Rear damper	REAR_DM	100 n.s/m
Rear stiffness	REAR_ST	20000 n/s
Front damper	FRONT_DM	100 n.s/s
Front stiffness	FRONT_ST	20000 n/s
Mass of the half car	CAR_MASS	1080 kg
Distance of rear wheel from C.G	a	1.1 m
Distance of front wheel from C.G.	b	0.9 m
Moment of inertia of the half car	J_CAR	250 kgm^2

3.3.2.4 Development of the Bondgraph

Bond graph based model have been developed using the notation and ideas from the Mukherjee et al [18]. The properties and the utilities of the junction can be brought about as follows:

In Bondgraph there are only two kinds of junctions, the **1** and the **0** junction. They conserve power and are reversible. They simply represent system topology and '1' junction represents a common mass point in a mechanical system, a series connection (with same current flowing in all elements) in a electrical network and a hydraulic pipeline representing flow continuity, etc. Two such junctions with four bonds are shown in the Fig. 3.2, below.

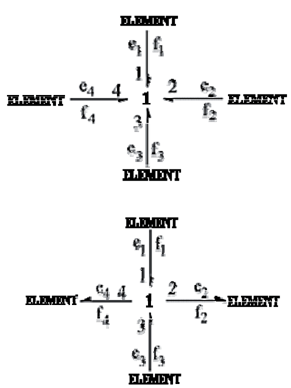

Fig. 3.2 '**1**' Junction element models

Using the inward power sign convention, the constitutive relation (for power conservation at the junctions) for the figure above may be written as follows;

$e_1 f_1 + e_2 f_2 + e_3 f_3 + e_4 f_4 = 0.$

As **1** junction is a flow equalizing junction,

$f_1 = f_2 = f_3 = f_4$.

This leads to, $e_1 + e_2 + e_3 + e_4 = 0$.

Now considering the other bond graph, the constitutive relation becomes,

$e_1 f_1 - e_2 f_2 + e_3 f_3 - e_4 f_4 = 0$, and, $f_1 = f_2 = f_3 = f_4$.

Thus, $e_1 - e_2 + e_3 - e_4 = 0$.

So, a **1** junction is governed by the following rules:

The flows on the bonds attached to a **1**-junction are equal and the algebraic sum of the efforts is zero. The signs in the algebraic sum are determined by the half-arrow directions in a bond graph.

0 - junctions have equality of efforts while the flows sum up to zero, if power orientations are taken positive toward the junction. This junction represents a mechanical, electrical node point and hydraulic pressure distribution point or Pascalian point. Two such junctions with four bonds are shown in the Fig 3.3, below.

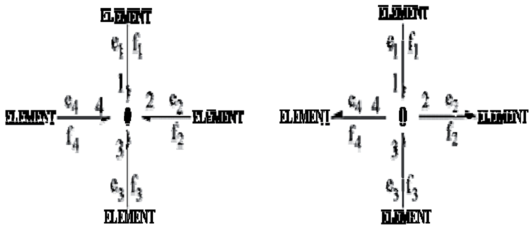

Fig.3.3 ‘**0**’ Junction element models

In case of the model in the left, the constitutive relation becomes,

$e_1 f_1 + e_2 f_2 + e_3 f_3 + e_4 f_4 = 0.$

Whereas, the model in the right is governed by the following relation,

$e_1 f_1 - e_2 f_2 + e_3 f_3 - e_4 f_4 = 0.$

As **0** junction is an effort equalizing junction,

$e_1 = e_2 = e_3 = e_4.$

This leads to, $f_1 + f_2 + f_3 + f_4 = 0$ and $f_1 - f_2 + f_3 - f_4 = 0$, for the left and the right models, respectively.

3.3.2.5 Input Excitation for the Vehicle

A sinusoidal bump has been selected for the vehicle excitation with the following details, Where h is the height (m), of the ground excitation of the bump, l is the length (m), of ground excitation of the bump, v is velocity (m/s), of half car, d is the diameter (mm), of the wheel and

T is time (s).

The bump excitation for front wheel is $y = h * \sin\left(\pi * \frac{v}{l} * t\right)$ for $0 \leq t \leq \frac{1}{v}$

$= 0,$ for $t > \frac{1}{v}$

And for rear wheel is $y = h * \sin\left(\pi * \frac{v}{l} * \left(t - \frac{d}{v}\right)\right)$ for $\frac{d}{v} \leq t \leq \frac{d+l}{v}$

$= 0,$ for $t > \frac{d+l}{v}$

3.3.2.6 Creating the Bondgraph Model

A modeling scheme for the half car model as shown in Fig 3.4, is visualised, using the literature and the Bondgraph logic with two '**0**' junctions and 4 numbers of '**1**' junctions with relevant transformers.

Entering into the Bondgraph Software, Symbol Shakti is carried out through the entry module called Bond Pad. Entry has been started with the typical flow equalizing '**1**' junction structure shown in Fig 3.2. The Software assists in various activities which include numbering, casuality checks etc. For example the inertial element (I17) is depicted as rotational inertia of the vehicle, compliant element (C18) is the suspension spring of the vehicle.

For the rear portion of the vehicle, resistive element (R1) is the suspension damper and the compliant element is (C2).

Effort equalizing junction (0) shows real suspension of the half car, Source of flow (SF4) shows ground excitation of the car of a real wheel, Transformer (TF6) a mass less lever is connected to the rear wheel of the half car, Flow equalizing junction (1) is given the vertical motion of C.G. of the half car, Inertial element (I7) is mass of the vehicle, compliant element (C9) is heave motion of the half car.

Similarly for front suspension of the half car the effort equalizing junction (0) shows the front suspension of the car, Source of flow (SF10) shows ground excitation of the car at the front wheel, Transformer (TF12) a mass less lever connected to the front wheel of the half car, Flow equalizing junction for the front wheel is (1), resistive element is (R13) the front suspension damper, compliant element is (C14).

The efforts on the bonds attached to a 0-junction are equal and the algebraic sum of the flows is zero. The signs in the algebraic sum are determined by the half-arrow directions in a bond graph

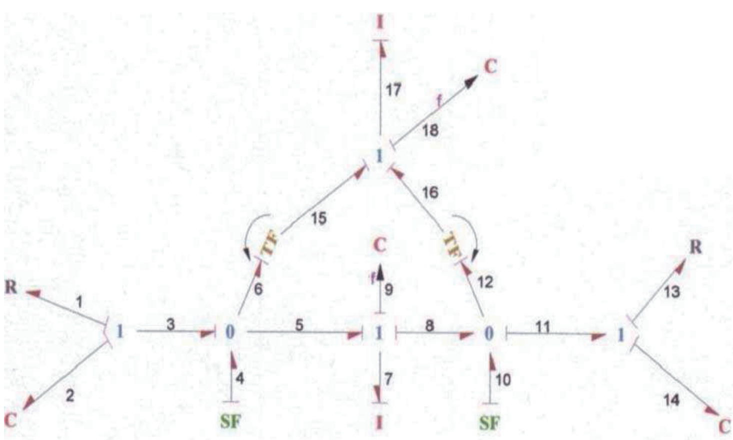

Fig. 3.4 Bondgraph model of half car model with a fixed suspension

3.3.2.7 Running the Model on Symbol Shakti

To run the model is started simulator by pressing simulate item from the process menu. Symbol Shakt simulator window opens with active document. “Parameters” item is choosen from the view menu, A box appears showing all system parameters followed by a small square box to the left of each parameter.Here, model parameter are set, Time range is set by selecting “ Simulation properties” item from the “View” menu.

Simulation time in seconds (say, 10) by typing in the final time edit field on Time group box The Plot block axes are set by selecting “Plot blocks” item from the View menu. Horizontal axis is set “0” for time, and two vertical axes Y [3]:Q4 for vertical “1” (in the display) and Y [2]: Q8 for vertical 2 are selected. Q4 is rocking motion of the half car and Q8 is heave motion of the half car etc.

Fig. 3.5 shows the equations generated for the half car model as obtained through the simulator.They have been automatically solved by the system. Some of the results obtained are shown in the next section.

```
e1=na*(K15*Q15+R16*(SF9+P5/M5-na*P1/M1))+b*(K17*Q17+R18*(SF12+P5/M5-b*P1/M1))
e5=-K15*Q15-R16*(SF9+P5/M5-na*P1/M1)-K17*Q17-R18*(SF12+P5/M5-b*P1/M1)
f8=P5/M5
f4=P1/M1
f17=SF12+P5/M5-b*P1/M1
f15=SF9+P5/M5-na*P1/M1
```

Fig. 3.5 Generation of equation of half car model with fixed suspension

3.3.2.8 Simulation and the Results Obtained

The half car model with fixed suspension has been tested with the parameters shown in the Table 3.2, along with the road bump described in section 3.3.2.5.

Fig.3.6, shows the rocking motion of the half car when the half car is driven at 1 m/s speed. These results match almost exactly with the results for this reported in Mukherjee et al [18].

Fig 3.7 shows the heaving motion of the half car under the same conditions. These results also match the ones reported in Mukherjee et al [18].

These two validate the model and also proper appreciation and the use of the Software Symbol Shakti.

Further studies have been carried out using the Bondgraph Model created.

Motion of only the front wheel is simulated in terms of rocking and heaving as it passes over the bump in terms of time and shown in Fig. 3.8, and Fig. 3.9,

Results show that the behavior of the individual wheels may be significantly different when taken alone, as compared to over all model.

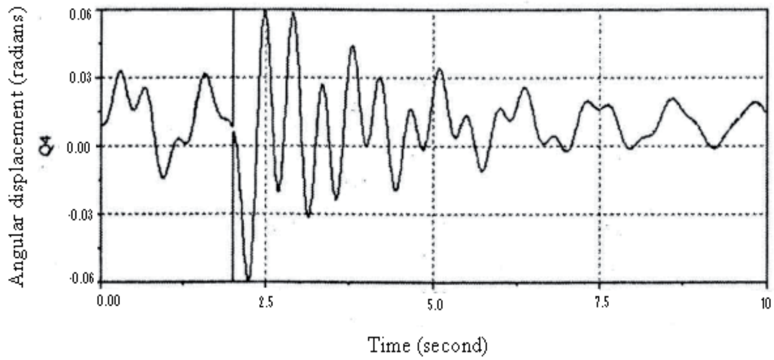

Fig. 3.6 Rocking motion of the half car model at front suspension with vertical angular displacement at C.G., Input parameters to Symbol Shakti, Angular displacement-(Q4 radians), Speed - 1m/s, Time-10 seconds

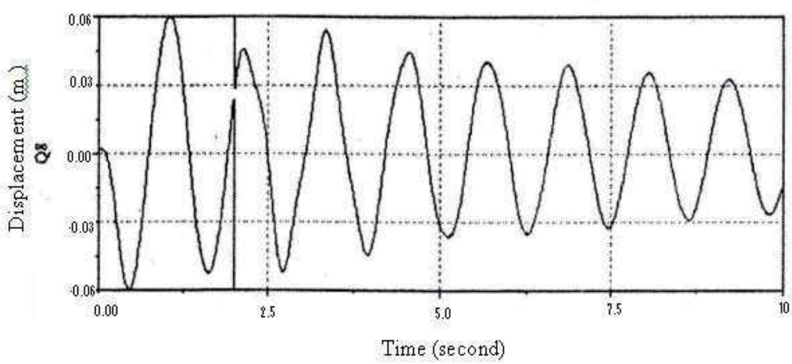

Fig. 3.7 Heaving motion of the half car model at rear suspension with vertical displacement at C.G., Input parameters to Symbol Shakti, Displacement- (Q8 m), Speed - 1m/s, Time -10 seconds

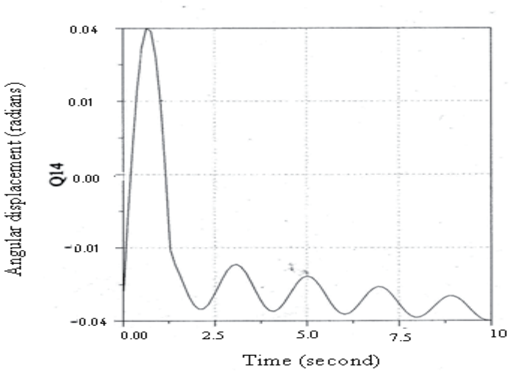

Fig.3.8 Rocking motion of the half car model at front suspension with vertical Angular displacement , Input parameters to Symbol Shakti, Angular displacement- (Q14 radians), Speed - 1m/s,Time- 10 seconds.

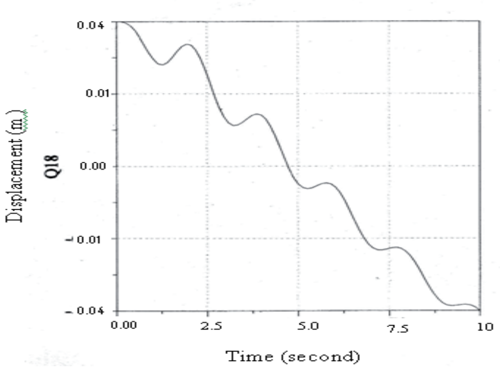

Fig. 3.9 Heaving motion of the half car model at rear suspension with vertical displacement, Input parameters to Symbol Shakti, Displacement- (Q18 m) , Speed - 1m/s, Time – 10 seconds.

3.3.2.9 Validation of Results and Possibilities of further Studies

Validation of the results obtained was carried out (as explained earlier) with the help of results reported in Mukherjee et al [18].

The first two graphs plotted using the data from Mukherjee et al [18] dealing with rocking and heaving motion of the half car identical values.

This validates proper use of the Software and use of correct units and values.

Using the model created has been further used to study the individual behavior of the front and rear suspensions.

The behavior of the suspension for going over a bump can be further studied by changing over the parameters in the model developed.

It has been further evolved to study hinged arm suspension as shown in the next section.

3.3.3 Modeling Hinged Suspension through Bondgraph based Half Car Model

3.3.3.1Introduction:

As discussed in literature review suspensions for automobiles have been continuously advancing and newer configurations permitting on line control have become practical. These require real time computation and thus simpler but roboust models permitting control, where Bondgraph models have become well adopted.

Suspensions have evolved with trailing and leading arms, Glass [4] helping in providing control elements besides affecting the dynamics.

Present effort is about a hinged arm suspension as suggested by Mukherjee et al [18] where basic configuration is of a leading arm (value of lead is a variable). This has to be modeled for a sinusoidal bump (as in the earlier section), and the heave and rocking motion (pitch) have to be studied.

3.3.3.2 Development of the Vehicle Model

The configuration of the vehicle elements adopted is shown in Fig 3.10, which has considerably more elements than the earlier model. Basically two more hinged platforms have been added which are connected to chassis of the vehicle. Sets of springs and dashpots between the hinged arm and the vehicle are part of the proposition Mukherjee et al [18].

Various elements and their nomenclature is also shown in the Fig 3.10, Then three hinged points for the half car also shown.

To simulate the half car, the model was made geometrically symmetric by setting distance (A_{f} + a_{f}) for front wheel and (A_r + a_r) for real wheel effectively placing the models centre of gravity between front and rear wheel of the vehicle. Most of the values for the physical elements of the vehicle have been taken from the vehicle describe in the previous section.

In the proposed model the distance of rear wheel from C.G is 1.1m, the. ,Distance of front wheel from C.G is 0.9 m ,Front damper is R_{2f} , Rear stiffness is K_{2r} , Front stiffness is K_{2f}, Rear damper is R_{2r} ,

Mass of the half car is Mc, Rear damper is R_{1r} , Front damper is R_{1f} , Rear stiffness is K_{1r} , Front stiffness is K_{1f} , Moment of Inertia of the half car is Jc , Weight of the hinged arm suspension of the half car is Mbfg , Weight of the rear arm suspension of the half car is Mbrg , Mass of the hinged arm suspension of the half car is Mbf, Mass of rear arm suspension of the half car is Mbr, Moment of Inertia of the rear arm suspension is Jbr,Moment of Inertia of the hinged arm suspension of the half car is Jbf , Front stiffness is K_{3f} , Rear stiffness is K_{3r} ,Front damper is R_{3f} , Rear damper is R_{3r} , Velocity of the car is v.

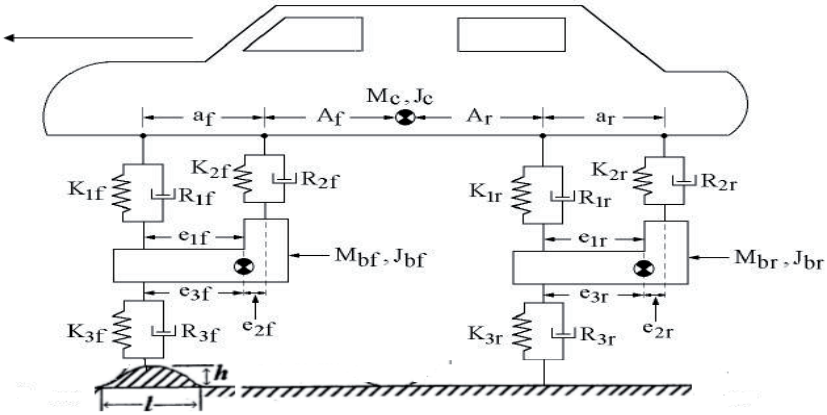

Fig. 3.10 Half car model with hinged suspension

3.3.3.3 Development of the Bondgraph Model

The basic effort towards starting a Bondgraph model is similar to what has been shown for a non hinged suspension vehicle (3.3.2.4).

The details of the road bump have also been adopted from the previous section. For the proposed configuration two additional hinges along with masses and moment of inertias have been introduced and two sets of additional springs and dashpots between the hinged platform and the body of the vehicle have been introduced. To develop a feasible system transformers have been introduced as shown in the Fig 3.11, as the two halves are symmetrical only half of the diagram has to be kept in focus.

3.3.3.4 Development of the Actual Bondgraph

To draw the model of the half car with hinged arm suspension the activities started by invoking the Bond Pad of the Bondgraph software Symbol Shakti.

Towards developing the Bondgraph description (Fig 3.11,) the software itself provides supports for a variety of activities like numbering the Bonds, displaying direction of casuality, power and effort flows etc. Infact the software checks the feasibility of the system modeled by displaying zero errors in end if it is logically correct.

Inspite of this fact, it does not insure that it is Bondgraph of the system that was actually to be modeled.

To gain further confidence some numerical results have also been obtained as shown in the next section.

In the Bondgraph Mc is the mass of the car body and Jc is the Moment of inertia (as shown in the Fig 3.10). Description of bonds and associated elements is as follows.

For the rear suspension compliant element is (C15) rear stiffness is $K_{2r,}$ For the resistive element is (R16) rear damper is R_{2r}, Resistive element is (R27) rear damper is R_{1r}, Compliant element (C30) the stiffness is K_{1r} and also for the resistive element of the rear damper is $R_{1r,}$ Source of effort is (SE54) , Mass of the rear arm suspension of the half car (M53) is Mbr, Moment of Inertia of the rear arm suspension (M57) is Jbr, For the resistive element (R78) rear stiffness is K_{3r} , Source of flow is (SF73) for the velocity of the half car is v.

For the front suspension resistive element (R18) is for the front damper is R_{2f}, Compliant element (C17) is for front stiffness is K_{2f}, Compliant element (C32) is for front stiffness K_{1f} , Source of effort is (SE54), Mass of the hinged arm suspension (M49) is Mbf, Moment of Inertia of the hinged arm suspension (M60) is Jbf, Resistive element (R76) front damper is R_{3f} , Source of flow (SF72) for the velocity of the car is v.

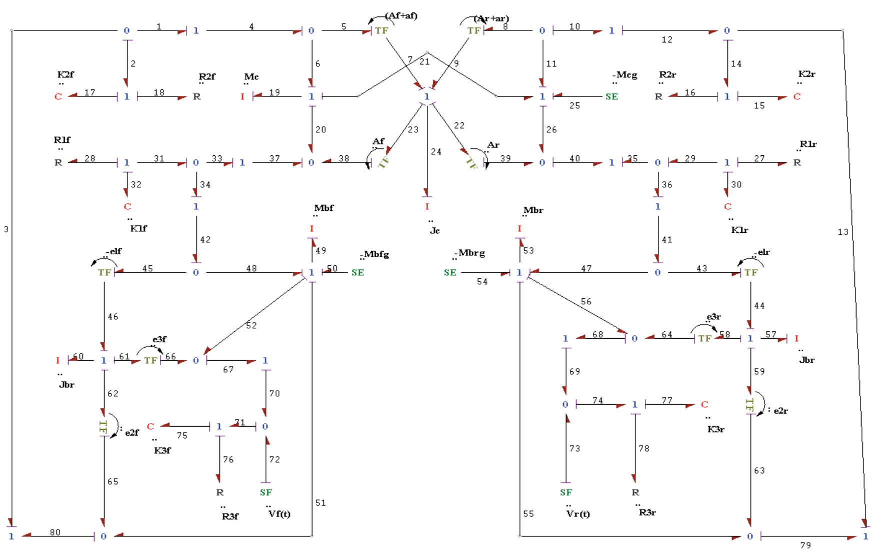

Fig. 3.11 Bondgraph model of half car model with a hinged suspension

3.3.3.5 Running the Model on Symbol Shakti

A model that has been successfully compiled can also be run on the system and results obtained.

Symbol Shakti also allows the provision of obtaining various plots by specifying the relationships.

Activation of the bonds can be done to obtain specific details of the variation of the values.

Equation as generated by the system for the half car model with a hinged suspension are shown in Fig 3.12,

These equations have been automatically solved by the Symbol Shakti solver for specific case as shown in Fig 3.13,

Results show that it the basic characteristics are not very different then the model may have to be tuned to drift towards any specific desirable objectives.

```
Od1=positive*P57/M57
Od2=1/positive*P57/M57
Od3=positive*P60/M60
Od4=positive*P60/M60
Od5=Od4+P49/M49
Od6=Od2+P53/M53
Od7=Od1+P53/M53
Od8=Od3+P49/M49
Od9=negative*P24/M24
Od10=positive*P24/M24
Od11=positive*P24/M24
Od12=negative*P24/M24
Od13=negative*P57/M57
Od14=1/negative*P60/M60
Od15=R76*(SF72+Od5)
Od16=R78*(Od6+SF73)
Od17=Od9+P19/M19
Od18=-Od10-P19/M19
Od19=-P19/M19-Od12
Od20=Od11+P19/M19
Od21=Od13+P53/M53
Od22=Od14+P49/M49
Od23=K75*Q75+Od15
Od24=K77*Q77+Od16
Od25=R16*(Od18-Od7)
Od26=K15*Q15+Od25
Od27=R18*(-Od17-Od8)
Od28=(-Od17-Od8)
Od29=R27*(Od20+Od21)
Od30=R28*(Od19+Od22)
Od31=positive*(-Od26)
Od32=1/positive*Od24
Od33=K17*Q17+Od27
Od34=-Od29-K30*Q30
Od35=-Od30-K32*Q32
Od36=Od34+SE54-(-Od26)-Od24
Od37=positive*(-Od34)
Od38=1/negative*Od35
Od39=positive*(-Od33)
Od40=-SE25-Od26+(-Od34)
Od41=SE50+Od35-(-Od33)-Od23
Od42=negative*Od34-Od32-Od31
Od43=Od38-positive*Od23-Od39
Od44=-Od35+Od33-Od40
Od45=negative*Od33+positive*Od26-Od37-negative*Od35
e60=Od43
e57=Od42
e53=Od36
e49=Od41
e24=Od45
e19=Od44
f77=Od6+SF73
f75=SF72+Od5
f32=Od19+Od22
f30=Od20+Od21
f17=Od28
f15=Od18-Od7
```

Fig 3.12 Generation of equation of half car model with hinged suspension

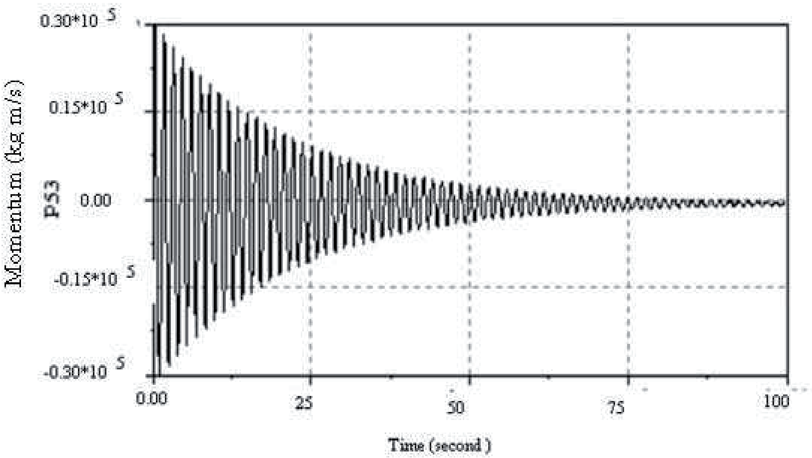

Fig.3.13 Heaving motion of the half car model at rear suspension. Observed through Momentum in bonds, Speed-15 m/s, Time- 100 seconds, Numerical data is taken from Table 3.2,

3.4. RESULTS AND CONCLUSIONS

Bondgraph based modeling of various systems of an automobiles has been carried out in large number of investigations, reported in literature. The present work attempts to model a car suspension through a half car model approach.

A hinged arm suspension selected has been modeled for a case when the vehicle goes over a bump of the shape of a sine wave. Efforts have been concentrated on evolving the model, as once the model is obtained a variety of results can be obtained.

The study has been carried out in two parts. The first part models a half car with a simple suspension. The results of this simulation have been successfully verified with the data available in the literature. The second part is an effort to model a hinged suspension with a half car model as an extension to the first model. A bond graph model for this has been successfully compiled. A few trail results for specific situations have been obtained.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 MODELING

Automobiles continue to be central to present day human activities and a vast literature exists on their evolution and adaptation. However, the pace of developments has not slowed down. Newer tools such as Bondgraphs make it possible to model more and more complex systems.

The efforts here examine the modeling of automobile suspensions, specially that may be useful in future to bring in active control systems etc.

A half car model of a conventional suspension using Bondgraphs has been created first for modeling heave and rocking motion of a car as it passes over a bump on the road. The results are specially obtained for a case where published value is available. A close match with that has validated the modeling strategy and the use of the Bondgraph package.

This model has then been extended to a hinged arm suspension half car model with two sets of springs and dashpots across the hinge. The model has been successfully compiled showing the logical correctness of the model. A few results have also been obtained more results from the model created can be obtained by giving relevant inputs.

4.2 POSSIBILITIES OF FUTURE WORK

Hinged arm suspensions have been around for quite some time. However, they become important for introducing active computerized control of vehicles. The present work has used a model without active elements but two sets of springs and dashpots.

While a successful compilation of the Bondgraph on the software (Symbol Shakti) of the proposed model validates a logically correct modeling, it does not insure that at worthwhile model has been created. It may be verified with further efforts in that direction.
A more systematic activity can be carried out of study the benefit of the hinged arm suspension under various conditions, through the model created.

The work can be extended by bringing in the control systems through the Bondgraphs model towards developing an active suspension model.

REFERENCES

[1] Andradottir,S., Healy, K.L. , Withers, D. H. and Nelson, B. L. , 1997, Introduction to modeling and simulation, winter simulation conference Binghamton, NY 13902-6000, USA.

[2] Tseng Hongtei Eric and Ashrafi Behrouz, 1999, The development of vehicle stability control at ford on Mechtronics. *IEEE/ASME International Journal of transaction on Mechatronics,vol. 4, no. 3, September 1999, pp 223-234*.

[3] Louca, Loucas. S., Jeffrey, L.Stein and Rideout, D. Geoff. , 2000, Generating proper integrated dynamic models for vehicle mobility using a Bond Graph formulation, Automated modeling laboratory G029 W.E. Lay Automotive Lab, Michigan.

[4] Glass, Jeffrey. L., 2001, Experimental evaluation of a trailing- arm suspension for heavy Trucks,M.Tech dissertation, Blackburg, Virginia, May 8, 2001.

[5] Kim, Hyung. Min., Rideout, D.Geoff.,Papalambros, Panos. Y. and. Stein, Geffrey. L., 2003, Analytical target cascading in automotive vehicle design. *ASME International Journal of Mechanical Design, V-125/48, September 2003,pp 481-489.*

[6] Kim Sang-Gyum, Kim Jung-Ha and Lee Woon-Sung, 2003, Hydraulic System design and vehicle dynamic modeling for the development of a tire roller . *International Journal of Control, Automation, and Systems, Vol. 1, No. 4, December, 2003, pp 484-494.*

[7] Maxim Derek and Nguyen Hieu, 2006, Analysis of an automobile suspension, School of Engineering, Grand Valley State University. August, 4, 2006, pp 43-67.

[8] Rideout, D.Geoff, Stein, L. Jeffery, and Louca, Loucas. S., 2007, Systematic identification of decoupling in dynamic system models. *ASME International Journal of dynamic system, measurement and control, V-129/503, July 2007 pp 503-513.*

[9] Granda, Jose. J., 2008, Vehicle dynamics and design, California State University, Sacramento,USA.

[10] Zoroofi Siavash, 2008, Modeling and simulation of vehicular power systems, International master's Program Electric Power Engineering, Chalmers University of Technology, Sweden.

[11] Silva Luis, Delarmelina Diego, Junco Sergio, K. M. Sirdi Nacer and Noura Hassan, 2008, Bond graph based full diagnosis of 4 wheeled vehicles suspension systems of passive suspension, A venue Escadrille Normadie Niemen, 13397 Marseille Cedex 20, France.

[12] Adibi-as Hadi, and Rideout Geoff, 2009, Bond graph modeling and simulation of a full car model with active suspenson, Faculty of Engineering, Memorial University, St. John's NL, Canada.

[13] Milner David, Goodell Jarrett, Smith Wilford, Pozolo Mike and Ueda Jason, 2009,Modeling and simulation of an autonomous hybrid-electric Military vehicle, *World electric vehicle Journal EVS24 Stavanger, Norway, Vol. 3, May 13-16,2009,pp 1-9.*

[14] Gauchia Lucia and Sanz Javier, 2010*, Dynamic modelling and simulation of electrochemical energy systems for electric vehicles, Sciyo, Croatia, ISBN 978-953-307-100-8, September, 2010 pp127-150.*

[15] Creed Ben, Kahawatte Nalaka and Varnhagen, 2010, Development of a full car vehicle dynamics model for use in the design of an active suspension control system, University of California, Davis.

[16] Lyons Jed , Morehouse Jeffrey and Rocheleau David, 2011, A proposed vehicle for delivering a mechanical engineering systems laboratory experience, University of South Carolina,.

[17] Wakeham, Keith.J. and Rideout,.D. Geoff, 2011,Model complexity requirements in design of half car active suspension controllers, ASME Dynamic systems and controls conference, Arlington, VA, October 31-November 2, 2011 .

[18] Mukherjee, A., Karmakar, R. and Samantaray, A., 2006 , *Bond graph in modeling and simulation and fault identification. I.K. Publishing House pvt. Ltd.Delhi.*

APPENDICES I

BOND GRAPH

I.1 INTRODUCTION

A bond graph is a graphical representation of a physical dynamic system. It is similar to the better known block-diagram and signal- flow- graph, with the major difference that the arcs in bond graphs represent bi-directional exchange of physical energy, while those in block diagrams and signal-flow graphs represent uni-directional flow of information. Also, bond graphs are multi domain and domain neutral. This means a bond graph can incorporate multiple domains seamlessly.

The Bond Graph is composed of the "bonds" which link together "single port", "double port" and "multi port" elements (see below for details). Each bond represents the instantaneous flow of energy (dE/dt) or power. The flow in each bond is denoted a pair of variables called 'power variables' whose product is the instantaneous power of the bond. For example, the bond of an electrical system would represent the flow of electrical energy and the power variables would be voltage and current, whose product is power. Each domain's power variables are broken into two types: "effort" and "flow". Effort multiplied by flow produces power, thus the term power variables. Every domain has a pair of power variables with a corresponding effort and flow variable. Examples of effort include force, torque, voltage, or pressure; while flow examples include velocity, current, and volumetric flow. The table below contains the most common energy domains and the corresponding "effort" and "flow".

A bond has two other features described briefly here, and discussed in more detail below. One is the "half-arrow" sign convention. This defines the assumed direction of positive energy flow. As with electrical circuit diagrams and free-body diagrams, the choice of positive direction is arbitrary, with the caveat that the analyst must be consistent throughout with the chosen definition. The other feature is the "causal stroke". This is a vertical bar placed on only one end of the bond. It is not arbitrary. As described below, there are rules for assigning the proper causality to a given port, and rules for the precedence among ports. Any port (single, double or multi) attached to the bond shall specify either "effort" or "flow" by its causal stroke, but not both. The port attached to the end of the bond with the "causal stroke" specifies the "flow" of the bond. And the bond imposes "effort" upon that port. Equivalently, the port on the end without the "causal stroke" imposes "effort" to the bond, while the bond imposes "flow" to that port. This is made more clear with the illustrative in Table I.1, below.

Table I.1 Efforts and Flow Variables

Systems	**Effort (e)**	**Flow (f)**
Mechanical	Force (F)	Velocity (v)
	Torque (τ)	Angular velocity (ω)
Electrical	Voltage (V)	Current (i)
Hydraulic	Pressure (P)	Volume flow rate (dQ/dt)
Thermal	Temperature (T)	Entropy change rate (ds/dt)
	Pressure (P)	Volume change rate (dV/dt)
Chemical	Chemical potential (μ)	Mole flow rate (dN/dt)
	Enthalpy (h)	Mass flow rate (dm/dt)
Magnetic	Magneto-motive force (e_m)	Magnetic flux (φ)

APPENDICES II

CONCEPT OF BOND GRAPH ELEMENTS

II.1INTRODUCTION

In the bond graph formalism energy is conserved, power flow paths can be identified, and power flow associated with elements and their connections can be readily determined. Generalized inertia 'I' and capacitance ‘C’ store energy as a function of the system state variables, which are generalized momentum and displacements. The time derivatives of generalized momentum P and displacement q (Q) are generalized effort ‘e’ and flow ‘f’, the product of which is power. Generalized resistors R remove energy from the system, and have a constitutive law relating generalized effort to generalized flow. Sources of effort and flow (SE and SF) represent ports through which the system interacts with its environment. Energy is transported among source, storage and dissipative elements through power- conserving “Junction structure” elements. Such elements include power- continuous generalized transformers and gyrators (TF and GY), which are functions of external variables for example of co-ordinates. Kirchhoff’s loop and mode laws are modeled by power conserving. 1 and 0-junctions, respectively. Elements bounded to a 1-Junction have common flow, and their efforts algebraically sum to zero. Elements bounded to a 0- junction have common effort, and their flows algebraically sum to zero.

The power bonds contain a half-arrow that indicates the direction of algebraically positive power flow, and a casual stroke normal to the bond that indicates whether the effort or flow variable is the input or output from the constitutive law of the connected element. Full arrows are reserved for modulating signals, and they represent powerless information flows, such orientation angels for co-ordinate transformation matrices. Table 1 defines the symbols and constitutive laws for energy storage and dissipative elements (“energetic’ elements), source, and power-conserving elements. The constitutive laws are written in an input –output- form consisted- with the placement of the casual strokes.

II.2 SOURCE OF EFFORT (SE)

The source of effort is external to a system in the sense of that the system does not influence its output. The source of effort determines the effort in the bond associated with it.

F (t): SE

F (t, Pm, Qn): SE

II.3 SOURCE OF FLOW (SF)

The source of determines the flow variables in the bond associated with it. For source, the effort variable is an extra system entity and does not influence the output of this source.

V (t): SF

V (t, P_m, Q_n): SF

The Pn and Qn in the arguments indicate that the corresponding functions are functions of system state as well.

II.4 THE INERTIA ELEMENT (I)

Creation of this element is motivated by the idea of a mass point in mechanics. This element relates effort or its integration (the impulse) to the rate of change of momentum or the velocity. The relations for a simple linear inertial element may be written in the following forms:

$$e(t) = \frac{d(m(t)f)}{dt} \qquad (1)$$

or;

$$f(t) = \frac{1}{m(t)} \int_{-\propto}^{t} e(\varepsilon)\, d\varepsilon \qquad (2)$$

In these expressions the generalized inertia m (t) is taken as function of time.

$$e(t) = \frac{d(p)}{d(t)} \qquad (3)$$

$$p(t) = \int_{-\propto}^{t} e(\varepsilon)\, d\varepsilon \qquad (4)$$

II.5 THE COMPILIANT ELEMENT (C)

The compliant elements role model is a Hookian spring or an electrical capacitor, which stored energy depending on their configurations. The basic constitutive relation of a single port compliant element in the bond graph maps is written as

$$e(t) = F\left(t_1 \iint_{-\propto}^{t} (\varepsilon)\, d\varepsilon\right) \tag{5}$$

$$f(t) = \frac{dG\left(t_1 e(t)\right)}{dt} \tag{6}$$

Where F and G are single value functions of their arguments. G is of course respect to its arguments. In bond graph theory the integration of flow is called generalized displacement or charge of the C element,

$$Q(t) = \int_{-\propto}^{t} f(\varepsilon) d\varepsilon \tag{7}$$

The above relation may thus be written as:

$$e(t) = F\left(t_1\, Q(t)\right) \tag{8}$$

$$\frac{dQ(t)}{dt} = \frac{dG\left(t_1\, e(t)\right)}{dt} \tag{9}$$

$$Q(t) = G\left(t\,, e(t)\right) \tag{10}$$

$$e(t) = k(t)\int_{-\propto}^{t} f(F)\,d\varepsilon = k\big(t\,, Q(t)\big) \qquad (11)$$

or;

$$f(t) = \frac{d\big(e(t)/k(t)\big)}{dt} \qquad (12)$$

This is linear Hookian spring this has a constitutive relations. The variations in compliant parameters k and c in these linear models indicate function of the stiffness or capacitance with time.

II.6 THE RESISTIVE ELEMENT (R)

Unlike I and C element which relate essentially the integration of effort or flow variables to flow or effort on a bond, the resistive element involves no integration and directly relates the efforts to flow or vice-versa

$$e = \varphi(f) \qquad (13)$$

or;

$$f = \psi(e) \qquad (14)$$

In their linear forms may be

$$e = Rf \qquad (15)$$

$$f = \frac{1}{R} e \qquad (16)$$

Where φ and Ψ are simple valued functions of flow or effort respectively

II.7 TRANSFORMER (TF)

The bond graphic transformer can represent an ideal electrical transformer, a mass less lever; etc. The transformer does not create, store or destroy energy.

II.8 CREATION OF SYSTEM EQUATION

Method of generation of system equations is through an augmented (power directed and causalled) bond graph, using a step by step procedure, The difference between equations derived from bond graphs and otherwise is that there will be a set of N first order differential equations, where N is the number of states. (The total number of lumped elements I and C with integral causality present in a system).

A system or a part of it, changes from one configuration to another because there have been, or there are causes which act on it, and the system or a part of it absorbs a part of these causes. The amounts of absorbed causes may be good measures of the states of the system and the way these absorbed causes are changing with time contains the description of its dynamics. The system variables in which the equations are derived are thus the absorbed causes in storage element with integral causalities. The spring or capacitor element (I or C) receives flow and thus displacement or total charge is the associated with this element as

$$Q = \int_{-\infty}^{t} f dt \qquad (17)$$

Q is the general symbol which is used for displacement as well as charge. An inertia or inductance with integral causality receives effort as cause. Thus the system variable associated with this element is a generalized momentum as given by

$$P = \int_{-\infty}^{t} e dt \qquad (18)$$

System Variables for Bond graph based analysis is

$$\int_{-\infty}^{t} (cause)\, dt \qquad (19)$$

Where Cause is the information going to storage elements with integral causality.

II.9 GENERATION OF SYSTEM EQUATION

The system equations may be generated by answering the following two equations, what do the elements (all) give to the system expressed in terms of system variables and sources? It is known as expanded method. What does the system give to storage elements with integral causality? It is known as matrix method. All the ID (identity) terms created in this method are dimensionless.

In this method the equation may be written in Matrix form as shown below:

$$\frac{d}{dt}\{X\} = [A][X] + [B]\{U\} \qquad (20)$$

Where

$\{X\}$ *is state vector*

$\{U\}$ *is input vector*

$[A]$ *is system matrix*

$[B]$ *is input matrix*

In this section the method of generation of system equations is discussed.

The differential equations describing the dynamics of the system are written in terms of the states of the system. All storage elements (I and C) correspond to stored state variables (P for Momentum and Q for displacement respectively) and equations are written for their time derivatives (i.e. effort and flow).

II.10 CREATION OF SYSTEM BOND GRAPH

The impressions or models of nature are produced by human facilities and are like pieces of art. Such modeling of nature or any part of it can never be complete.

Each model is a creation of mind, thus incomplete due to limitations of understanding, due to the purpose for which it is created or due to limitations of the mediums used. Each system bond graph is a piece of art.

Like any other art, the art of creating system bond graphs may be acquired through a proper combination of learning processes by Mukherjee et al [18].

APPENDICES III

MAIN FEATURE OF SYMBOL SHAKTI SOFTWARE

III.1SYMBOLS BONDPAD

Bondpad can now generate equations as compared to earlier versions where a separate interface was needed. It can generate Simulation Files with handlers and capability for online variation of parameters. The one can build a sub-model and it can be encapsulated as Capsule allowing it to be incorporated in other models. It derives completely reduced set of state equations and for observed detector variables like power. The new equation derivation module is faster and optimizes instruction sets.

The Bond Pad of Symbolshakti software shown in Fig. III.1, can now export energetic and signal based sub-models to Matlab using S-function blocks. Models and sub-models are automatically compiled in MEX file format and all non-linear expressions, function calls and libraries are linked transparently to a single block.

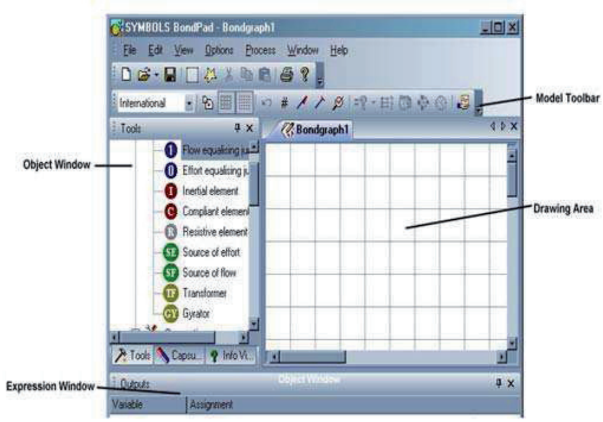

Fig.III.1 Symbols bond pad

III.2 SYMBOLS SIMULATOR MODULE

Simulator is the base post-processing module of Symbol Shakti. Here, the usual procedure of simulation using the simulator is presented.

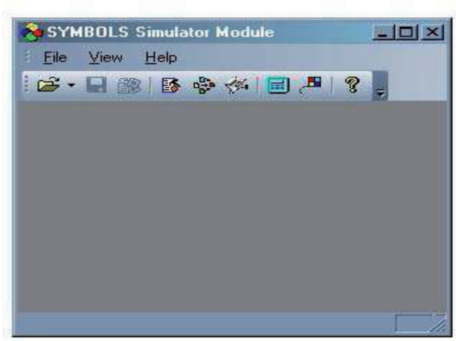

Figure III.2 Symbols simulator module

The simulator module shown in Fig.III.2, above has a multiple document interface (MDI), which allows multiple simulations of different projects at the same time.

The entry point to simulator is the "module definition File" (.sym file) generated from the bondpad. One must compile this file using the compile command from the file menu to create a simulator experiment file. After selecting the desired .sym file (Circuit.sym in example), the compilation interface continues as is shown below.

The user can select to set path for Executable files, Library files and include files from the combo-box. Then using the Add and Remove button, the path list can be updated for each of them. For example, if Borland compiler has been installed in the folder "C:\borland\bcc55" then the executables path must contain "C:\borland\bcc55\bin", they include file path must contain "C:\borland\bcc55\include" and the library files path should contain "C:\borland\bcc55\lib". The save button can save the path information into a initialization file and the user does not need to specify these paths for next time compilation.

III.3 SYMBOLS COMPILATION

Once the compilation commences, the compilation status, errors and warnings, if any, are displayed in a window as shown in Fig. III.3, and III. 4, below.

The compilation process creates a dynamic link library (DLL) file of the code. This DLL file is loaded by the solver program during simulation of the system. When user presses Close button, the "simulator experiment file" (.sxp file) is created and opened in the IDE. One can open this experiment file any time afterwards to directly continue simulation without taking recourse to the compilation tool unless any changes are made to the base bondgraph model and the .sym file. Any changes to the model requires recompilation. Now one can save simulation file by choosing “ Generate simulation code” item from the “Process” menu. Next the simulation code is compiled by choosing “Compile” from the process menu. Thus it is necessary to transform these codes to a C++ code and compilation so that it will run with SYMBOLS solver engine to produce numerical results. By choosing “Compile” it will open the compilation window with active documents simulation code as set path, create, compile and cancel. Now one can Press “Create” button to transform the internal code to a C++ code. Then one can press “Compile button to compile the newly created C++ codes to a machine language executable code which SYMBOLS SHAKTI simulator module will run along with its solver engine.

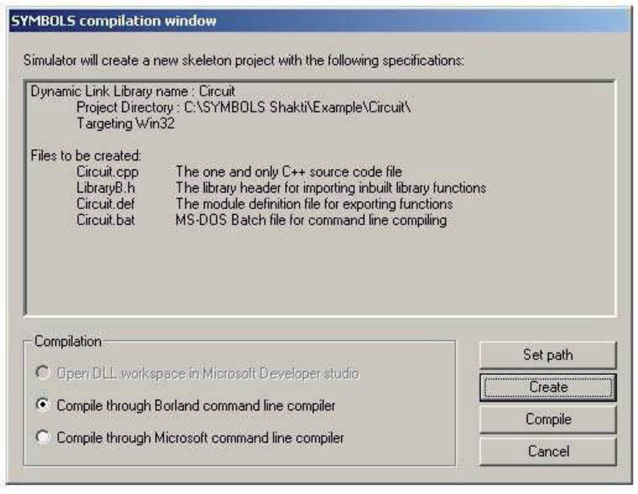

Fig. III.3 Symbols compilation window

If any error occurs in compilation, it will show an appropriate error or warning message in “Compile” tab. By double clicking on the error line, one can see where the mistake really occurs. Otherwise, it will show “0 error(s), 0 warning (s)”.

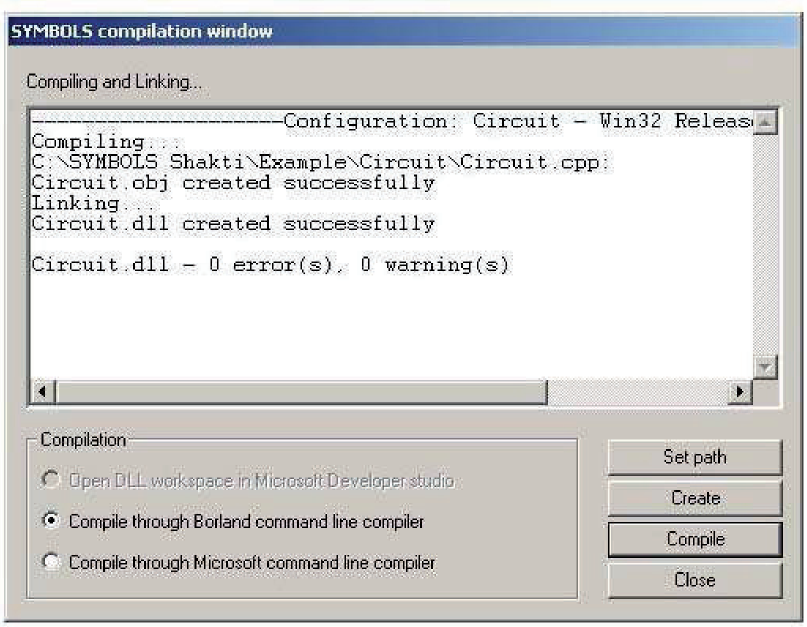

Fig. III.4 Symbols compilation window

III.4 SYMBOLS SIMULATOR MODULE CIRCUIT

The experiment window has three panes, which can be adjusted in size through vertical and horizontal splitters. The left part of the window is the system and plotting data setting control. The top-right part is used for online result plotting and the bottom-right part is the simulation specific control window shown in Fig.III.5, simulator module- Circuit.ext.

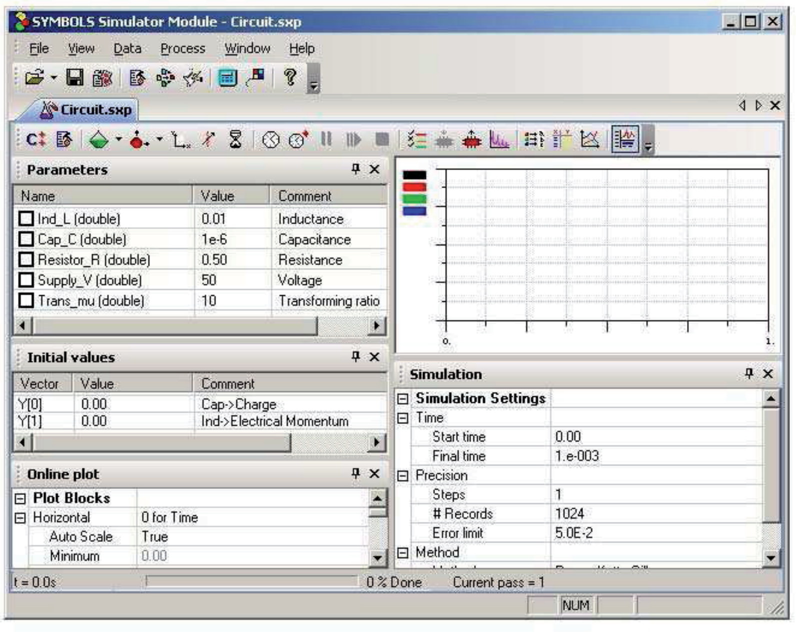

Figure.III.5 Symbols simulator module- Circuit.ext

Before starting simulation one must enter the appropriate parameter data and initial values, if necessary. When the mouse pointer is placed over a parameter or initial value entry box, the associated comment with that parameter or state is displayed as tool-tip to help the user identify the parameter. In simulator, the states and user states of the system are mapped to two separate 0-indexed arrays, namely Y and User. Information on this mapping can be obtained using the show map command from the view menu.

III.5 SYMBOLS ONLINEPLOT

If online plotting is desired, the plotblocks may be set appropriately.

They are situated below the initial value entry space as shown Fig.III.6; one can select upto four plotting variables in the vertical using the combo boxes. The horizontal block may be time or any other state or user variable to obtain either time response plots or phase plots. When the number of states is more, an alternate method of specifying plotting variables can be used by invoking the state map command from view menu. This displays mapping of all the default states of the model, their derivatives and User states. When you double click on any item on the list, a pop-up menu appears as shown below. One can select the appropriate plotting block there.

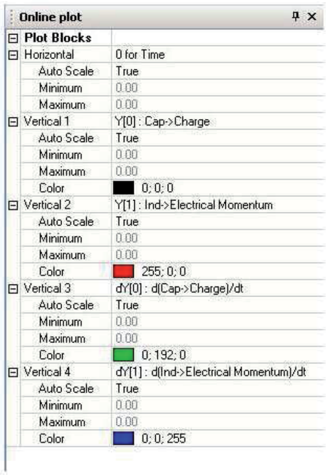

Fig.III.6 Onlineplot

This window allows one to set the starting and finishing time of the simulation, steps to be taken within recording interval, number of output points or records and the integration error limit.

Graph window by can be opened selecting “graphics display” item from the view menu. Thus a window will appear showing two graphs on it. One can choose “Tile vertically” item from the Mode menu of this window to see two graphs in tile mode as displayed.

APPENDICES IV

C++ CODE TO A MACHINE LANGUAGE EXCUTABLE CODE

For the second part of the dissertation involving a half car model with hinged suspensions, the following results have been obtained

```
//D:\manoj.cpp :
//Defines the initialization routines for the DLL.
#include <Windows.h>
#include <Fstream.h>
#include "LibraryB.h"
//Input variables...
double negative;//negative
double positive;//positive
double R18;//R18
double K15;//K15
double K17;//K17
double M19;//M19
double R16;//R16
double SE25;//SE25
double R27;//R27
double R28;//R28
double K30;//K30
double K32;//K32
double M24;//M24
double SE50;//SE50
double SE54;//SE54
```

```
double M49;//M49
double M53;//M53
double M57;//M57
double M60;//M60
double K75;//K75
double K77;//K77
double R76;//R76
double R78;//R78
double SF72;//SF72
double SF73;//SF73
//Global variables...
//Additional Global variables...
//Other include files..
#include "math.h" //for default math library
#include "float.h" //for default floating point operations
void FAR PASCAL _export Get_Params(double* Param_Values)
{
 negative=Param_Values[0];
 positive=Param_Values[1];
 R18=Param_Values[2];
K15=Param_Values[3];
K17=Param_Values[4];
M19=Param_Values[5];
R16=Param_Values[6];
SE25=Param_Values[7];
R27=Param_Values[8];
R28=Param_Values[9];
K30=Param_Values[10];
K32=Param_Values[11];
M24=Param_Values[12];
SE50=Param_Values[13];
SE54=Param_Values[14];
M49=Param_Values[15];
M53=Param_Values[16];
M57=Param_Values[17];
```

```
M60=Param_Values[18];
K75=Param_Values[19];
K77=Param_Values[20];
R76=Param_Values[21];
R78=Param_Values[22];
SF72=Param_Values[23];
SF73=Param_Values[24];
}
void FAR PASCAL _export State_Equations(double t,double* Y,double* dY,double* U)
{
Flush();
Report("Iterating in State_Equations ..");
//Local variables...
//Expressions for variables as entered in the expressions window
//Equations genereted from the model
dY[0]=1/negative*(-R28*(-Y[5]/M19-negative*Y[4]/M24+1/negative*Y[0]/M60
    +Y[3]/M49)-K32*Y[8])-positive*(K75*Y[7]+R76*(SF72+positive*Y[0]/M60
    +Y[3]/M49))-positive*(-K17*Y[10]-R18*(-negative*Y[4]/M24-Y[5]/M19
    -positive*Y[0]/M60-Y[3]/M49));
 dY[1]=negative*(-R27*(positive*Y[4]/M24+Y[5]/M19+negative*Y[1]/M57+Y[2]/M53)
    -K30*Y[9])-1/positive*(K77*Y[6]+R78*(1/positive*Y[1]/M57+Y[2]/M53+SF73))
    -positive*(-K15*Y[11]-R16*(-positive*Y[4]/M24-Y[5]/M19-positive*Y[1]/M57
    -Y[2]/M53));
 dY[2]=-R27*(positive*Y[4]/M24+Y[5]/M19+negative*Y[1]/M57+Y[2]/M53)-K30*Y[9]+SE54
    +K15*Y[11]+R16*(-positive*Y[4]/M24-Y[5]/M19-positive*Y[1]/M57-Y[2]/M53)
    -K77*Y[6]-R78*(1/positive*Y[1]/M57+Y[2]/M53+SF73);
 dY[3]=SE50-R28*(-Y[5]/M19-negative*Y[4]/M24+1/negative*Y[0]/M60+Y[3]/M49)
    -K32*Y[8]+K17*Y[10]+R18*(-negative*Y[4]/M24-Y[5]/M19-positive*Y[0]/M60
    -Y[3]/M49)-K75*Y[7]-R76*(SF72+positive*Y[0]/M60+Y[3]/M49);
 dY[4]=negative*(K17*Y[10]+R18*(-negative*Y[4]/M24-Y[5]/M19-positive*Y[0]/M60
    -Y[3]/M49))+positive*(K15*Y[11]+R16*(-positive*Y[4]/M24-Y[5]/M19
    -positive*Y[1]/M57-Y[2]/M53))-positive*(+R27*(positive*Y[4]/M24+Y[5]/M19
    +negative*Y[1]/M57+Y[2]/M53)+K30*Y[9])-negative*(-R28*(-Y[5]/M19
    -negative*Y[4]/M24+1/negative*Y[0]/M60+Y[3]/M49)-K32*Y[8]);
 dY[5]=R28*(-Y[5]/M19-negative*Y[4]/M24+1/negative*Y[0]/M60+Y[3]/M49)+K32*Y[8]
```

```
     +K17*Y[10]+R18*(-negative*Y[4]/M24-Y[5]/M19-positive*Y[0]/M60-Y[3]/M49)
     +SE25+K15*Y[11]+R16*(-positive*Y[4]/M24-Y[5]/M19-positive*Y[1]/M57
     -Y[2]/M53)-R27*(positive*Y[4]/M24+Y[5]/M19+negative*Y[1]/M57+Y[2]/M53)
     -K30*Y[9];
 dY[6]=1/positive*Y[1]/M57+Y[2]/M53+SF73;
 dY[7]=SF72+positive*Y[0]/M60+Y[3]/M49;
 dY[8]=-Y[5]/M19-negative*Y[4]/M24+1/negative*Y[0]/M60+Y[3]/M49;
 dY[9]=positive*Y[4]/M24+Y[5]/M19+negative*Y[1]/M57+Y[2]/M53;
 dY[10]=-negative*Y[4]/M24-Y[5]/M19-positive*Y[0]/M60-Y[3]/M49;
 dY[11]=-positive*Y[4]/M24-Y[5]/M19-positive*Y[1]/M57-Y[2]/M53;
}

void FAR PASCAL _export User_Equations(double t,double* Y,double* dY,double* U)
{
 Report("Calculating User_Equations ..");

//Local variables...

 //Expressions for variables as entered in the expressions window
 //Equations genereted from the model
 //User variables as defined in handlers
}

void FAR PASCAL _export Slider_Equations(int nSlider, double data)
{
 switch (nSlider)
 {
       case 0: negative=data; break;
       case 1: positive=data; break;
       case 2: R18=data; break;
       case 3: K15=data; break;
       case 4: K17=data; break;
       case 5: M19=data; break;
       case 6: R16=data; break;
       case 7: SE25=data; break;
```

```
        case 8: R27=data; break;
        case 9: R28=data; break;
        case 10: K30=data; break;
        case 11: K32=data; break;
        case 12: M24=data; break;
        case 13: SE50=data; break;
        case 14: SE54=data; break;
        case 15: M49=data; break;
        case 16: M53=data; break;
        case 17: M57=data; break;
        case 18: M60=data; break;
        case 19: K75=data; break;
        case 20: K77=data; break;
        case 21: R76=data; break;
        case 22: R78=data; break;
        case 23: SF72=data; break;
        case 24: SF73=data; break;
        default: break;
 }
}

bool FAR PASCAL _export Jacobian(double t, double* Y, double* U, double* ddt, double **ddY)
{
 //TODO: This member function is called by the simulator to allow you to code
        //your own Jacobian routine. The parameters passed to your function
        //reflect the parameters received by the application when this routine
        //was called. If you call the base-class implementation of this
        //function, that implementation will use the parameters originally
        //passed with the function and not the parameters you supply to the function.
        //To call base-class implementation simply return "false", otherwise write your
        //own equations for derivatives and return "true".

        //ddt vector holds time derivative of the differential equations in State_Equations,
        //i.e. ddt[0]=d(dY[0])/dt where dY[0]=d(Y[0])/dt.
```

```
        //Matrix ddY stores derivatives of differential equations in State_Equations with
        //respect to all the states, i.e. ddY[i][j]=d(dY[i])/dY[j].

 return false;
}

void FAR PASCAL _export OnSimulationStart(double t,double* Y,double* dY,double* U)
{
 //TODO: Add your specialized code here
}

void FAR PASCAL _export OnSimulationSettle(double t,double* Y,double* dY,double* U)
{
 //TODO: Add your specialized code here
}

void FAR PASCAL _export OnSimulationStop(double t,double* Y,double* dY,double* U)
{
 //TODO: Add your specialized code here
}
```