Topology Considerations in Hybrid Electric Vehicle Powertrain Architecture Design

by

Alparslan Bayrak

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2015

Doctoral Committee:

Professor Panos Y. Papalambros, Chair
Research Engineer Kukhyun Ahn, Ford Motor Company
Professor James S. Freudenberg
Group Manager Madhu Raghavan, General Motors Corp.
Professor Jeffrey L. Stein

For all the people

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge General Motors Corp. and the Automotive Research Center, a US Army Center of Excellence at the University of Michigan for funding my research activities in the Optimal Design (ODE) Lab.

I would like to give my special thanks to Prof. Panos Papalambros for his support and guidance throughout my research in the ODE Lab. In the college I learned that science was not just generating new knowledge but also communicating it. Panos helped me a lot to develop many skills to do that more effectively by challenging and correcting me all the time. There is still so much to learn.

The other person whom I owe so much is Prof. Gullu Kiziltas Sendur. She was the one who taught me the first steps to moving in the path of academia. Her endless support was an important factor for me to come to the University of Michigan. I can never repay her.

I would also like to thank Dr. Max Yi Ren and Dr. Namwoo Kang for their help at certain parts of the dissertation. Without their help, this dissertation would take much longer to finalize than it was supposed to. Additionally, it was a privilege to be a part of the ODE family. The friendship and all the fun we had together in the ODE Lab were so valuable to me.

Finally, I would like to thank my family. Looking back to all the difficulties we overcame together, receiving a PhD degree from the University of Michigan was beyond my imagination. Their unconditional love, which I believe is the most precious feeling in the world, was what kept me going.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

<u>**Table**</u>

LIST OF APPENDICES

Appendix

ABSTRACT

Topology Considerations in Hybrid Electric Vehicle Powertrain Architecture Design

by

Alparslan Bayrak

Chair: Panos Y. Papalambros

Optimal system architecture (topology or configuration) design has been a challenging design problem because of its combinatorial nature. Parametric optimization studies make design decisions assuming a given architecture but there has been no general methodology that addresses design decisions on the system architecture itself. The electrification of vehicles with the introduction of mechatronic devices such as motors and generators to vehicle powertrains has drawn renewed attention to the automotive powertrain architecture optimization problem. Hybrid Electric Vehicle (HEV) powertrains allow various architecture alternatives created by connecting the internal combustion engine, motor/generators and the output shaft in different ways through planetary gear systems. Addition of clutches to HEV powertrains allows changing the connection arrangement (configuration) among the powertrain components during the vehicle operation. Architectures with this capability are referred to as multi-mode architectures while architectures with fixed configurations are referred to as single-mode architectures. Design decisions made on both the powertrain's component sizes and its configuration have significant impact on the fuel economy and vehicle performance. System architecture optimization requires designing the configuration and sizing simultaneously. Additionally, evaluation of an HEV architecture design depends on a power management (control) strategy that distributes the power demand to the engine and motor/generators. Including this control problem increases the complexity of the HEV architecture design problem. Methodologies developed specifically for HEV powertrain architecture design work only when the problem size is significantly reduced by eliminating many architecture design candidates or target only a small portion of the design space of architecture alternatives.

This dissertation focuses on a general methodology to make design decisions on HEV powertrain architecture and component sizes. The representation of the architecture design problem is critical to solving this problem. A new general representation framework capable of describing all architecture alternatives is introduced. Using the representation, all feasible configurations are generated to create a new design space of feasible configurations only. These configurations are used to create single- and multi-mode HEV architectures. The architecture design alternatives are evaluated based on fuel economy, vehicle performance and complexity. Three types of design problems are formulated: (i) single-mode architecture design for given component sizes (ii) multi-mode architecture design for given component sizes (iii) architecture design combining the configuration and sizing. Solution strategies for all three types of design problems are developed. The high complexity of the resulting optimization problem does not allow us to claim true optimality rigorously; therefore, the terms “promising” or “near-optimal” are more accurate in characterizing the results of the optimization studies. Case study results show that different architectures must be designed for different applications. The case studies designing architectures for some available vehicles from the market find the architectures already implemented in these vehicles under some design constraints. Alternative architectures that improve these designs under different design constraints are also demonstrated. Architectures for a new application that is not available in the market are also designed.

CHAPTER I

Introduction

A system designer has various alternatives to consider when designing a system with certain design goals. While these alternatives may share a common architecture consisting of components with different specifications, the same goal can also be achieved by systems with very different architectures (topologies or configurations). Starting the design process with a "clean sheet", an effective selection of architecture design candidates among these alternatives requires a systematic process. Parametric optimization studies concentrate on the design decisions assuming a given architecture. Making the design decisions on the architecture itself is by far the most challenging design problem. Most often architecture design alternatives are unknown and must be generated using the designers intuition. Thus, a systematic approach to both generate and select alternative architectures is desirable.

In this dissertation we investigate the problem of designing hybrid electric powertrain architectures to maximize the fuel economy with satisfactory vehicle performance. We present a systematic way to generate and select promising design candidates for a general hybrid electric architecture design problem. We use a modified bond graph framework for the architecture representation. Generation and selection of architecture alternatives is done using an optimization formulation. The high complexity of the resulting optimization problem does not allow us to claim true optimality rigorously; therefore, the terms “promising” or “near-optimal” are more accurate in characterizing the results of the optimization studies.

1.1 Problem Definition

1.1.1 Classes of Design Problems

We introduce some terminology to create a common basis for the rest of the exposition. *Snavely et al.* (1990) provided some vocabulary to classify design problems in general that also applies to the problem discussed in this dissertation. Following the definitions in their work, a system consists of components, a collection of which forms a *catalog*. Each component in the catalog is called a *catalog entry*. Catalog entries are connected at *slots* and the configuration, i.e., the connection scheme among catalog entries defines the *topology* of the system. Figure 1.1 and Figure 1.2 illustrate these concepts on an abstract system.

In the design literature, the terms *topology*, *configuration*, *layout*, and *architecture* design tend to be used interchangeably to refer to the same design problem. In this dissertation the terms *configuration* and *architecture* that are more common in the automotive field are used with a subtle difference: Some electrical and mechanical systems can change the connection schemes among their components during their operation using switching components such as electrical switches or mechanical clutches. Each of these particular configurations is defined as a different *mode of operation* of the system and the combination of these configurations is called the *system architecture*. If the system does not have the ability to change its configuration, there is no difference between configuration and architecture.

The system design alternatives achieving a certain objective may have the same architecture with different catalog entries, which we call different instances of an architecture, or they might have different architectures while achieving the same

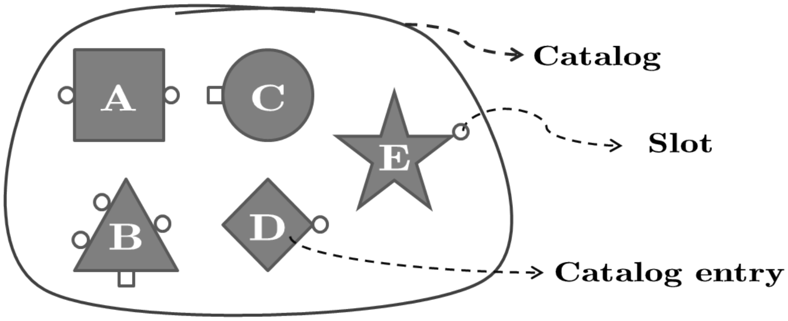


Figure 1.1: Illustration of design terminology

desired objective. Figure 1.3 shows examples of two DC/DC converters being able to achieve the same task with different architectures. The catalog entries used in both of these designs are different. While the half bridge DC/DC converter has two transistors (Q1 and Q2), the full bridge utilizes four transistors. By applying different control strategies to these two systems with different architectures, the same DC voltage input can be converted into the same DC voltage output. However, the losses in these systems (or in the general case the performance of the systems) are different.

Optimization is an important tool in the design process since finding the best or a set of "most promising" designs among the design alternatives is a key task in the system design process. "Proportional" or "sizing" optimization studies concentrate

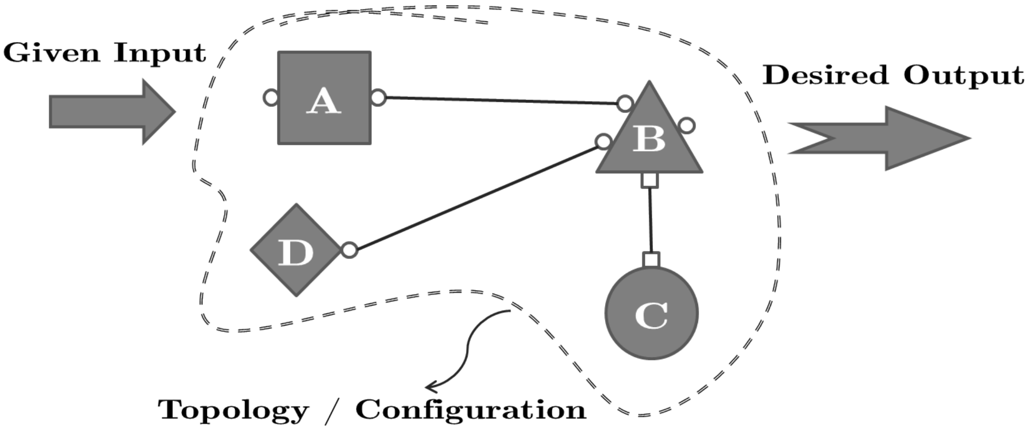
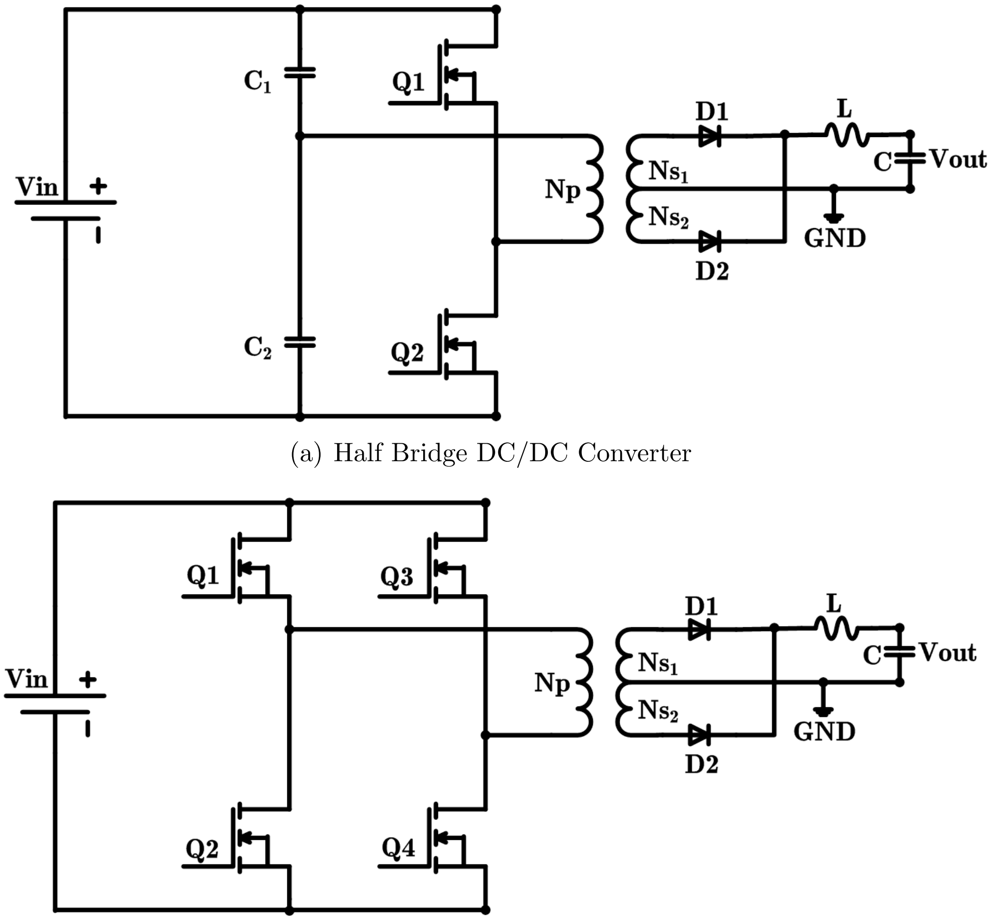


Figure 1.2: Topology of an abstract system to achieve a desired output using a catalog shown on Figure 1.1



(a) Half Bridge DC/DC Converter

(b) Full Bridge DC/DC Converter

Figure 1.3: Two DC/DC converters of different architectures

on optimizing the design variables associated with the catalog entries for a given architecture. Except for a small number of fields, such as structural or antenna optimization, the system architecture is considered fixed during optimization. The main reason for fixing the architecture is our inability to represent multiple architectures under a single optimization model. This dissertation addresses the problem of designing Hybrid Electric Vehicle (HEV) powertrain architectures considering fuel economy, vehicle performance and architecture complexity. These concepts are defined later in the dissertation.

1.1.2 Hybrid Electric Vehicle Architecture

The introduction of mechatronic devices such as motors and generators to vehicle powertrains has drawn renewed attention to the powertrain architecture optimization problem. Increasing demand for higher fuel efficiency places higher expectations on what can be achieved through powertrain electrification. Available architectures in the market are designed for specific types of vehicles, and a different vehicle with different specifications may require a different architecture designed to accommodate the particular vehicle and driving conditions. For example, an available architecture designed for a passenger vehicle might not result in satisfactory fuel economy when implemented in a heavy-duty truck. Architecture design has been shown to have a significant impact on the fuel economy (*Liu*, 2007). Hence, effective electrification requires a systematic approach to identify the appropriate vehicle powertrain architecture.

The powertrain architecture in a HEV mainly consists of an engine, one or more motor/generators (MGs), high-voltage and low-voltage batteries, power electronics (PE) devices, such as inverter and DC/DC converters and a transmission to connect the propulsion devices to the vehicle power output. We can divide the automotive architecture into two parts, as electrical and mechanical architectures. Electrical architecture refers to the configuration of the components on the electrical path, i.e., how batteries, PE devices and MGs are connected. Mechanical architecture refers similarly to the configuration of MGs, engine, transmission and vehicle power output. In this dissertation the focus is on the mechanical architecture.

Similar to the general definition given before, a mechanical powertrain architecture may consist of several configurations allowing switching from one configuration to another during vehicle operation. In an HEV architecture with this capability, each particular configuration is called a *driving mode* or simply a *mode*. In this case, the HEV architecture can be defined as a combination of driving modes. Having different driving modes can be used to achieve better drivability and efficiency under different driving conditions.

A common way to classify configurations or modes is based on how engine and electric motors are connected. The following subsections describe three classes of configurations, namely, *series*, *parallel* and *power-split*.

1.1.2.1 Series Configuration

The series configuration owes its name to the components being connected in a series manner, see Figure 1.4. An engine is connected to a generator that charges the battery. The battery is also connected to typically multiple motors that drive the vehicle. This design provides simplicity in the architecture. In this configuration, since the only purpose of the engine is to charge the battery, the engine runs independently of the vehicle motive operation. That way the engine can run in its efficient operating region regardless of the vehicle's motive operation. However, energy conversion at the generator and motor stages is a major source of energy loss. Also, at high speeds electric motors do not run as efficiently as the internal combustion engines which makes such a design not suitable for highway applications but a good alternative for city driving. A series hybrid architecture, i.e., an architecture consisting of only a series hybrid mode has not been implemented in any of the commercially produced hybrid vehicles but the Chevrolet Volt has a series hybrid mode in addition to other driving modes; see *Conlon et al.* (2011).

1.1.2.2 Parallel Configuration

In the parallel configuration, an engine and an MG drive the vehicle together, properly connected by a transmission. See Figure 1.5. The speeds of the MG and engine are linked via gearing and torques from both sources are combined to drive the vehicle. The MG is also used to charge the battery in which case the engine

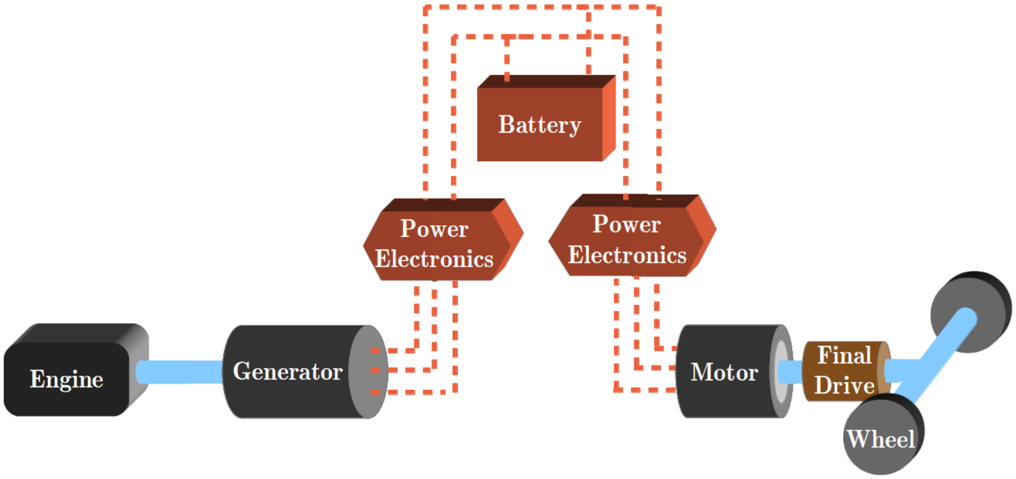


Figure 1.4: Series Hybrid Configuration

drives both the vehicle and motor together. In order to obtain a variety of operating regions a parallel hybrid configuration typically includes either a conventional automatic transmission or a continuously variable transmission. "The Integrated Motor-Assist System" developed by Honda Motor Company uses this architecture in mass-produced vehicles such as the Honda Insight Hybrid (*Aoki et al.*, 2000).

1.1.2.3 Power-split Configuration

A key element of the power-split configuration is the planetary gear (PG) set due to its versatility and high efficiency. An engine and MGs are connected to one or more PGs in different configurations driving the vehicle together; see Figure 1.6. A PG set is composed of a ring gear at the outer surface, a sun gear at the center and pinions in between. The pinions are connected via a bar, altogether forming the carrier gear as shown on Figure 1.7.

Depending on the number of PGs in the architecture, there are many possible ways to connect engine, motors and vehicle output providing many alternative configurations to select from. Power-split configurations that have been proposed in the literature include 1-PG, 2-PG, and even 3-PG systems, see Table 1.1. Commercial implementations of this configuration include the Toyota Prius and the Ford C-Max.

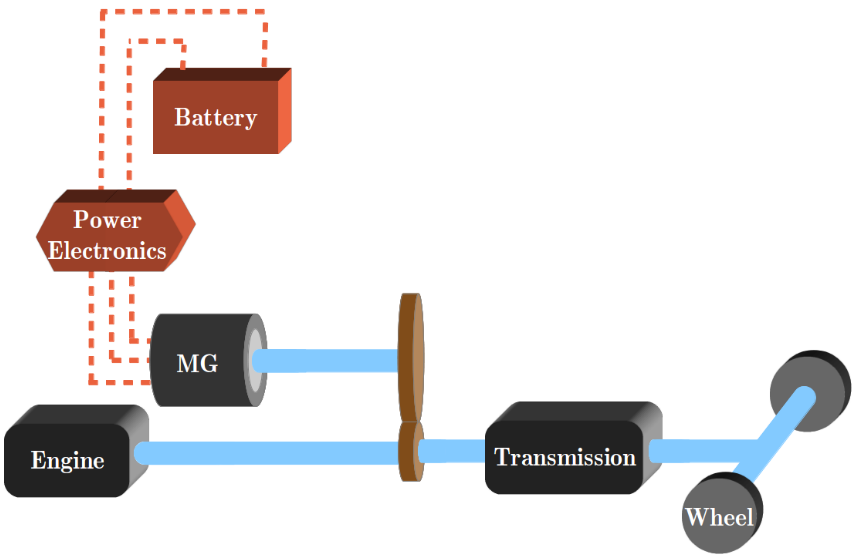


Figure 1.5: Parallel Hybrid Architecture

Also one of the modes of the Chevrolet Volt is a power-split configuration (*Conlon et al.*, 2011).

Table 1.1: Existing architectures from literature

Number of PGs	Reference
1-PG	*Conlon et al.* (2011) *Sasaki* (1998)
2-PG	*Schmidt* (1996b) *Schmidt* (1996a) *Holmes et al.* (2003) *Holmes and Schmidt* (2002) *Ai and Anderson* (2005)
3-PG	*Schmidt* (1999b) *Raghavan et al.* (2007)

The use of PGs allows creation of many architecture alternatives. For example, by grounding certain gears in the PG set a parallel configuration can be obtained. Due to the versatility provided by PGs, the main focus in this dissertation is on architectures obtained by configuring powertrain components with PGs in different ways.

When there is no parametric representation available to define the space of archi-

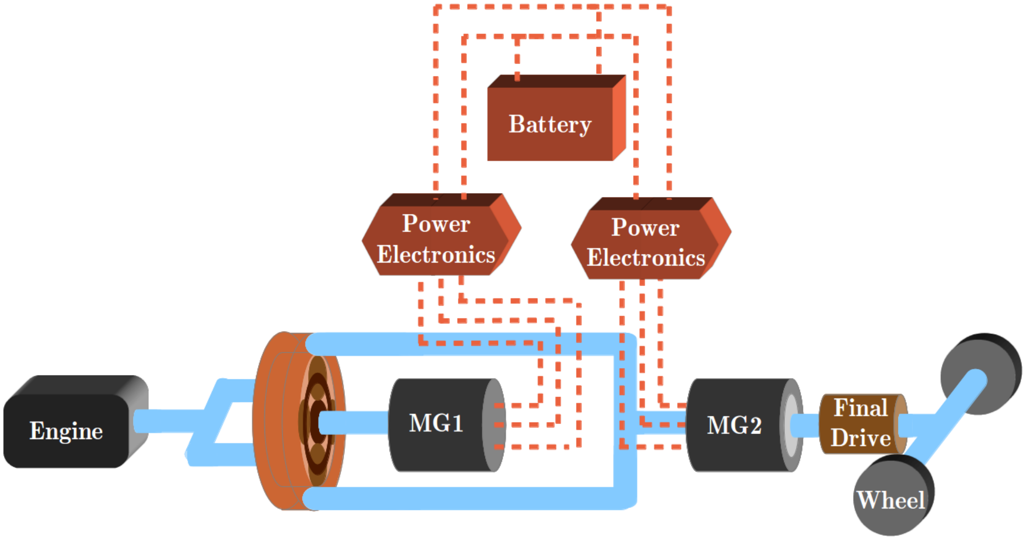


Figure 1.6: Power-Split Hybrid Architecture

tecture alternatives, a model capable of representing all the architecture alternatives is necessary. We developed a new "modified bond graph" representation for this purpose. Chapter III describes this representation in detail and proposes a framework for generating all configuration alternatives.

1.1.3 HEV Powertrain Control

The availability of the additional MGs compared to the conventional automotive powertrains provides flexibility to distribute the power demand from the vehicle output shaft among the engine and MGs. A systematic way to perform this power management requires a control strategy. In the literature, there are some available control strategies applied to the HEV powertrain control problem that perform this power distribution optimally, such as Dynamic Programming (*Lin et al.*, 2003; *Liu and Peng*, 2006) or Pontryagin's Minimum Principle (*Delprat et al.*, 2002, 2004; *Kim et al.*, 2011). In addition, some heuristic methods driven by the Pontryagin's minimum principle were introduced such as the Equivalent Consumption Minimization Strategy (*Paganelli et al.*, 2002). A detailed description of these strategies is given in

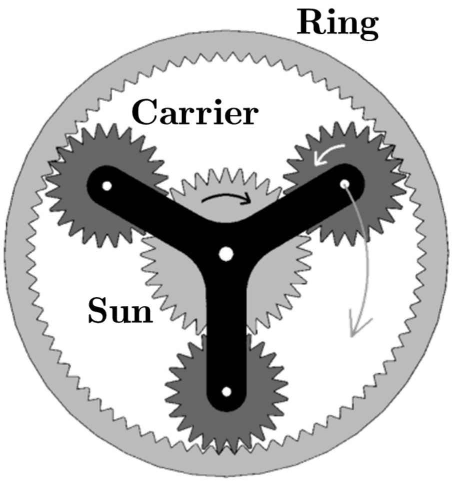


Figure 1.7: Cross section representation of a PG set

Chapter V. Some rule-based control strategies are also available but they typically perform much worse than the optimal control strategies.

Addition of this control problem increases the complexity of the HEV architecture design problem. We cannot separate the design and control problems from each other since they are coupled and both have significant impact on the fuel economy of the designs. Solving the control problem inside the design problem adds significant computation cost to the evaluation of HEV architecture design alternatives and makes the enumeration of all possible designs intractable except in certain specific cases to be discussed later in the dissertation. In Chapter IV, we give a more detailed discussion on the coupling between the design and control problems, and describe the solution strategy implemented in this dissertation.

Considering an HEV architecture as a combination of multiple modes, finding promising ("near-optimal") architectures is a challenge because of the combinatorial nature of the problem. When the number of architecture alternatives is small, enumeration of all possibilities might be used to find the best; but the large number of alternatives for a hybrid electric vehicle powertrain makes the assessment of all possible alternatives computationally intractable mainly because of the combined design and control problem. Therefore, some form of efficient search is necessary to find the optimal selection or to reduce the number of alternatives that must be explicitly evaluated.

True system optimization requires combining both architecture and sizing optimization. Unless the component specifications are fixed by some constraints, two architectures can only be compared at their best instances. As we will explore in later chapters, optimizing simultaneously architecture and sizing of their components is very challenging. An initial approach might be to consider architectures comprised of fixed components. This approach is taken in Chapters V and VI. The more general optimization problem in discussed in detail in Chapters IV and VII.

1.2 Related Work

Optimal architecture design has been a taunting problem in design research. There has been no general approach developed that works for the different types of architecture problems. The main difficulty is lack of appropriate problem representation.

Creating new design solutions (concepts or configurations) from scratch or by combining existing designs is also referred as design synthesis. *Shea and Starling* (2003) and *Cagan et al.* (2005) describe the design synthesis process in four phases which are *representation*, *generation*, *evaluation* and *guidance*. Representation refers to the task of describing the system of interest with a general enough model capturing form or function of system. The representation task is usually done by the designer manually based on the analysis of existing systems. The generation task uses the representation to create feasible design alternatives. So the generation task must have some rules on the representation describing feasibility. Evaluation refers to the process of measuring quality of the design in terms of the design criteria. Finally, guidance is feeding the output of the evaluation task back to the generation in order to find good search directions for new better alternatives in the design space. The last three phases are usually automated and work in a loop until the process converges to promising design solutions. Representation is the key which determines the computational efficiency of the generation, evaluation and guidance tasks. A representation allowing the use of available search methods proven to be effective can help to solve the design problem in an efficient way.

The design synthesis concept above parallels the standard design optimization process. The representation task is the creation of a mathematical optimization model that will allow exploration of design alternatives using optimization algorithms. Architecture design is a challenging problem exactly because such mathematical models have been proven difficult to create (with few exceptions).

The following subsections describe works in other fields incorporating all these phases in their design process. A more detailed literature review is given in Chapter II.

1.2.1 Structural Design Synthesis

Kicinger et al. (2005) classify the structural topology problems into two categories as *continuum* and *discrete* design problems based on the way the design space is represented. Continuum type topology design, also simply called *topology optimization* within its own field is the only type of architecture problem which is addressed in an efficient and generalized way. In the attempt to find the optimum topology of structures given some loading conditions, the problem is converted into finding the optimal material distribution over a domain; see *Bendsøe and Kikuchi* (1988). What enables this computationally efficient solution is the continuous representation of the topology over a design domain. The design domain is discretized into a finite number of cells where having a design variable value of “1” means existence of material and “0” otherwise. This problem with binary variables is then represented by a continuous function which penalizes the intermediate values to either “0” or “1”. With the help of this continuous representation, the problem can be solved using conventional optimization techniques applied to a finite element model of the structure. Although this is classified as a continuum approach, discrete formulations of the same concept have been implemented using Genetic Algorithm (GA) (*Sandgren et al.*, 1990; *Jensen*, 1992). This concept has been successfully applied to non-structural domains from fluid mechanics (*Borrvall and Petersson*, 2003) to electromagnetics (*Kiziltas et al.*, 2003) where a field property like material could be distributed in the design domain and field equation can be used to evaluate system performance for different distributors.

The second category in structural design is the *discrete* topology design problems. This type of formulation starts with a given finite set of elements, i.e., a catalog, and optimizes the connectivity of the selected catalog entries. Truss and frame structures are two common applications of discrete topology design problems. The most common formulations include, location coordinates, sizing of links and connectivity of joints as design variables. Evolutionary algorithms, GA being the most common one, are extensively used to solve the resulting optimization problem. As the problem size and number of constraints in the system increase, evolutionary algorithms have difficulty in finding feasible design solutions and this impacts computational solution efficiency significantly (*Kicinger et al.*, 2005).

Design synthesis of kinematic chains is also similar to the structural design synthesis problems. In this type of design problems, topology, size of the linkages, required torque from the actuators are design variables. Available work in the literature focused on open and closed kinematic chain design in order to meet the position (*Roth*, 1967) and force requirements (*Raghavan*, 1989; *Huang and Roth*, 1993) from the resultant system.

1.2.2 Electrical Circuit Design

Synthesizing electrical circuits that provide a desired output to a given input electrical signal is a problem that has been investigated by *Koza et al.* (1997); *Zebulum et al.* (1998); *Lohn and Colombano* (1998). The general approach is to start from a given catalog of circuit elements and explore different ways to connect these elements until the desired output is obtained. A typical application is filter design where the desired output is expressed in terms of frequency response of the system. Additional selection criteria may include the number of catalog entries used, power consumption and overall space occupied by the design. In these studies, not only the configuration but also the sizing of the catalog entries are incorporated into the problem formulation. Instead of a continuous representation, sizing variables are allowed to take only commonly manufactured values. Evolutionary Algorithms (EAs) are extensively used to solve these problems. A graphical representation or a string representation of variable length that allows incorporating EA operations are commonly used.

1.2.3 Chemical Structure Design

In the chemistry domain, the architecture design problem corresponds to finding chemical structures possessing certain chemical properties. A common application area is the synthesis of new drugs. Desired chemical properties are typically formulated as objectives such as an interaction score or similarity to a known ligand which are both estimates of the tendency of a molecule to bind to a target protein. In addition, other considerations such as absorption, distribution, metabolism, excretion and toxicity may serve as additional objectives or constraints in an optimization formulation (*Schneider and Fechner*, 2005). Since the number of feasible alternatives is estimated to be on the order of 10^{100}, enumeration over all feasible candidates is not possible (*Walters et al.*, 1998). Therefore, evolutionary algorithms combined with a graphical representation of the molecules are commonly used to search the design space efficiently (*Schneider et al.*, 2000; *Nicolaou et al.*, 2009; *Weber*, 2002).

1.2.4 Automotive Powertrain Configuration Design

An extensive configuration design study for conventional automotive powertrains was proposed by *Kahraman et al.* (2004). The goal was to find planetary gear configurations giving the desired speed ratios within a range. Speed ratio for a conventional powertrain is defined as the ratio between input engine speed and vehicle power output speed. Conventional powertrain architectures are one-Degree Of Freedom (1-DOF) systems where input and output speeds are linked by a speed ratio. As described earlier, by changing the connections among the planetary gear sets as well as the nodes where powertrain components are connected, many configuration alternatives are possible. *Kahraman et al.* (2004) formulated some feasibility constraints in terms of clutch connections, maximum gear and bearing speeds, and forces. These constraints are then used in an enumeration framework to eliminate infeasible designs. All configurations satisfying the constraints and giving the target speed ratios are selected for further consideration such as efficiency or packaging.

In the case of hybrid electric powertrains, additional MGs in the powertrain make the 2-DOF systems also useful. The MGs allow engine to operate independent from the vehicle motive operation,i.e., input and output speeds are not necessarily linked together. However, this additional degree of freedom increases the problem size compared to the conventional case. *Liu* (2007) proposed a HEV architecture design framework capable of finding 2-PG power-split system configurations with two modes. Liu represented a driving mode with a dynamic system matrix explained in further detail in Chapter III, and classified the power split system into three froms: *input-split* (one MG is connected to the vehicle output), *output-split* (one MG is connected to engine) and *compound-split* (none of MGs is connected to engine or vehicle output). Leaving output-split configurations out, Liu found all feasible driving modes using this representation and assuming that the optimal architecture must have an input-split and a compound-split mode. By applying a set of constrains the number of feasible mode alternatives was reduced to a small number for the input-split mode in the architecture. Then, the compound-split modes that can be switched from the remaining feasible input-split-modes were identified by allowing only two clutches. Integrating some shifting considerations, the number of input-split alternatives was further reduced and powertrain component sizing optimization was performed for the remaining candidates.

1.3 Summary

Finding the promising HEV architecture candidates considering vehicle performance, fuel economy and complexity is a challenging task because of the combinatorial nature of the problem. There is no available methodology that works well for general system architecture design problems. Available methodologies are limited to specific domains and cannot be applied to the HEV architecture design problem. Methodologies developed specifically for HEV architecture design problems either work only when the problem size is significantly reduced by eliminating many design candidates through additional constraints or target only a small portion of the design space to search for architecture alternatives.

The goal of this dissertation is to introduce a more general methodology to identify promising HEV architectures and gear ratios allowing any number of PGs and powertrain components to be incorporated into multiple driving modes. Since representation is a crucial element to solve this problem efficiently, a general representation framework capable of describing any HEV powertrain architecture is introduced. Using the proposed representation, all configuration design alternatives are generated to create the feasible configuration design space. An evaluation process based on vehicle performance, fuel economy and complexity is then used by a search algorithm to identify near-optimal solutions. Different solution strategies are proposed for different problem types. Three problem types are investigated: (i) single-mode architecture design for given gear ratios, (ii) multi-mode architecture design for given gear ratios, (iii) architecture design combining configuration and gear ratios.

1.4 Dissertation Contributions

Three main contributions of this dissertation are as follows

(i) A new representation of hybrid electric vehicle (HEV) architectures is created based on concepts from bond graphs that is general enough to represent architectures with any number of powertrain components connected through any number of planetary gears (PGs). Using this representation, a general formulation is derived to generate all feasible configurations, i.e., driving modes, for any given number of powertrain components and PGs.

(ii) A new, combined HEV architecture design and control problem formulation is derived and solution strategies for single and multi-mode architecture design problems for given powertrain component sizes and gear ratio selections are demonstrated.

(iii) A general decomposition-based formulation for the combined HEV architecture design and control problem is developed for the more general case that includes component sizing as design decisions. This formulation includes a partitioning model and a coordination strategy using analytical target cascading.

In order to demonstrate the capabilities of the methods proposed, we will present the generation results for all feasible power-split type 1-PG, 2-PG and 3-PG driving modes with one engine, two MGs and one ground. Then, using these feasible driving modes, we will show the design of 1- and 2-PG single- and multi-mode architectures for different vehicle applications by considering fuel economy, vehicle performance and architecture complexity. Finally, for the design problem of combined architecture and component sizing, we will present separate decompositions for single and multi-mode architectures. We will demonstrate this formulation by designing 1- and 2-PG single- and multi-mode architectures and gear ratios simultaneously for a particular vehicle application.

1.5 Dissertation Overview

The rest of the dissertation is organized as follows: In Chapter II an analysis of the relevant literature and a discussion on the gap in the literature to address the problem investigated in the present dissertation is given. Chapter III describes a new general HEV architecture representation followed by an automated framework that generates the feasible driving modes relying on this representation. Chapter IV formulates the general HEV architecture optimization problem for different scenarios and gives an overview of the solution strategies. Chapter V describes the architecture evaluation method based on vehicle performance and fuel economy. The solution method for single-mode HEV architecture design problem with given gear ratios and vehicle parameters is also describe in that chapter. Chapter VI defines a complexity measure to be used for multi-mode architecture evaluation and describes the solution strategy to find promising architectures in terms of fuel economy, vehicle performance and complexity for given gear ratios and vehicle parameters. Chapter VII discusses the combined configuration and gear ratio design problem for single and multi-mode architectures and presents a decomposition-based formulation to solve the combined problem. Chapter VIII gives a discussion on the results, contributions of the dissertation work and some future directions to improve and extend the proposed methods.

CHAPTER II

Literature Review

In Chapter I we gave a brief overview of some existing work on architecture design in different domains. In this chapter we discuss a more detailed architecture design literature review specific to automotive powertrains. Since in this dissertation, we use a bond graph based representation for HEV architecture, we also give a review on similar approaches from other fields. The purpose is to point to gaps in the literature for solving the general problem posed in Chapter I. In this chapter we take a more detailed look at the HEV architecture design problem and describe the available approaches from the automotive design literature as well as some useful tools introduced in electrical and mechatronic system design.

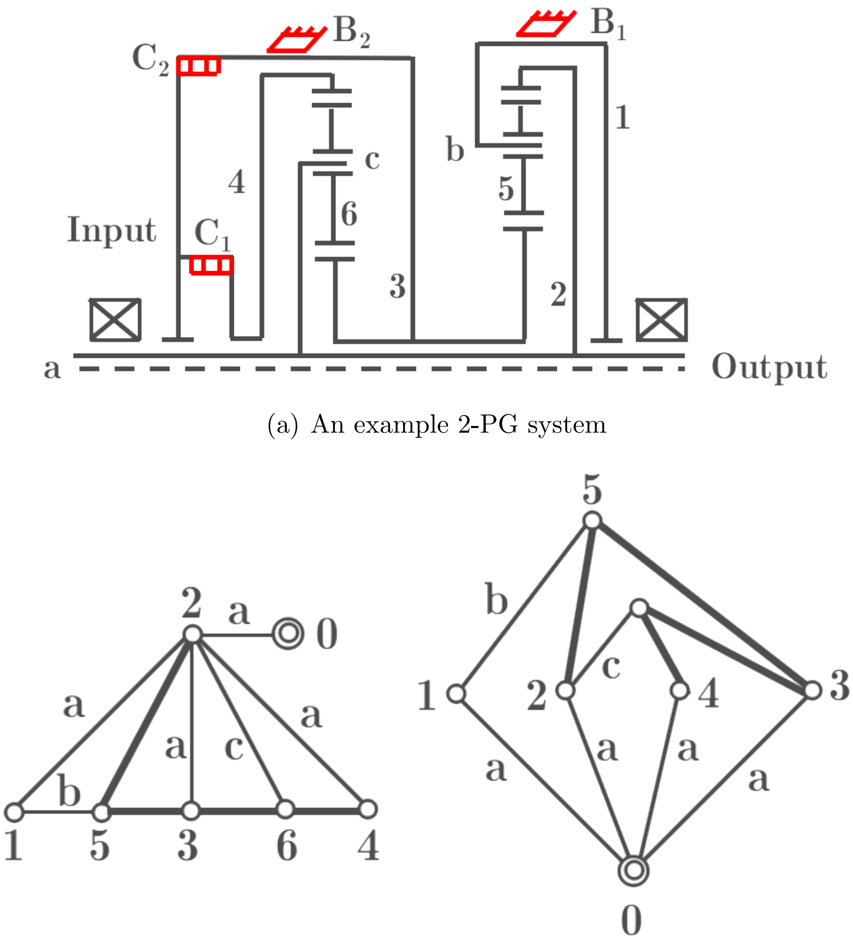
2.1 Automotive Powertrain Architecture Design

We group the automotive architecture design literature into two categories; automatic transmission configuration design, and hybrid powertrain configuration design. Both design problems use PG sets as the key elements to connect the powertrain components. In the case of automatic transmission design, the components to connect are input shaft (engine), grounds and output shaft (vehicle power output). For hybrid vehicles, the powertrain components to connect include a prime mover, a secondary mover, grounds and vehicle output. Primary mover is usually an internal combustion engine but the secondary mover varies depending on the application. For instance, hybrid electric powertrains use electric motor/generators as secondary movers and hydraulic hybrids use hydraulic pump/motors. The common problem for both automatic transmission and hybrid powertrain configuration design is to find the gear ratios and connection arrangement of PGs and powertrain components satisfying the design goals. Since the number of possible configurations is typically a large number, a systematic method to search among these configurations is necessary. Sections 2.1.1 and 2.1.2 describe the existing approaches used to address these problems.

2.1.1 Automatic Transmission Design for Gasoline Vehicles

For automatic transmission design, since there is only one (primary) mover which is the internal combustion engine, the interest is to design a 1-DOF system where only one independent variable is enough to determine all other states. In a 1-DOF system, the input shaft connected to the engine and the output shaft connected to the wheel shaft are linked via gearing. When designing an automatic transmission for a given application, i.e., for a given set of vehicle parameters and drive cycles, some target speed ratios can be determined to match the vehicle power requirements considering engine characteristics (*Kahraman et al.*, 2004). The design problem then becomes finding the PG configurations with the necessary clutches and PG ratios achieving these design targets.

As discussed in Chapter I, representation is the key challenge in design synthesis problems. Using graphical representations to synthesize geared kinematic chains is addressed in early work by *Buchsbaum and Freudenstein* (1970). We will refer to this representation as *conventional graph representation*. In this representation, the links between components are represented by vertices and the joints are represented by edges. Joints representing the gear pairs are represented by thick edges and revolute joints are represented by thin edges. Figure 2.1 shows an example.



(a) An example 2-PG system

(b) Conventional graph representation

(c) Canonical graph representation

Figure 2.1: An example 2-PG system and its conventional and canonical representations given by *Chatterjee and Tsai* (1996)

Two important aspects to consider when representing PG systems using graphs are identifying *isomorphism* and *pseudo-isomorphism*. In a graphical representation, assignment of node and edge labels to the physical components is arbitrary. By changing these labels the same configuration can be represented by "different-looking" graphs. Such graphs with different labels are called *isomorphic* graphs. Isomorphic graphs have different adjacency matrices representing the connections among the nodes in the graph. Isomorphism always exist in any graphical representation. However, they can be identified and eliminated before evaluating the systems.

Gear train systems that have coaxial links can be reconfigured to result in the same kinematic relationships. After reconfiguration, although there is a different non-isomorphic graph to represent the reconfigured system, it gives the same set of kinematic equations to be used in the evaluation of the configuration. Such a case is referred as *pseudo-isomorphism*. Examples of pseudo-isomorphism can be found in *Chatterjee and Tsai* (1996).

The conventional graph representation of PG systems offered by *Buchsbaum and Freudenstein* (1970) eliminates isomorphic graphs but cannot identify pseudo-isomorphism. For some applications, keeping pseudo-isomorphic graphs can be useful. For instance, when connecting gear mechanisms together, among the pseudo-isomorphic pairs, the one that allows connection in the simplest way can be preferred. A discussion on pseudo-isomorphic graphs in line with the goals of the present dissertation will be given in Chapter VI.

For applications where these pseudo-isomorphic graphs are not desired, *Chatterjee and Tsai* (1996) introduced a *canonical graph representation* of PG systems. By imposing some rules on the canonical graph representation, they obtained a unique graph representation for kinematically equivalent mechanisms. Once there is a good representation which allows generating different gear configurations, evaluating these configurations is not computationally expensive for automatic transmission designs. As discussed earlier in this section, the goal of the automatic transmission designs is to find configurations resulting in the desired speed ratios. Speed ratio analysis does not have any dynamic component and it can be extracted from a graph representation instantly with the present computational power. The same idea applies to the steady-state speed and torque analysis. Several methodologies linking the graph representation to static and kinematic analysis have been introduced. For instance, *Freudenstein and Yang* (1972) introduced the concept of fundamental circuits to perform the static torque and kinematic analysis of the PG systems from a conventional graph representation. Also, *Hsieh and Tsai* (1996a) used another concept, namely, fundamental geared entities to perform similar analysis for the canonical graph representation. The low computational cost of these analyses allows enumeration of possible designs to search for configurations meeting the design targets.

Hsieh and Tsai (1996b) used their canonical graph representation combined with the fundamental geared entities concept to enumerate the most efficient clutching sequences for three and four-speed transmission designs. *Kahraman et al.* (2004) performed a larger enumeration study to find all 2-PG systems and clutching sequences for a 6-speed transmission satisfying the design targets.

2.1.2 Hybrid Architecture Design

In the case of hybrid powertrains, in addition to the prime mover, there are secondary movers that aid the prime mover to propel the vehicle as well as recovering some of the braking energy. Having multiple movers makes 2-DOF configurations feasible for hybrid powertrains. As described in Chapter I, series and parallel configurations are still 1-DOF configurations (excluding the degree of freedom coming from additional transmission) but power-split configurations that allow a variety of configurations can be 1-DOF, 2-DOF or 3-DOF systems depending on how many nodes are grounded in the configuration. Available implementations of power-split configuration are 2-DOF systems. When discussing the hybrid powertrain control strategies later in this dissertation, we will show that 2-DOF systems offer some flexibility in control as opposed to 1-DOF systems. 3-DOF systems offer more flexibility in control but require more sophisticated control methods.

One of the most extensive studies to design 2-PG hybrid electric power-split architectures with two modes was done by *Liu* (2007). A matrix representation was used to model the dynamic behaviour of the configurations. Liu classified the power-split configurations into three types: *input-split* (one MG is connected to the vehicle output), *output-split* (one MG is connected to engine) and *compound-split* (none of MGs is connected to engine or vehicle output). Using this representation and leaving output-split configurations out, 1152 possible hybrid modes were generated. Assuming certain gear ratios and vehicle parameters, a filtering process to eliminate the designs failing to satisfy the driveability constraints resulted in 17 input-split feasible configurations assuming that an optimal two-mode hybrid electric architecture must have an input-split and a compound-split mode. For the remaining input-split configuration candidates, the compound-split configurations that require only two clutches were generated. Among the generated two-mode architectures, considering some shifting mechanics, infeasible architectures are eliminated further and only two design candidates remained for further sizing optimization.

Although a similar problem to the one posed in Chapter I was investigated, the work by *Liu* (2007) is limited with respect to our present scope for the following reasons: (1) The representation is not general enough to generate architectures with any number of PGs and any number of modes. (2) Not all configuration possibilities are included in the analysis. For instance configurations having engine and motor connected to the same gear (referred to as output-split) are excluded from the analysis. Also the representation is limited to hybrid configurations only, i.e., a pure EV configuration cannot be generated. (3) An initial filtering of configurations is performed assuming some gear ratio values. However, each configuration should be evaluated with the best set of gear ratios (with respect to the chosen design objective), unless the gear ratios are fixed by some other constrains. (4) The vehicle parameters such as battery size, motor sizes etc. are selected such that only a handful of configurations pass the filtering process. This work cannot be applied to the problems with many feasible configurations passing the filtering process.

Another study on the design of multi-mode hybrid electric architectures was done by *Zhang et al.* (2012). Given some vehicle parameters, they investigated the benefit of designing additional modes to the existing hybrid power-split configurations of the Toyota Prius (input-split) and the Chevy Volt (output-split). They enumerated all possible clutches that can be added to these two configurations and analyzed every possible mode obtained from the enumeration to identify promising designs in terms of fuel economy. They performed the same analysis for other possible 1-PG input and output-split configurations but concluded that only existing configurations give promising results. They then proposed one additional pure electric mode for both the Toyota Prius and the Chevy Volt hybrid configurations. They also showed that one of the existing pure EV modes and the series hybrdi mode in the Chevrolet Volt do not provide any significant benefit to the vehicle in terms of fuel economy.

The work by *Zhang et al.* (2012) is limited to 1-PG configurations with given gear ratios. Also, only two-mode systems are investigated in that study. As we will discuss in Chapter V for 1-PG systems, since the number of architecture possibilities is limited, enumeration over all architectures is possible when all the gear ratios are fixed. As the number of PGs in the architecture increases and if gear ratios are taken as design variables, the enumeration is not computationally feasible anymore. A generalized systematic method becomes necessary.

A hybrid configuration design framework including gear ratios as part of the problem was proposed by *Cheong et al.* (2011) for hydraulic hybrid powertrains. They used the kinematic relationship matrix to represent the hybrid configuration. Details of that representation will be explained in Chapter III in detail. One key contribution of that work was to show that any kinematic matrix can be realized by using a specific 2-PG configuration where there was an extra gear in front of each powertrain component and in-between PG sets. When all possible kinematic matrices are possible to realize, they were able to use the elements of this matrix as design variables to optimize the fuel economy.

One important limitation of their study is that the proposed methodology is valid only for the configuration proposed in their study. Extra gears connected to the powertrain components and in-between PG sets adds extra complexity to their architecture, which might not be desired. A 1-PG configuration or a 2-PG configuration without additional gears cannot be designed since all possible kinematic matrices are not possible to realize anymore. Also creating multiple modes by adding clutches in between PG sets cannot be considered in that design framework.

A hybrid configuration design method that can be generalized to any number of PGs, allowing any number of modes and including gear ratios in an overall optimization formulation seems to be unavailable in the literature. The key contribution of this dissertation to propose such a design methodology.

2.2 System Architecture Design Using Evolutionary Algorithms

Graphical tools that allow generation of design alternatives are common representations of configurations. The graphical representations commonly used in the automotive literature, such as canonical graphs, represent the geometrical arrangement of the components in the configuration. In this section we will discuss one more example from the electrical domain using a netlist representation based on the geometric arrangement of circuit components. In addition, we will also discuss the use of alternative representation tools, namely, bond graphs (*Karnopp et al.*, 2012) that are based on the power flow among the components. bond graphs, details of which are given in Chapter III, are extensively used in this dissertation to model and generate HEV configurations.

Because of the discrete nature of the graphical representations, using optimization algorithms to design graphical structures in a computationally efficient way is a challenging task. Especially when there are many feasibility constraints on the graphical structures, e.g., only certain connections are allowed and these constraints form a disjoint set where feasible designs are dispersed in the design domain, the available derivative-free algorithms fail to find feasible designs. One way to overcome such difficulties is to use enumeration, if possible as in the case of automatic transmission designs, to generate and evaluate all feasible configurations.

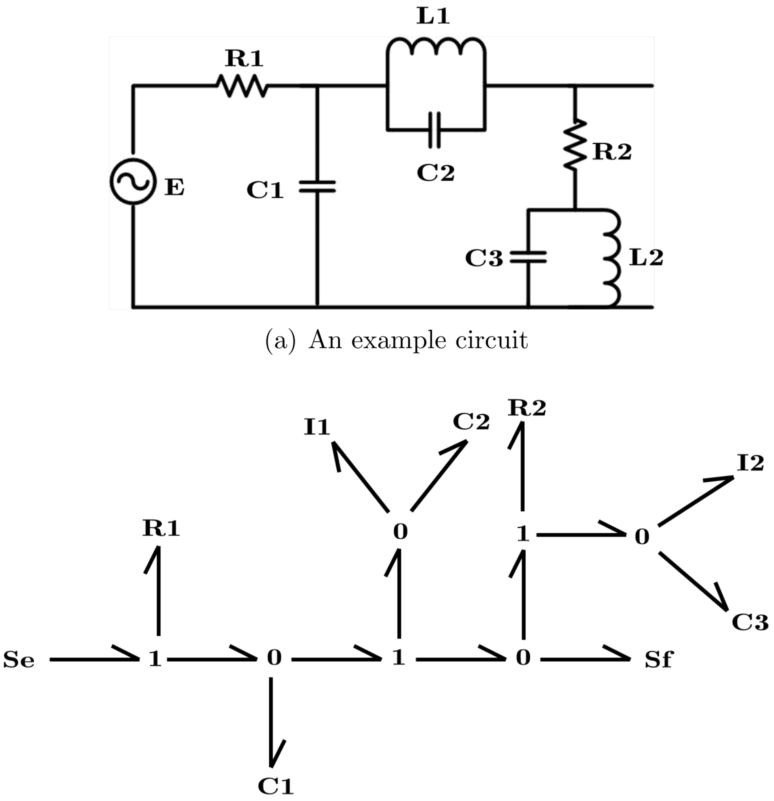
There are some examples from the literature that do not suffer from a highly-constrained design space and therefore successfully implement derivative-free methods for configuration design. One example is from *Koza et al.* (1997). They modeled electrical circuits using a netlist representation and applied Genetic Programming (GP) that evolved the netlist to find circuit configurations giving the desired behavior. A netlist is a graphical representation of a circuit containing information about the type and value of each components, the nodes which each component is connected to. They start with a given catalog of circuit components consisting of capacitors, resistors, inductors, transistors, voltage sources and grounds. Starting from an initial fixed set of components from the catalog that must exist in every circuit configuration, they define "'component-creating functions" that add a new component from the catalog to the circuit, "arithmetic-performing functions" that assign the component values, "topology-modifying functions" that modify the layout of the existing components and "automatically defined functions" that enables reuse of certain structures of the circuit. Then, a GP was used to evolve the netlist by utilizing these four types of functions until the desired functionality is obtained from the evolved circuit. Using that design framework *Koza et al.* (1997) designed circuit configurations for filters, measurement circuits, amplifiers, and circuit performing arithmetic operations.

Genetic Programming worked well in this example mainly because there were not constraints in the problem. For instance, there were no constraints on the connections among components and any component from the catalog was allowed to be connected to any other. The only mechanism to identify bad connections was to evaluate the design and compare the output with the desired behavior. In such a case finding feasible designs is not a difficult task for the evolutionary algorithm.

Another work by *Fan et al.* (2001) on circuit configuration design uses a bond graph representation, which is not based on a geometric arrangement but on the power flow among the circuit components. An example circuit and its bond graph representation is given on Figure 2.2. An advantage of this representation is the ability to identify some bad designs, referred as “singular designs”, using causality rules even before evaluating them. Such singularities correspond to “derivative causalities” in the bond graph representation which can be detected by checking the causalities of the capacitor and inductors easily. Using this representation, similar to *Koza et al.* (1997), they defined some functions to add or modify components and junctions to be used by a GP-based design framework. Starting from a simple initial design, referred as“embryo design”, they generated more complex circuit configurations for analog filters having the desired frequency response.

Since bond graphs are general modeling tools which can be used to model dynamic systems from different domains, a similar idea applies to the general system designs with similar design goals. For instance, a generalized capacitance can both be used to model an idealized capacitor in an electrical circuit and an ideal spring in a mechanical system. The work from *Fan et al.* (2001) was extended to the mechatronic system design by *Seo et al.* (2003). The frequency response targets in the electrical domain correspond to the eigenvalue targets in the mechatronic system design. Starting from a simple embryo design, they applied the same methodology to find more complex designs satisfying the design targets.

An important commonality between the works from *Fan et al.* (2001) and *Seo et al.* (2003) is that all bond graph outputs from GP can be realized by some elementary electrical or mechanical components such as resistors, capacitors, inductors or springs, mass elements and dampers. Their work cannot be applied to the systems where



(a) An example circuit

(b) Bond Graph representation

Figure 2.2: An example circuit and its bond graph representation given by *Fan et al.* (2001)

there some restrictions the bond graph output to make the realization possible. An example for such restrictions can be given from PG based systems. As explained in Chapter III in detail, the bond graph of a PG system must have certain properties related to the transformer moduli or number and type of junctions. If such restrictions form a highly-constrained feasible space with dispersed elements in the feasible space, evolutionary algorithms cannot find any feasible designs other than an initial set of designs provided.

2.3 Summary

In this chapter we reviewed some PG system configuration design studies for both conventional and hybrid vehicles. We argued that the enumeration-based approaches proposed for automatic transmission designs cannot be applied to the general hybrid architecture design problem. In the remainder of the dissertation we will describe cases where the enumeration of all configurations is possible but also discuss larger design problems that make enumeration computationally intractable. We also reviewed some hybrid configuration studies and described their limitations.

Additionally, we showed the application of alternative graphical representation method to electrical and mechatronic configuration designs. Among these alternative representation methods, bond graphs are used to model and generate hybrid architectures. The next chapter describes these representation and generation processes in detail.

CHAPTER III

Hybrid Electric Vehicle Architecture Representation

As discussed in Chapter I, a design synthesis process can be described as composed of four phases as *representation*, *generation*, *evaluation* and *guidance*. In this chapter, we will discuss the representation and generation phases of the HEV architecture design process that will be used by the evaluation and guidance phases to search for good design selections.

In order to find promising architecture candidates among all possible designs, an architecture representation framework that allows the generation of architecture alternatives is necessary. In this chapter, we introduce such a graphical representation. Then, by using the representation we propose a method to generate all feasible driving modes for some given powertrain components. These modes can either form a single-mode HEV architecture alone or can be combined to create multi-mode HEV architectures.

3.1 Means of Architecture Representation

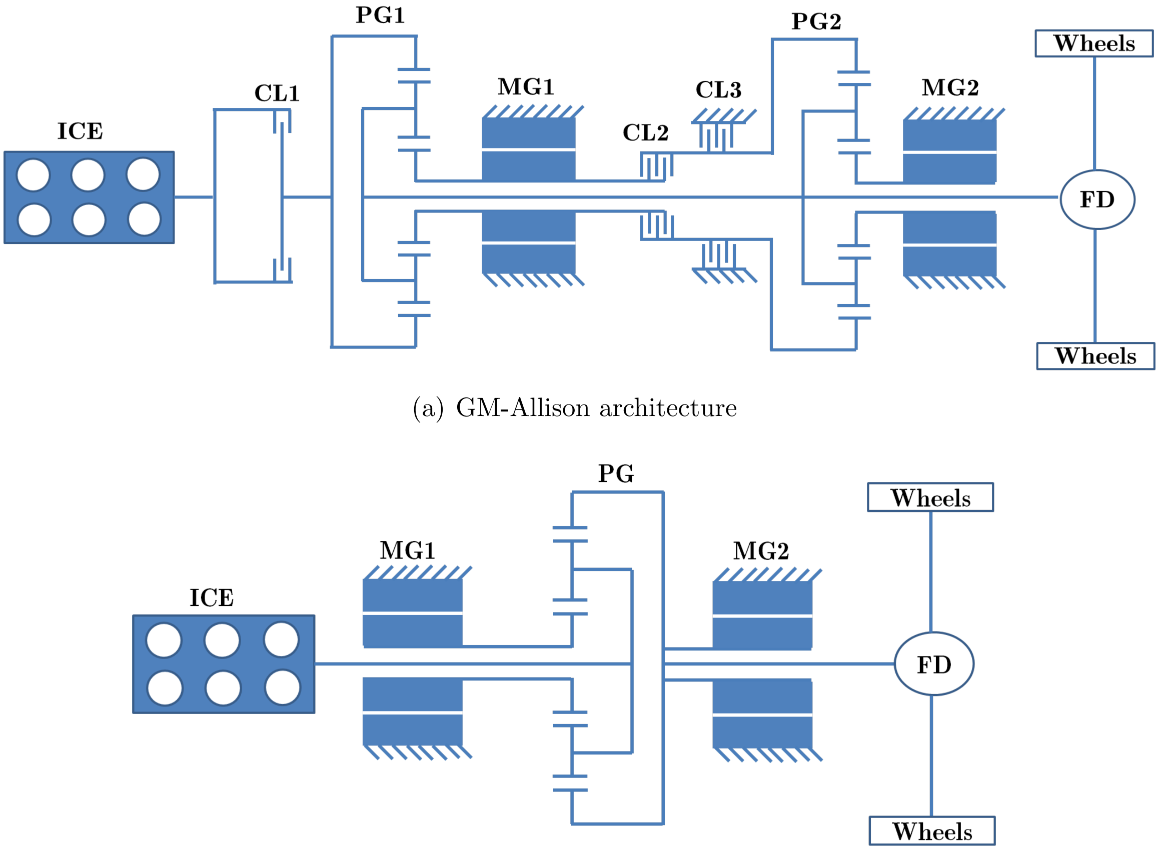
The representation method selected to design the HEV architecture must allow having multiple modes in the architecture. Otherwise the design framework becomes limited to single-mode architectures only. Some available HEV architecture representations show all the modes exist in the architecture. Examples are given later in this section. Since it is possible to perform separate analyses for different modes in the architecture, the representations that only represent a single-mode at a time can still be used for multi-mode architecture design. However, such representations require some extra steps to combine modes in the same architecture.

Another requirement for an HEV architecture representation is the ability to be generalized to any degree of complexity. For instance, representation of a 1-PG architecture should easily be extended to any number of PGs. Also the representation should not be limited by the number or type of powertrain components used in the architecture.

In the remainder of this section, we give a brief overview of available representations used for HEV architectures. Four representation methods of interest are given below in the order of level of abstraction.

3.1.1 Stick Diagram

A stick diagram is the closest representation to the actual realization of the architecture. The diagram is usually drawn as symmetric to its centerline providing some understanding of how shafts are routed in the architecture. The components in the architecture are represented by special icons and the connections by the lines. It is possible to see all different modes in the architecture on the stick diagram representation. Figure 3.1 shows the stick diagram representations of the GM-Allison compound split hybrid architecture (*Holmes and Schmidt*, 2002) and the Toyota Prius architecture. By looking at Figure 3.1(a), one can see that the engine is connected to the ring gear of PG1 via Clutch (CL) 1 and MG1 is connected to the sun gear of the same PG set. MG2 is connected to the sun gear of the PG2 where vehicle shaft is connected to the carrier gear. Both PG sets are connected at the carrier gears. CL2 controls the connection between sun and ring gears of PG1 and PG2, respectively and CL3 grounds or releases the ring gear of PG2. However, there is no information about the gear ratios; therefore, it is not possible to derive the kinematic relationship among powertrain components. So, this representation is useful to describe how the architecture looks in reality but it does not give much information about how the architecture works.



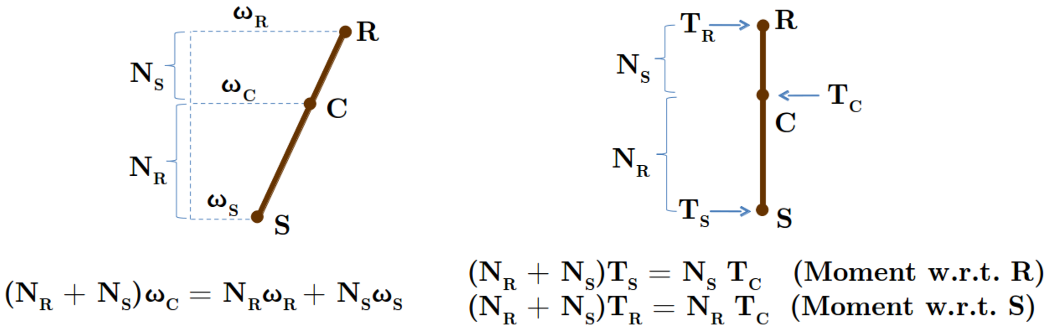
(a) GM-Allison architecture

(b) Toyota Prius Architecture

Figure 3.1: Stick diagram representation of two architectures

3.1.2 Lever Analogy

The analogy between geometric properties of a lever and speed/torque relationships between different gears on a PG was first introduced by *Benford and Leising* (1981). In summary, ring, carrier and sun gears on a PG set are represented as three nodes on a lever as shown in Figure 3.2. On a conventional PG set, the ratio of the vertical distance from carrier to sun and the distance from ring to carrier nodes are equal to the ratio of the number of teeth on the ring gear to sun gear.


ω_R
R
N_S
ω_C
C
N_R
ω_S
S
$(N_R + N_S)\omega_C = N_R\omega_R + N_S\omega_S$
T_R
R
N_S
T_C
C
N_R
T_S
S
$(N_R + N_S)T_S = N_S\ T_C$ (Moment w.r.t. R)
$(N_R + N_S)T_R = N_R\ T_C$ (Moment w.r.t. S)

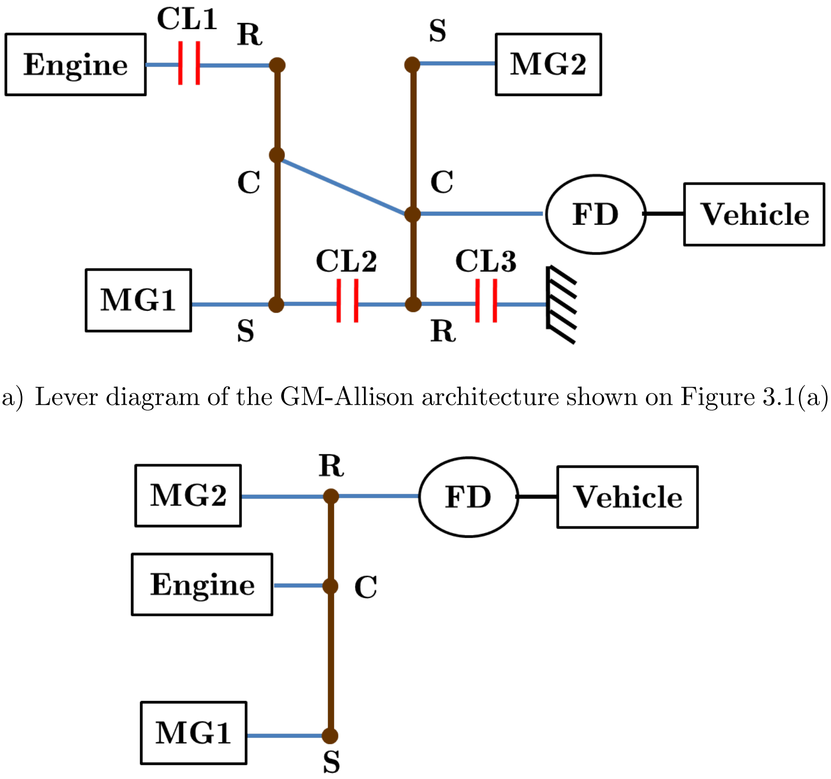
(a) (b)

Figure 3.2: Lever analogy to analyze a PG set

There are two analogies. First, the kinematic relationship among ring, carrier and sun gears is equivalent to the geometric relationship among the horizontal distance of each node on an inclined lever away from a vertical line. Figure 3.2(a) depicts that relationship where R, C and S denote ring, carrier and sun gears, respectively. Also ω_R, ω_R, ω_R denote the speeds and N_R, N_C, N_S denote the radii of each respective gear. The second analogy is related to the torque relationships. As Figure 3.2(b) shows, steady-state torques on each gear, T_R, T_C, T_S can be related by performing a "moment" analysis over torques. Note that, here,"moment" does not correspond to any physical quantity but the analogy is used just to make analysis easier.

Since every PG set can be modeled by a lever, the stick diagram can be converted into a lever diagram by replacing the PG boxes by the corresponding lever. As a result all available modes can be shown on the lever diagram as well. Figure 3.3 shows the lever representation of two example architectures.

If multiple PGs are connected in the architecture, the lever analogy allows combining levers to simplify the representation in which case the lever is called *compound lever*. Examples of such cases can be found in *Benford and Leising* (1981). In such



(a) Lever diagram of the GM-Allison architecture shown on Figure 3.1(a)

(b) Lever diagram of the Toyota Prius architecture

Figure 3.3: Architecture examples

cases, it is not straightforward to visualize the stick diagram from the lever representation. The lever diagram is a higher level architecture abstraction than the stick diagram.

3.1.3 Bond Graphs

Bond graphs are graphical modeling tools to represent the power flow in a dynamic system (*Karnopp et al.*, 2012; *Rosenberg and Karnopp*, 1972; *Rosenberg*, 1971). They are used as an additional representation method to the usual equations for dynamic system modeling. In bond graphs, power flow is represented by a *bond* between two nodes and power is modeled by two quantities: *flow* and *effort*. Since these are generic terms to model a system, the meaning of flow and effort vary depending on the domain of analysis. For instance, in an electrical system, flow corresponds to current and effort to voltage. In a mechanical system, specifically in rotational systems, flow and effort mean angular speed and torque, respectively. The direction of a bond indicated by a half arrow at one end is used to determine the sign convention. Positive power flows towards the direction of the bond. The *causality strokes* indicated by a line at one end of the bond are used to model the direction of the flow and effort. Flow on a bond is always pointing away from a causality stroke while effort points the opposite direction.

Two types of power sources, namely, *flow source* and *effort source*, supply energy to the system. There are generalized *capacitor* (C) and *inductor* (I) elements modeling the energy storage while a generalized *resistor* (R) element models energy dissipation. *Transformer* (TF) blocks relate flows and efforts at both ports by a *transformer modulus*, keeping power equality at two ports. Similarly, a *gyrator* (GY) relates efforts to flows and vise versa by a *gyrator modulus*. Also, two types of *junctions*, which are simply the connection points of the bonds, define certain relationships among flow and efforts around them. At a 0-junction, the sum of the flows is zero while efforts are the same; at a 1-junction, the sum of the efforts is zero and flows remain the same. More details of this modeling tool can be found in *Karnopp et al.* (2012).

This graphical modeling tool can be used to represent HEV architectures, as well. As an example, Figure 3.4 depicts a bond graph representation of the Toyota Prius architecture given on Figure 3.3(b) ignoring gear losses and inertias. Note that only a single mode is modeled in this representation. Separate bond graph models must be built for different modes in a multi-mode architecture.

3.1.4 System of Equations

The most common representation of a system is a set of equations. In the conventional way, in order to describe the system by equations, one needs to break down the

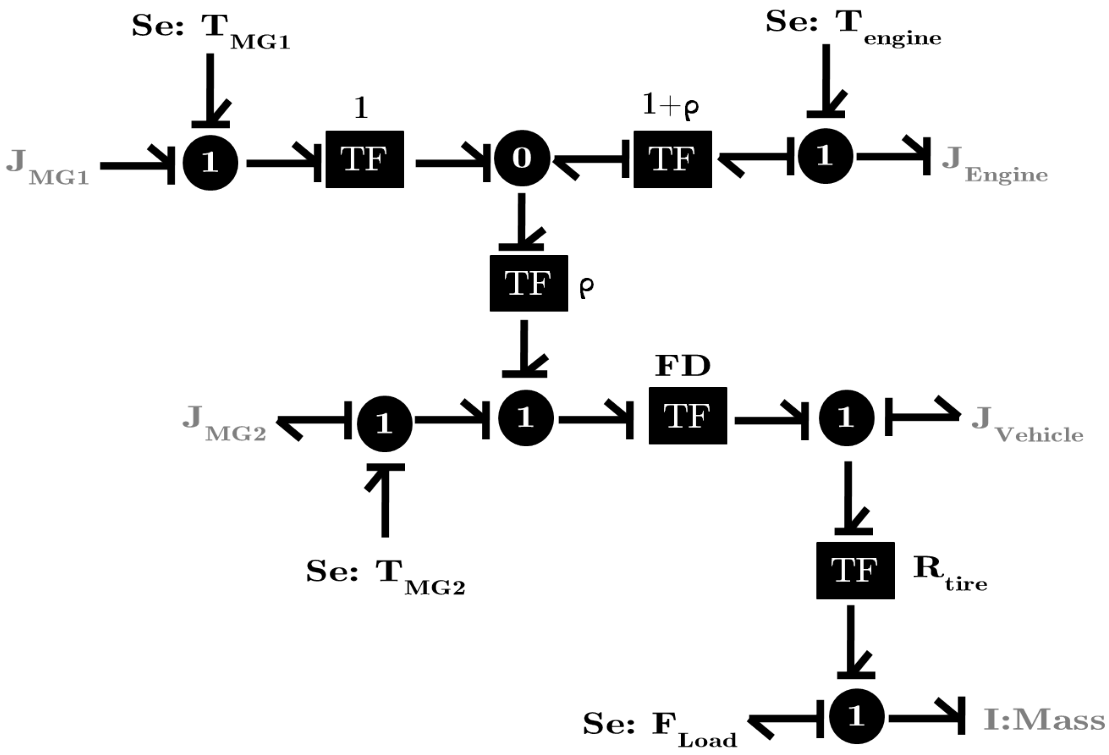


Figure 3.4: Bond graph representation of Toyota Prius architecture with ring to sun gear ratio of ρ ignoring the gear inertias and losses

system into its elements, draw free body diagrams for each element and combine the equations of motion obtained from each element. Note that it is possible to extract a system of equations from a bond graph model. The details of the link between a system of equations and the bond graph model are given later in Section 3.2.4. The following subsections build the dynamic and static model of the Toyota Prius architecture shown on Figure 3.3(b)

3.1.4.1 Dynamic model

Details of the free body diagrams are omitted here. Three equations are written for ring, sun and carrier gears accounting for the torques coming from engine, MGs and vehicle body (resistive torque) as well as the internal force on the gear set. A fourth equation that relates the speed of each gear to each other yields the following model.

$$\begin{bmatrix} J_{out}+J_{MG2} & 0 & 0 & -N_R \\ 0 & J_{eng} & 0 & N_R+N_S \\ 0 & 0 & J_{MG1} & -N_S \\ -N_R & N_R+N_S & -N_S & 0 \end{bmatrix} \begin{bmatrix} \dot{\omega}_R \\ \dot{\omega}_C \\ \dot{\omega}_S \\ F \end{bmatrix} = \begin{bmatrix} T_{MG2}-T_{res} \\ T_{eng} \\ T_{MG1} \\ 0 \end{bmatrix} \tag{3.1}$$

where J_{eng}, J_{MG1}, J_{MG2} and T_{eng}, T_{MG1}, T_{MG2} denote the inertias and torques of engine and MGs, respectively. Also, J_{out} denotes the inertia term coming from the vehicle output and final drive, and T_{res} is sum of the resistive terms from the vehicle body. Finally the term F is the internal force on the planetary gear set and usually not of importance in powertrain simulations.

3.1.4.2 Quasi-static model

In that approach, instead of drawing the free body diagrams, the formulation starts by writing down the kinematic relationship among the gears as shown in Equation (3.2).

$$(N_R+N_S)\omega_C = N_R\omega_R + N_S\omega_S \tag{3.2}$$

Denoting $N_R/N_S = \rho$ and substituting the carrier, ring and sun gear speeds by engine, MG2 (also equal to the output speed) and MG1 speeds give the relationship in Equation (3.3)

$$\begin{bmatrix} 1+\rho & -\rho \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_{eng} \\ \omega_{out} \end{bmatrix} = \begin{bmatrix} \omega_{MG1} \\ \omega_{MG2} \end{bmatrix} \tag{3.3}$$

The 2x2 matrix on the left is a mode-specific kinematic matrix. In the rest of this section, that matrix will be denoted as $\mathbf{C}_{mode}$. At steady state, assuming a lossless gear set, the following power relationship must hold.

$$P_{eng} + P_{MG1} + P_{MG2} = P_{out}, \tag{3.4}$$

where P_{eng}, P_{MG1} and P_{MG2} are the powers from engine and MGs, and P_{out} is the total driving power applied on the vehicle output. Substituting each power term by the corresponding $T \cdot \omega$ expression, the following relationship can be obtained.

$$T_{eng} \cdot \omega_{eng} + T_{MG1} \cdot \omega_{MG1} + T_{MG2} \cdot \omega_{MG2} = T_{out} \cdot \omega_{out} \tag{3.5}$$

Since the relationship among the speeds is known from Equation (3.3), it is possible to obtain a relationship among the torques:

$$-\mathbf{C}_{mode}^{-T} \begin{bmatrix} T_{eng} \\ -T_{out} \end{bmatrix} = \begin{bmatrix} T_{MG1} \\ T_{MG2} \end{bmatrix}. \tag{3.6}$$

In the dissertation, as part of the hybrid powertrain architecture design process, we use a “modified bond graph” representation to generate architecture alternatives, and quasi-static system of equations are used to evaluate an architecture. The details of the modified bond graph representation is given in Section 3.2. The quasi-static model that describes the steady state behavior of the system requires very small simulation (computation) time while sacrificing the system transients. In an HEV powertrain, since the time constants of the engine and MG inertias are small compared to the time constant of the vehicle output shaft, the engine and motor transients fade quickly. As a result, if the purpose of the analysis is to calculate fuel and battery consumption, these transients have only little impact on the final results.

As it is seen in Equations (3.3) and (3.6), in the quasi-static model the kinematic matrix alone gives all the information needed to evaluate the system. This important fact is extensively used in the HEV architecture design process. Another important observation is the number of degrees of freedom (dof) being equal to two for a hybrid mode, i.e., only two independent variables are needed to determine the other variables in the system. This is useful especially in the supervisory control of the HEV powertrain described in Chapter V. When a drive cycle is given, the state of the vehicle output shaft following the drive cycle are determined by the driver model. In a 2-dof hybrid mode the engine state remains as the control variable. In a 3-dof hybrid system one of the motor states becomes also a control variable but such systems would require sophisticated control methods. 1-dof hybrid systems such as parallel hybrids exist in the market but since engine state is dependent on the state of the vehicle output shaft, there is less room for control. They require an additional transmission in order to provide a range of operating points for the engine to select from.

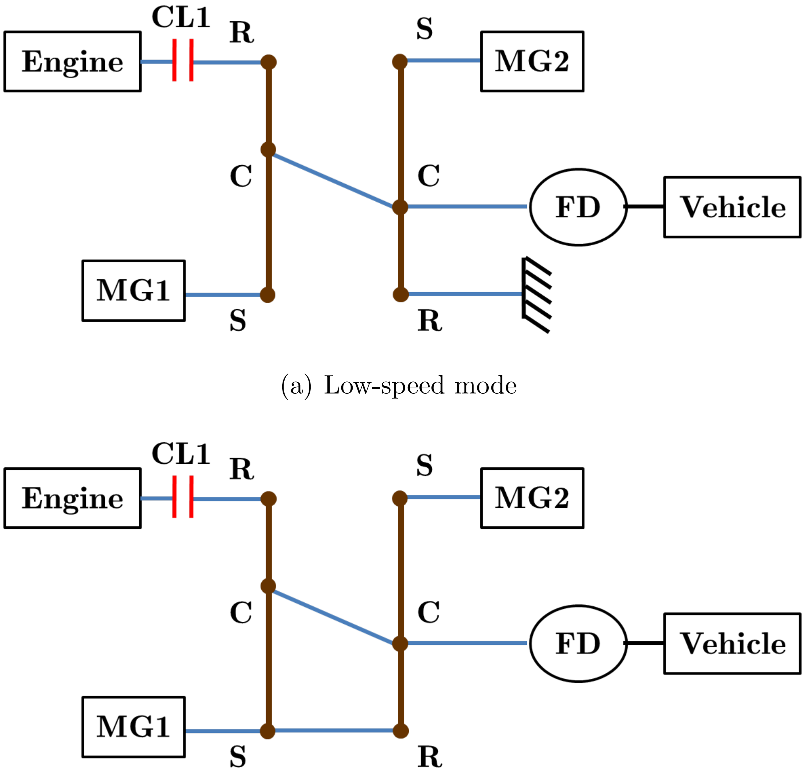
A similar argument can be made for pure electric modes. A kinematic matrix describing the relationship between electric motors and vehicle power output is used to evaluate the design. 1 and 2-dof pure electric modes are considered in this dissertation.

The next section explains the use of these tools to generate architecture alternatives.

3.2 Representation and Generation of Architectures

A key contribution of this dissertation is the creation of architecture representation tools that allow generation of architecture alternatives where an architecture is created by combining driving modes. For instance, the architecture shown in Figure 3.3(a) can be obtained by combining the modes shown in Figure 3.5. Given some external components and the number of PGs to use, the number of possible driving modes is limited and the modes can be enumerated. However, the number of architecture alternatives for a multi-mode architecture increases exponentially with the number of modes, i.e., combining M possible driving modes in an architecture with N_{mode} number of modes yields $M^{N_{mode}}$ architecture alternatives that is usually a large number. Thus, an exhaustive enumeration of all possible architectures is not a computationally efficient effort. However, the number possible driving modes is relatively small and they can be enumerated. Then a smart method to combine the driving modes in the same architecture can be used to design a multi-mode architecture.

One can argue that a derivative-free search algorithm could be applied to a given representation in order to find promising architectures. However, as it will be described later in this section, there are many design constraints to ensure a graphical representation corresponds to an actual mode. These constraints define not only a small but also a highly-dispersed feasible space. For such problems, the available derivative free methods cannot find feasible designs other than any initial feasible designs provided. Hence, we propose the enumeration of feasible driving modes to create a new design space consisting of feasible designs only.



(a) Low-speed mode

(b) High-speed mode

Figure 3.5: Two modes of the architecture shown on Figure 3.3(a). CL1 is only used to obtain a neutral gear instead of a new mode.

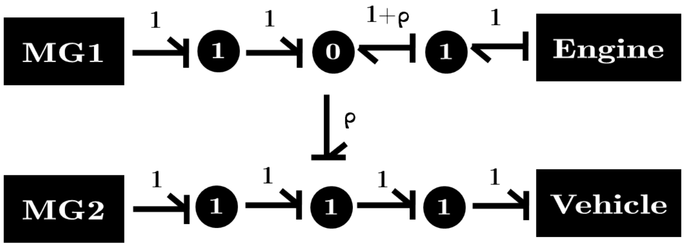


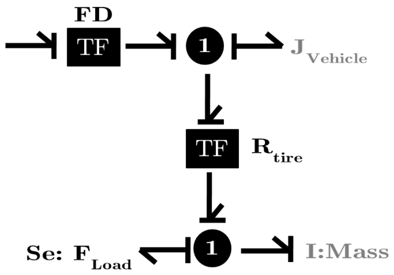
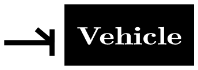
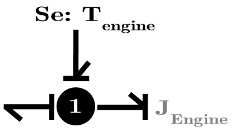
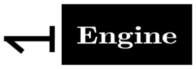
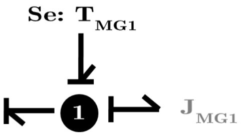
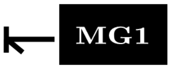
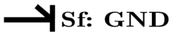
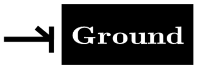
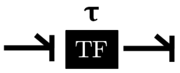
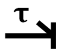
Figure 3.6: Modified bond graph representation of Toyota Prius architecture

In this section, a method to enumerate possible driving modes is introduced while the evaluation of these modes in a single-mode architecture is given in Chapter V and in a multi-mode architecture is given in Chapter VI. Both hybrid modes, referred to as HEV modes, and pure electric modes, referred to as EV modes, are considered in this section. In an HEV mode both engine and MGs are actively running while in a pure EV mode the engine is not active. Engine can be deactivated by either disconnecting the engine from the architecture or connecting engine to a ground node.

To start with, we introduce some simplifications to the bond graph representation. First, it is possible to group the fixed components that appear in every design. Since the connections to the power sources and vehicle output are the same at every possible design candidate, they can be represented as a box left out of the design domain. Also TF blocks can be represented as weights on the bonds to make visualization simpler. If there is no transformer block between two junctions, a default bond weight of 1 is assigned. Table 3.1 summarizes those simplifications and Figure 3.6 shows the "modified bond graph" representation of the Toyota Prius architecture model on Figure 3.4.

After these simplifications, we refer to the junctions connected to the external components as *external junctions* and the rest of the junctions we call *internal junctions*. External components can be engine, MGs, vehicle output or grounds. The number of external junctions is equal to the number of external components since every external component is assumed to be connected to a unique external junction.

Table 3.1: List of Bond Graph simplifications

Actual Model	Simplified
 FD TF 1 J Vehicle TF R tire Se: F Load 1 I:Mass	
	
	
	
	

The number of internal junctions in the bond graph representation of a driving mode depends on the existence of a ground connection, the number of external junctions and the number of PGs considered in the design.

In order to generate all possible modified bond graphs corresponding to feasible driving modes, all possible simple, connected, undirected graphs are generated first. Then, junction types and causalities are assigned to convert the undirected graphs to bond graphs. Finally, the graph weights are assigned to the bonds in order to represent a PG system. Once a modified bond graph structure corresponding to a driving mode is created, the quasi-static system of equations is extracted to be able to evaluate the designs within a vehicle model. The following subsections elaborate on each of these steps. Our references to bond graphs relate to the modified bond graph described in this section.

3.2.1 Graph Enumeration

This section discusses the enumeration of all undirected graphs for external junctions and internal junctions. We introduce some graph properties, first.

Bond Graph Property 1. *A valid bond graph must be simple and connected.*

A simple graph does not have more than one bond between any pair of junctions and contains no loops around a single junction, i.e., a junction is not directly connected to itself. Figure 3.7 shows those two cases. Also, on a connected graph no junction remains disconnected. Therefore, each external junction is connected to an internal junction, representing some power flow from (or to) the sources to (or from) the architecture.

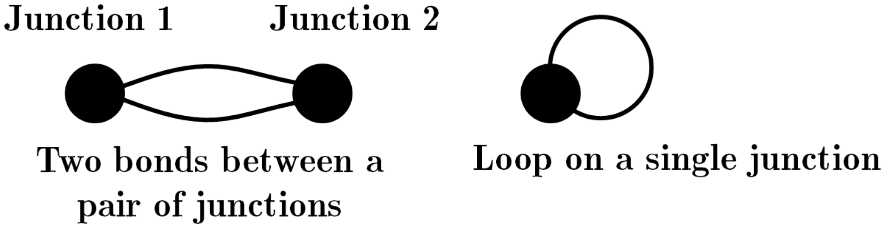

Junction 1
Junction 2
Two bonds between a pair of junctions
Loop on a single junction

Figure 3.7: Violation of simple graph assumption

In a general bond graph representation, a junction can have any number of bonds connected to it. However, in this study, the number of bonds is restricted to three for an internal junction and to one for an external junction for simpler implementation. The following property shows that such a restriction does not limit any modeling capabilities.

Bond Graph Property 2. *A junction with $n > 3$ bonds can equivalently be replaced by $n - 2$ junctions of the same type with 3 bonds for each.*

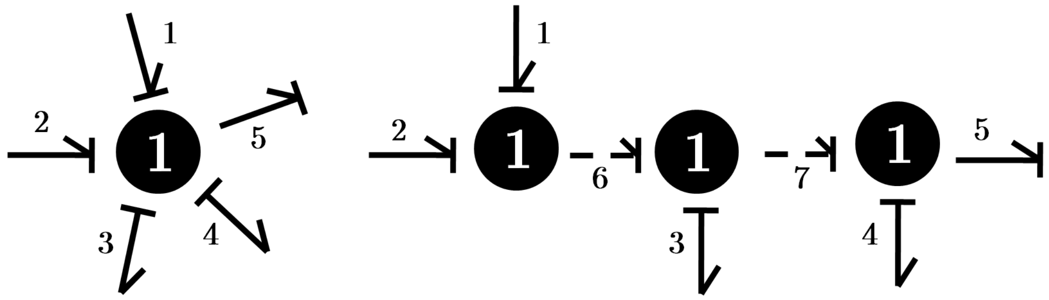


Figure 3.8: A junction with 5 bonds is equivalently replaced by 3 junctions with 3 bonds each

Figure 3.8 explains this property for a 1-junction with 5 bonds. As it can be seen in the figure, in a chain of bonds both first and last junctions will have 2 bonds from the original model, leaving $n-4$ bonds for the remaining junctions. Each intermediate junction can only have 1 bond from the original model, since 2 junctions are needed to connect to other junctions. In total $2 + (n - 4) = n - 2$ junctions are needed to represent the original model. Addition of these intermediate junctions and bonds does not change the overall system of equations.

Using the two properties given before, a new property related to the number of junctions in a bond graph representation of a mode can be obtained as follows.

Bond Graph Property 3. *A bond graph must have an even number of junctions if each internal junction has three bonds and each external one has one bond.*

From the Bond Graph Properties 1 and 2, a bond graph with J^{ext} external junctions and J internal junctions has $(3J - J^{ext})/2 + J^{ext}$ bonds. Since the number of bonds has to be an integer, $J + J^{ext}$ must be even.

Recall that the external junctions are connected to the external components and the number of external components is usually small. In a typical hybrid powertrain, the external components are an engine, two MGs, one vehicle output and a ground. For these external components, Table 3.2 summarizes the number of junctions needed for 1-PG, 2-PG and 3-PG systems. Modes are classified in four types and examples of each type is given on Figure 3.12. With an additional bond graph property to be discussed later this table can be extended to any number of PGs and any number of external components. However, it is unlikely to consider designs with more than three PGs and more than six external components as the cost and packaging constraints make them impractical.

Table 3.2: Number of junctions needed for different types of modes

Number of PGs	HEV Mode without Ground	HEV Mode with Ground	Pure EV Mode with Engine	Pure EV Mode without Engine
1-PG	$J^{ext} = 4$, $J = 2$	– –	$J^{ext} = 5$, $J = 3$	$J^{ext} = 3$, $J = 1$
2-PG	$J^{ext} = 4$, $J = 4$	$J^{ext} = 5$, $J = 3$	$J^{ext} = 5$, $J = 3$ or 5	$J^{ext} = 3$, $J = 3$
3-PG	$J^{ext} = 4$, $J = 6$	$J^{ext} = 5$, $J = 5$	$J^{ext} = 5$, $J = 5$ or 7	$J^{ext} = 3$, $J = 5$

With this background information, we can discuss the process to enumerate all possible graphs for a given number of PGs. Let $\mathbf{G}_{(J^{ext}+J)\times(J^{ext}+J)} = [\mathbf{0}\ \mathbf{A}; \mathbf{A}^{\mathbf{T}}\ \mathbf{B}]$ be the adjacency matrix representing a graph where $\mathbf{A}_{J^{ext}\times J}$ and $\mathbf{B}_{J\times J}$ are binary matrices. The matrix $\mathbf{A}$ represents the connections between external and internal junctions and $\mathbf{B}$ represents the connections among internal junctions. The corresponding value in the matrix is 1 when there is a connection between two junctions and 0 otherwise. As a result, the following two observations can be made:

(i) If the junctions i and j are connected, $B_{ij} = B_{ji} = 1$. Then, the matrix $\mathbf{B}$ is symmetric.

(ii) A $\mathbf{0}_{J^{ext}\times J^{ext}}$ block prevents any direct connection among external junctions. However, it does not restrict any indirect connections among external junctions. For instance, both engine and MG1 can be connected to the same gear in a PG.

The designs of the matrices $\mathbf{A}$ and $\mathbf{B}$ determine the graph under the following constraints:

(i) An external junction must have only one bond:

$$\sum_{j=1}^{J} A_{ij} = 1, \qquad \forall i = 1, 2, 3, ..., J^{ext} \tag{3.7}$$

(ii) Each internal junction must have exactly 3 bonds:

$$\sum_{i=1}^{J^{ext}} A_{ij} + \sum_{i=1}^{J} B_{ij} = 3, \qquad \forall j = 1, 2, 3, ..., J \tag{3.8}$$

In order to enumerate all possible graphs satisfying the constraints given by Equation (3.7) and Equation (3.8), all possible $\mathbf{A}$s are generated from Equation (3.7) and for each generated $\mathbf{A}$, all possible $\mathbf{B}$s are found from Equation (3.8). Since J^{ext} and J do not take large values, the computational cost of this process is small.

The enumeration process described above might create replicated graphs with different ordering of junctions. Figure 3.9 shows such a case. By changing the order of the junctions, the same systems can be obtained from each other. This well-known phenomenon is referred as *graph isomorphism* in graph theory (*Read and Corneil*, 1977). Isomorphism exists in any graphical representation and must be identified to eliminate the replicates. Mathematically, two graphs represented by their adjacency matrices G_1 and G_2 are considered to be the isomorphic if and only if there is a permutation matrix P which satisfies the following:

$$G_2 = P\ G_1\ P^T \tag{3.9}$$

In other words, the adjacency matrices of isomorphic graphs can be obtained by reordering the rows and columns simultaneously. It has been known that two non-isomorphic graphs have distinct eigenvalues but the converse is not true (*Harary*

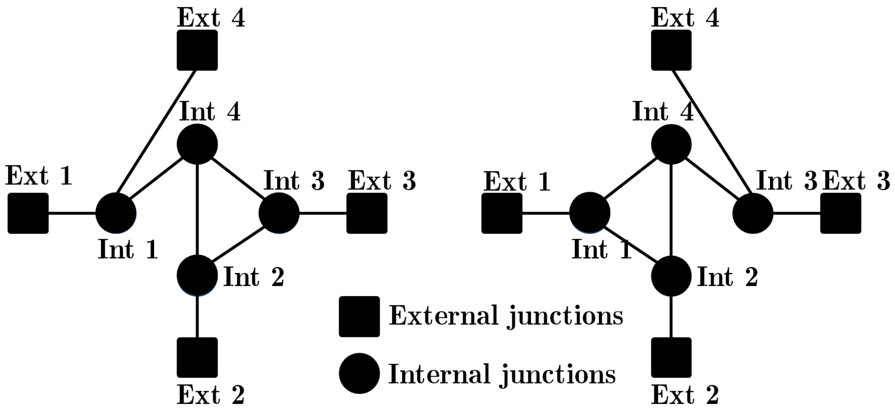


Figure 3.9: Two replicates generated from the enumeration process. Both graphs result in the same equation sets after assigning the junction type and bond weights

et al., 1971). It is possible to find non-isomorphic graphs with the same eigenvalues. It is called *cospectral non-isomorphism*. So, if the eigenvalues are used alone to detect isomorphism, some non-isomorphic graphs could alos be filtered in the process. There are some algorithms introduced in the literature to identify graph isomorphism (*Fortin*, 1996; *McKay et al.*, 1981). We use the implementation from *MATLAB* (2014) to filter the isomorphic graphs.

3.2.2 Junction Type and Causality Assignment

After the enumeration of all undirected graphs with no replicates, it is necessary to assign 0 and 1-junctions. This assignment process requires special attention to the causality stroke assignment since the causality restricts the junction assignment. The process starts with the external junction assignment. All external junctions are assigned to be 1-junctions. We also need to assign causalities to the external junctions. It can be seen from Figure 3.6 that for a hybrid mode the causality stroke is on the internal junction side for MG1 and MG2, and on the external junction side for engine, vehicle output. These assignments are done considering how the control strategies work. The details of the control strategies used for HEV vehicles are given in Chapter

V. We assume that the vehicle speed is imposed by the drive cycle and the causality is assigned to be on the external junction side. We find the best engine speed which gives the best fuel consumption and engine imposes this best speed determined by the controller. The speeds of the MGs are imposed by the kinematic relationship. The causality of the ground is always on the external junction side since ground always impose the zero speed.

For a pure electric modes the causality assignment of the engine (if exists) is opposite since the engine is grounded. The ground imposes the zero speed to the engine. The causality of one of MGs is also flipped as necessary to make the assignment consistent, i.e. the speed of one of the MGs is freely determined by the controller if the MG is not grounded. For instance, pure EV modes if causality of MGs are both on the external junction side we obtain 1-dof EV modes. If the causality of one the MGs is on the internal junction side, we obtain 2-dof EV modes. When engine connected to the grounded gear, we can obtain both 1-dof and 2-dof modes but when engine is removed from the external junctions, only 1-dof modes give feasible bond graph structures. Both 1-dof and 2-dof EV modes are included in this dissertation. (Note that in this dissertation, dof is used in a kinematic sense. 1-dof mode means that resultant PG system has only one independent speed to determine all other speed values in the system.)

The causality assignment employed here restricts some of the configurations. For instance, since engine and vehicle output have the same causality, they can never be connected to the same gear of a PG set directly. However, such a configuration links the engine speed to the vehicle speed directly, leaving no possibility to control engine speed. Thus, such less efficient designs are not considered in this configuration generation stage. When the designs of interest include parallel configurations, engine must be allowed to be connected to the same node of PG set by flipping the causality of the engine. Recall that, in order to allow some variety of speed ratios, this design requires an additional transmission at the vehicle output.

Some definitions are necessary before explaining the assignment process further. Let $\beta = (3J - J^{ext})/2$ denote the number of bonds connecting the internal junctions to each other. Also denote the junction type for the junction j as t_j. Let $t_j = 1$ when the junction type is 1 and $t_j = -1$ when the junction type is 0. This assignment is useful when formulating the junction type assignment problem. Denote the causality on the bond connecting the junctions j_1 and j_2 as $c_{j_1 j_2}$ and assign $c_{j_1 j_2} = 1$ if the stroke is on the junction j_1 side, implying $c_{j_2 j_1} = -1$. In With these definitions and according to the bond graph rules, the following equation is stated:

$$-t_{j_1} + \sum_{j_2 | G_{j_1 j_2} = 1} c_{j_1 j_2} = 0, \qquad \forall j_1 = 1, 2, 3, ..., J. \tag{3.10}$$

Since the total number of equations is J and the number of unknowns to be determined is $\beta + J$, multiple solutions exist.

One last property of the bond graphs we introduce helps to list all the solutions to the Equation (3.10).

Bond Graph Property 4. *The number of 0-junctions is equal to the number of PGs in the system.*

Let J_0 and J_1 denote the number of 0 and 1 junctions where $J_0 + J_1 = J$. Summing all the equations from Equation (3.10) gives:

$$\sum_{j_1=1}^{J} (-t_{j_1} + \sum_{j_2 | G_{j_1 j_2} = 1} c_{j_1 j_2}) = 0. \tag{3.11}$$

The sum of all causalities between internal junctions is zero since for every $c_{j_1 j_2}$ there exists a $c_{j_2 j_1}$ such that $c_{j_1 j_2} + c_{j_2 j_1} = 0$. Also, the causalities on the bonds from the engine and vehicle output cancel the ones from MG1 and MG2. If there is a ground engaged in the mode, a causality of -1 remains. As a result, the following two cases appear:

(i) If ground is not engaged, then

$$\sum_{j_1=1}^{J} t_{j_1} = 0 \tag{3.12}$$

From the definition of t_j, it can be concluded that $J_0 = J_1$ in this case.

(ii) If a ground is engaged, then

$$\sum_{j_1=1}^{J} t_{j_1} = -1 \tag{3.13}$$

In this case $J_0 = J_1 + 1$.

The number of junctions for different types of modes obtained using all the bond graph properties are given in Table 3.2. It can be seen that there is no feasible mode obtained for 1-PG hybrid system with a ground. Since there are only three nodes on a single PG, using one engine, two motor generators, one vehicle output and a ground it is not possible to obtain meaningful 2-dof hybrid design. Also for pure EV modes with engine connected to the grounded gear, the number of internal junctions can take two different values. It is mainly due to the causality assignment of the MGs. If one of the MGs is assigned to impose flow on the system, we need less internal junctions to make the bond graph feasible.

3.2.3 Bond Weight Assignment

In order to complete the modified bond graph enumeration process, the proper bond weights (weights for the TF blocks) need to be assigned. Let ρ be the ring to sun PG ratio. As Figure 3.4 also shows, a PG set is modeled with 0 and 1-junctions together, with weights around a 0-junction being 1, ρ and $1 + \rho$, and the rest of the

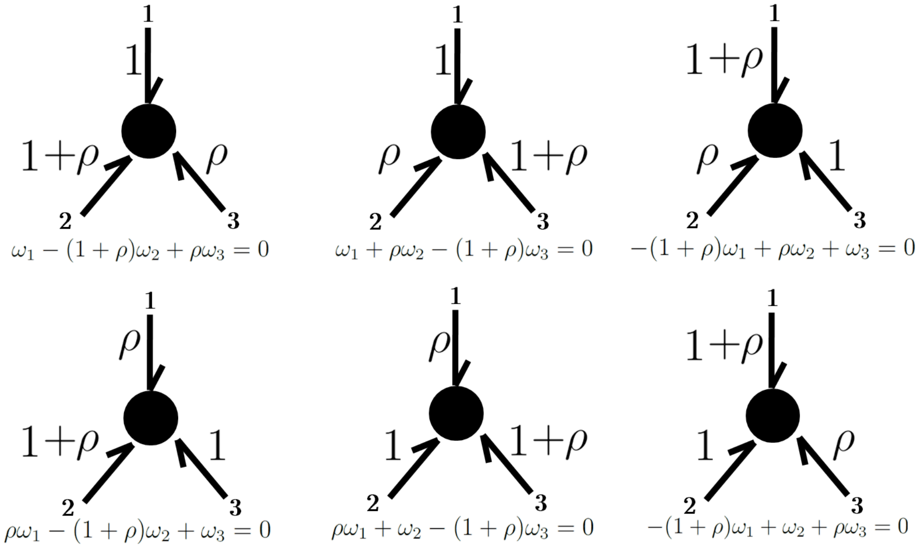

1
1
1+ρ
ρ
2
3
$\omega_1 - (1+\rho)\omega_2 + \rho\omega_3 = 0$
1
1
ρ
1+ρ
2
3
$\omega_1 + \rho\omega_2 - (1+\rho)\omega_3 = 0$
1
1+ρ
ρ
1
2
3
$-(1+\rho)\omega_1 + \rho\omega_2 + \omega_3 = 0$
1
ρ
1+ρ
1
2
3
$\rho\omega_1 - (1+\rho)\omega_2 + \omega_3 = 0$
1
ρ
1
1+ρ
2
3
$\rho\omega_1 + \omega_2 - (1+\rho)\omega_3 = 0$
1
1+ρ
1
ρ
2
3
$-(1+\rho)\omega_1 + \omega_2 + \rho\omega_3 = 0$

Figure 3.10: All possible six combinations for the bond weight assignment around a 0 junction

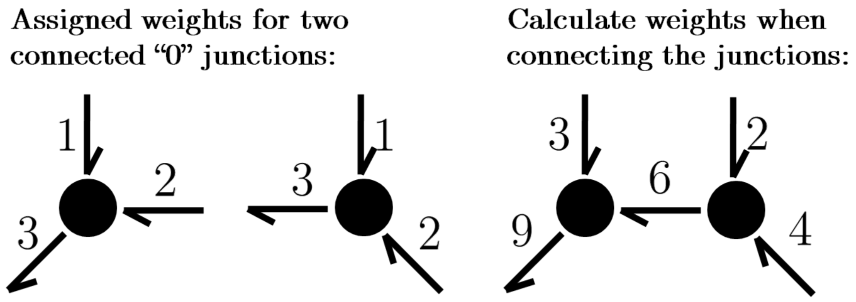

Assigned weights for two connected "0" junctions:
1
2
3
3
1
2
Calculate weights when connecting the junctions:
3
2
6
9
4

Figure 3.11: Bond weight scaling for a 0 to 0 junction connection

bond weights are 1. Figure 3.10 shows all possible six combinations for the bond weight assignment around a 0-junction.

Implementation of these six combinations must not result in any inconsistency with the bond weights. When connecting two 0-junctions together, scaling of the bond weights might be necessary to prevent such inconsistencies. Figure 3.11 shows an example for a 0-to-0 junction connection when scaling is needed.

3.2.4 Equation Generation

Once the enumeration process is complete, the link between the graph representation and the system equations need to be formed in order to evaluate the designs. This section describes how to extract the necessary equations from a modified bond graph. Recall from Section 3.1.4 that only the matrix $\mathbf{C}_{mode}$ that defines the kinematic relationship among the components is needed to analyze a mode.

Let ω^{ext} be the vector of speeds from external components. Define also ω as the $\beta \times 1$ vector of speeds on the bonds connecting internal junctions. Following the bond graph rules for 0-junctions and 1-junctions, i.e., speeds around a 0-junction sum up to 0 and speeds around a 1-junction are the same, the following system of equations can be written,

$$\mathbf{W_0}\omega^{ext} + \mathbf{W}\omega = 0 \tag{3.14}$$

where $\mathbf{W_0}$ and $\mathbf{W}$ are the matrices containing the bond weights from the graph.Recall that J_0 and J_1 are the number of 0-junctions and 1-junctions, respectively. Then, $\mathbf{W}$ is a matrix of the size $(J_0+2J_1)\times\beta$ since a 0-junction defines a single relationship for the speeds and 1-junction defines two. Bond Graph Property 4 implies that $J_0+2J_1 > \beta$. So $\mathbf{W}$ has more rows than columns. When $\mathbf{W}$ has the rank β, Equation (3.14) can be rewritten as,

$$\omega = -(\mathbf{W^T W})^{-1}\mathbf{W}^T\mathbf{W_0}\omega^{ext} \tag{3.15}$$

Substituting Equation (3.15) into Equation (3.14) gives the following equation.

$$(\mathbf{W_0} - \mathbf{W}(\mathbf{W^T W})^{-1}\mathbf{W^T W_0})\omega^{ext} = \mathbf{0} \tag{3.16}$$

Call $\widetilde{\mathbf{W}} = (\mathbf{W_0} - \mathbf{W}(\mathbf{W^T W})^{-1}\mathbf{W^T W_0})$. This matrix of size $(J_0 + 2J_1) \times J^{ext}$

contains all the linear kinematic equations relating all external components. The rank of $\widetilde{\mathbf{W}}$ that determines the dof of the PG system is ranging from 1 to 3. As pointed out earlier, only 1 and 2-dof systems are of interest in this study. After this step, separate analyses need to be performed for hybrid and pure electric modes.

(i) For hybrid modes, in order to obtain the kinematic relationship matrix mentioned in Section 3.1.4.2 from Equation (3.16), a rearrangement is necessary. Consider a specific case with $\omega^{ext} = [\omega_1, \omega_2, \omega_3, \omega_4]$ being the vector of speeds from engine, vehicle output, MG1 and MG2. Let $[\widetilde{\mathbf{W}}_1, \widetilde{\mathbf{W}}_2] = \widetilde{\mathbf{W}}$ where both $\widetilde{\mathbf{W}}_1$ and $\widetilde{\mathbf{W}}_2$ are of the size $(J_0 + 2J_1) \times 2$. When both $\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_1$ and $\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_2$ are invertible, rearrangement on Equation (3.16) gives,

$$\begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix} = -(\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_2)^{-1}(\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_1) \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \tag{3.17}$$

where the kinematic relationship matrix is $\mathbf{C}_{mode} = -(\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_2)^{-1}(\widetilde{\mathbf{W}}_2^T\widetilde{\mathbf{W}}_1)$.

In the case where ω^{ext} contains speeds from ground nodes, the same idea holds after an additional step. Since the speed of a ground node is always zero, columns of the matrix $\widetilde{\mathbf{W}}$ to be multiplied by the ground speed can be removed. In that case, the definition of $\widetilde{\mathbf{W}}_1$ and $\widetilde{\mathbf{W}}_2$ must be modified as $[\widetilde{\mathbf{W}}_1, \widetilde{\mathbf{W}}_2, \widetilde{\mathbf{W}}_{gnd}] = \widetilde{\mathbf{W}}$ where $\widetilde{\mathbf{W}}_{gnd}$ are the columns to be removed. With that definition, Equation (3.17) holds for the cases including ground nodes.

(ii) For pure electric modes, if engine exists in the mode, the following definitions can be made $[\widetilde{\mathbf{W}}_{eng}, \widetilde{\mathbf{W}}_1, \widetilde{\mathbf{W}}_{gnd}] = \widetilde{\mathbf{W}}$ where $\widetilde{\mathbf{W}}$ is the column multiplied by engine speed. Since engine is always grounded in a pure electric mode, i.e. it is always at zero speed, both $\widetilde{\mathbf{W}}_{eng}$ and $\widetilde{\mathbf{W}}_{gnd}$ can be removed. Then linearly independent rows of $\widetilde{\mathbf{W}}_1$ are used to model the kinematic relationship of pure electric modes. Since we are only interested in kinematically 1-dof and 2-dof modes, only $\widetilde{\mathbf{W}}_1$ matrices with rank 1 and 2 are used. For a 1-dof mode, the kinematic relationship matrix is defined as:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega_2 = \mathbf{C}_{mode} \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix}. \tag{3.18}$$

Similarly, for a 2-dof mode, the same kinematic matrix becomes:

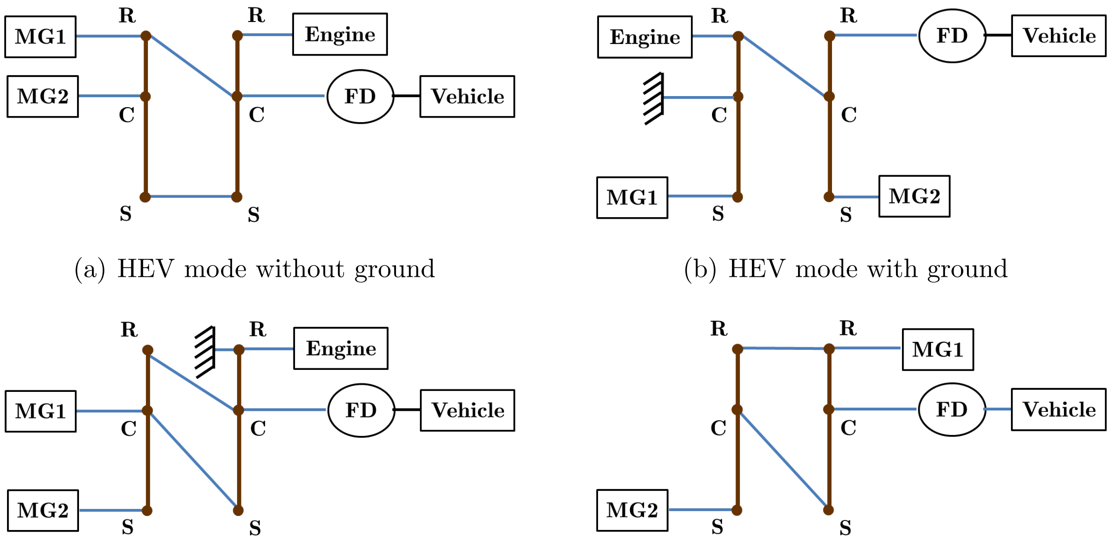
$$\omega_2 = \mathbf{C}_{mode} \begin{bmatrix} \omega_3 \\ \omega_4 \end{bmatrix}. \tag{3.19}$$

3.3 Results

The formulation presented in this chapter can be used to generate all possible driving modes for systems with any number of PGs and any number of external components. In addition to the HEV modes, by removing the engine from the external components or connecting the engine and ground to the same gear, pure EV modes can be generated.

As discussed earlier, 2-dof HEV modes and 1-dof and 2-dof EV modes are the main focus of interest. A common practice in hybrid vehicle design is to use two MGs in addition to the engine, and some designs include an optional ground. We classify the HEV and EV modes in four groups. HEV modes are grouped into two based on the existence of a ground or not and EV modes are also grouped into two based on the existence of the engine or not. Figure 3.12 shows lever representation of 4 samples from these groups for 2-PG systems.

The process described above results in 52 unique feasible graphs of driving modes with one PG. Among the 52 modes,16 of them are HEV modes and 36 are pure EV modes. These results include some available designs from the market such as the Toyota Prius mode shown in Figure 3.3(b) as well as hybrid and pure EV modes of



(a) HEV mode without ground

(b) HEV mode with ground

(c) Pure EV mode with engine grounded

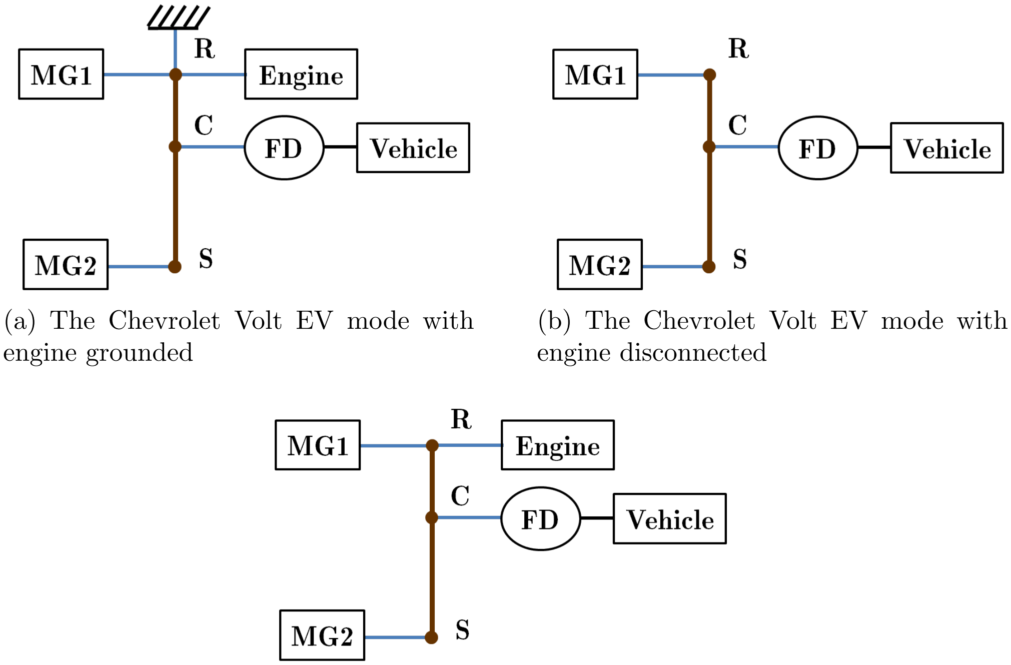
(d) Pure EV mode without engine

Figure 3.12: Four samples among all possible 2-PG driving modes

the Chevrolet Volt shown in Figure 3.13.

The number of feasible graphs for 2-PG systems is 3420 where 2124 of them are HEV and 1296 of them are pure EV modes. Among these 2-PG configurations, there some designs which look different but give the same kinematic relationship. Their graph representations are not isomorphic but they are kinematically equivalent. Such a case is called *pseudo-isomorphism* (*Hsieh and Tsai*, 1996a). When the gear ratio of the 2 PGs are both equal to 2, these 2124 HEV modes will have 1178 unique kinematic matrices for instance. We keep these kinematically equivalent designs as they can be useful for multi-mode architecture designs. More detailed discussion with examples are given Chapter VI. Also some of the configurations are only feasible when PG ratios are different. For instance, 36 of all 2-PG configurations fall into this category. When PG ratios of the two PGs are the same, these configurations given non-invertible $\mathbf{C}_{mode}$ matrices.

Among the 2-PG designs generated there are some available designs from literature such as those by *Ai and Anderson* (2005) given in Figure 3.14 and *Schmidt* (1999a), given in Figure 3.15.

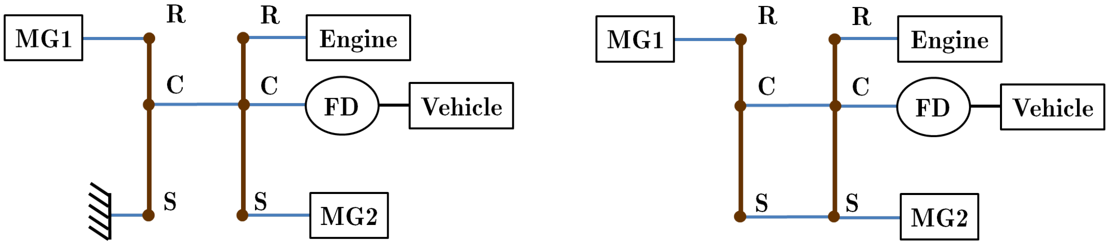


(a) The Chevrolet Volt EV mode with engine grounded

(b) The Chevrolet Volt EV mode with engine disconnected

(c) The Chevrolet Volt HEV mode

Figure 3.13: Three modes of the Chevrolet Volt generated by the process



(a)

(b)

Figure 3.14: All modes of the dual-mode architecture by *Ai and Anderson* (2005)

In addition, it possible to find 2-PG modes with the similar kinematic matrix as 1-PG designs. We refer to them as 2-PG equivalence of 1-PG designs. For example, the second mode of the architecture from *Ai and Anderson* (2005) shown in Figure 3.14(b) is the 2-PG equivalence of the 1-PG hybrid mode of the Chevrolet Volt. The mode given in Figure 3.16(b) can also be seen as the 2-PG equivalence of the hybrid mode of the Chevrolet Volt with an additional final drive, since the second PG only adds an extra gear ratio before the vehicle output shaft. Figure 3.16(a) shows 2-PG equivalence of the Toyota Prius mode.

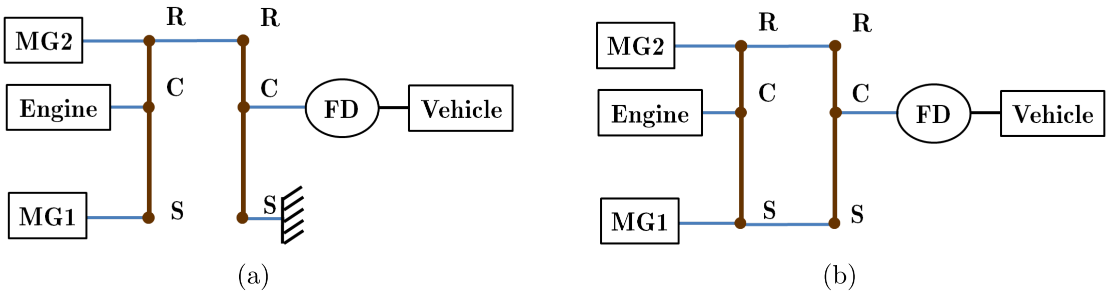
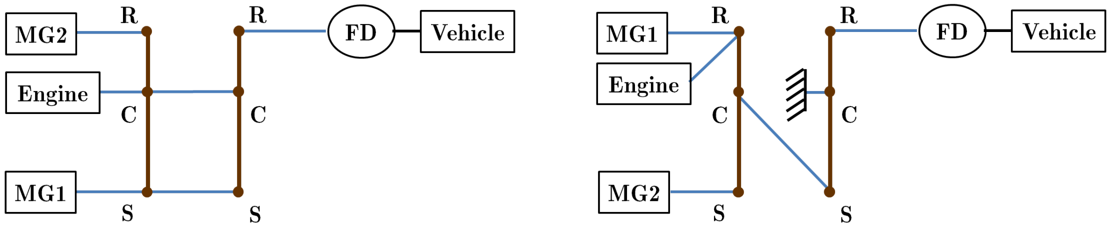


Figure 3.15: All modes of the dual-mode architecture by *Schmidt* (1999a)



(a) Toyota Prius-like 2-PG mode

(b) Chevy Volt-like 2-PG mode, the PG on the right serves as an extra final drive

Figure 3.16: 2-PG modes equivalent to the Toyota Prius and the Chevrolet Volt (with extra final drive)

Appendix A shows all 1-PG designs and Appendix B shows some selected 2-PG modes generated by the process.

3.4 Summary

In this section we reviewed some available representations used for HEV architectures. Among these representations, we introduced a modified bond graph representation based on the concepts from bond graphs in our study and introduced a systematic process to generate all possible modes for given set of external components. The main focus of this chapter was the power-split type of modes since only this type of mode allows a variety of configuration possibilities. Parallel and series type of modes can be configured in a single way and do not provide any room for configuration design. If such modes are of interest, they can be included manually in the analysis.

The generation process was general enough to be extended to any number of external components and any number of planetary gears. In our representation, we can capture both HEV and EV modes. The use of these modes in the design process is described later in this dissertation.

CHAPTER IV

Hybrid Electric Vehicle Architecture Design Optimization

Chapter III described the generation and representation phases of the design process introduced in Chapter I for the HEV architecture design problem. In order to discuss the last two phases of the design process, namely evaluation and guidance, the design problem formulation must be discussed, first. This chapter defines the general HEV architecture optimization problem formally, including component sizing design, and discusses the solution strategies for different cases based on whether component sizing is part of the design problem or not. The details of each case will be discussed later in the dissertation in Chapters V, VI, VII.

4.1 General Problem Formulation

The general HEV architecture optimization problem can be defined formally as:

General HEV Architecture Problem

$$
\begin{aligned}
\text{min} \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \\
\text{with respect to (w.r.t.)} \quad & \{\mathbf{x}_c(N_{mode}),\ \mathbf{x}_s,\ \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})\} \\
\text{subject to (s.t.)} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \leq \mathbf{0} \\
& g_{complex}(\mathbf{x}_c(N_{mode})) \leq \mathbf{0} \\
& \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p}) \text{ is attainable} \\
& lb \leq \mathbf{x}_s \leq ub \\
& N_{mode} \in \{1, 2, 3, 4, ...\} \\
& \mathbf{x}_c \text{ is feasible}
\end{aligned}
\tag{4.1}
$$

where the objective to minimize, denoted by f_{cons}, is the fuel consumption of the vehicle under a given set of drive cycles. This objective depends on the configuration described by a vector $\mathbf{x}_c$, the number of modes in the architecture denoted by N_{mode}, the size of the powertrain components including gear ratios $\mathbf{x}_s$, the supervisory control policy ψ which distributes the demanded power to engine and MGs, and the vehicle parameters $\mathbf{p}$ such as vehicle mass, wheel inertia, vehicle frontal area, aerodynamic drag, and so on.

The first set of constraints denoted by $\mathbf{g}_{perf}$ describes all the performance constraints expected to be satisfied by the powertrain architecture. These constraints may include gradeability, 0-60 Miles Per Hour (MPH) time, 30-50 MPH time, towing capability or top speed (*Freyermuth et al.*, 2008; *Nelson et al.*, 2007; *Whitefoot et al.*, 2010). The second set of constraints denoted by $g_{complex}$ limits the maximum complexity of the architecture. A formal definition of a complexity measure for HEV architectures is given in Chapter VI. The constraint on the ψ comes from the limitations of the powertrain components. For instance, engine and MGs have both speed and torque limits and batteries used in HEVs are limited to work in a predefined range of state of charge (SOC) values determined by the manufacturer. Since battery hybrid electric vehicle are not charged by an external source, another constraint on ψ may be imposed as sustaining the final battery SOC at the initial charge level.

The final sets of constraints are on the design variables. Variables $\mathbf{x}_s$ have lower and upper bounds denoted by lb and ub, respectively; N_{mode} which is the number of driving modes in the architecture can only be integer. Typically the maximum number of driving modes is limited by the complexity constraint on the architecture; $\mathbf{x}_c$ which is the vector describing the configurations of all driving modes in the architecture must correspond to feasible configurations. As described in Chapter III, if the modified bond graph representation is used to define the configuration, only simple and connected graphs, obeying the causality rules of bond graphs with the bond weights describing the kinematic relationship of the PG set are considered as feasible. In addition to these constraints, no vehicle and ground connection is allowed and the number of dof the system is constrained to two.

4.2 Solution Strategies

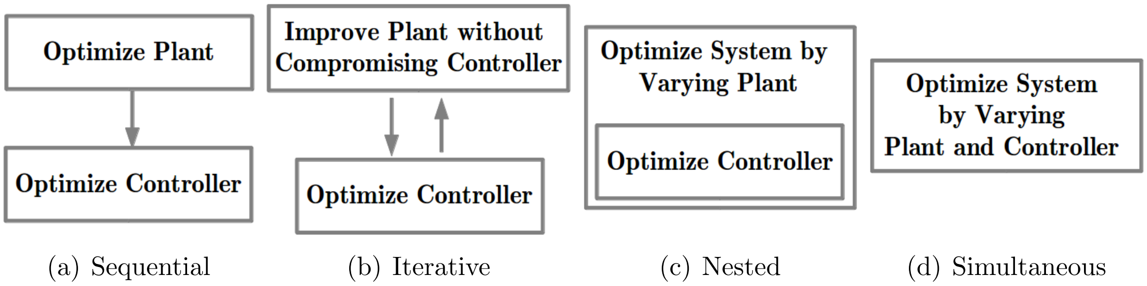
In this section, we discuss the solution strategies for the problem given in Equation (4.1). The objective function and performance constraints in the problem formulation depend both on design and control decisions. Section 4.2.1 gives a brief description of available approaches to solve combined design and control problems. Then, in Section 4.2.2, we discuss the solution strategies of the overall problem under two scenarios: (i) design of the architecture when the component sizes and gear ratios are given; (ii) simultaneous design of architecture and component sizing.

4.2.1 Combined Design and Control Problem

The problem given in Equation (4.1) is a combined design and control problem, where the objective and some constraints depend on both design and control decisions. Multiple strategies have been developed to address this type of problems. *Fathy et al.* (2001) gives a review of four strategies shown in Figure 4.1. The first method discussed in that work is the sequential approach which assumes the design and control problems are fully separable. In this approach, the design problem is solved first assuming an initial control strategy. Then, based on the optimal solution of the design problem, the controller is optimized. Since, generally design and control problems are coupled, this strategy finds non-optimal solutions. *Peters et al.* (2010) give a detailed discussion on the coupling between the pant and controller design.

The methods accounting for the coupling between design and control are commonly referred to as *co-design*. The iterative approach is one step forward from the sequential approach to consider the design and control coupling. This method solves the design and control problem multiple times in a sequential way until convergence. As discussed by *Fathy et al.* (2001), this method does not necessarily converge to the true optimum. The next co-design method is the nested approach which solves the control problem in an inner loop within the design problem. So in the outer loop, each design is evaluated with its best control strategy. This method is also referred to as bi-level solution strategy. If the combined problem is convergent, this method is guaranteed to converge to an optimal combined design and control solution. The final strategy is the simultaneous design and control strategy where the design and control variables are optimized together in a single formulation. This method also finds the true optimum.

Sequential and iterative approaches are not used in this dissertation because of their inability to find the optimum of the combined problem. While simultaneous approach can find the true optimum, it requires sophisticated methods to solve the


Optimize Plant
Optimize Controller
Improve Plant without Compromising Controller
Optimize Controller
Optimize System by Varying Plant
Optimize Controller
Optimize System by Varying Plant and Controller

(a) Sequential (b) Iterative (c) Nested (d) Simultaneous

Figure 4.1: Strategies for combined plant (design) and controller optimization

combined problem. In the literature, there are available HEV optimal control methods developed for a given design (*Lin et al.*, 2003; *Delprat et al.*, 2002; *Kim et al.*, 2011; *Paganelli et al.*, 2002). Since the nested approach can benefit from these available control methods, it is preferred in this dissertation for practical reasons. In the nested approach, the design (upper loop) problem becomes:

$$\begin{aligned}
\text{min} \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \\
\text{w.r.t.} \quad & \{\mathbf{x}_c(N_{mode}),\ \mathbf{x}_s\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \leq \mathbf{0} \\
& g_{complex}(\mathbf{x}_c(N_{mode})) \leq \mathbf{0} \\
& lb \leq \mathbf{x}_s \leq ub \\
& N_{mode} \in \{1, 2, 3, 4, ...\} \\
& \mathbf{x}_c \text{ is feasible}
\end{aligned} \tag{4.2}$$

where ψ^* is the optimal control strategy obtained by the nested control problem solution for a given design. The variables of the upper loop are $\mathbf{x}_c$ and $\mathbf{x}_s$ only. The control (inner loop) problem can be given as:

$$\begin{aligned}
\min \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \\
\text{w.r.t.} \quad & \{\psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \leq \mathbf{0} \\
& \psi(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p}) \text{ is attainable}
\end{aligned} \tag{4.3}$$

Here, since $\mathbf{x}_c$ and $\mathbf{x}_s$ are given from the design problem, only ψ is the variable. The optimal HEV control methods to solve this problem are given in Chapter V. The next section elaborates on the solution strategies for the design problem under different cases while the control problem is solved in the same way for all the cases.

4.2.2 HEV Architecture and Sizing Problem

Given the control problem nested in the design problem formulation, we provide separate solution strategies to solve the problem given in Equation (4.1) depending on whether the sizing variables $\mathbf{x}_s$ are fixed or not. For each case design of single-mode and multi-mode architectures are discussed.

4.2.2.1 HEV Architecture Design with Fixed Sizing

The sizing variables may be fixed by other constraints or good initial values for these variables might be provided beforehand. When the component sizes are given, they are included in the parameter set $\mathbf{p}$ and are no longer part of the design variable set. In the case of single-mode architecture design, assuming that the control problem is solved in a nested formulation, the problem given in Equation (4.1) is reduced to the following form:

Single-mode HEV Architecture Problem with Fixed Component Sizes

$$\begin{aligned} \min \quad & f_{cons}(\mathbf{x}_c, \psi^*(t, \mathbf{x}_c, \mathbf{p})) \\ \text{w.r.t.} \quad & \{\mathbf{x}_c\} \\ \text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c, \psi^*(t, \mathbf{x}_c, \mathbf{p})) \leq \mathbf{0} \\ & \mathbf{x}_c \text{ is feasible} \end{aligned} \tag{4.4}$$

In the single-mode architecture design, complexity is not part of the formulation, since complexity, as will be defined in Chapter VI, is the same for all configurations generated in the previous chapter. In formulation of Equation (5.1), $\mathbf{x}_c$ defining the configuration is the only design variable vector which can be represented by a modified bond graph as shown in Chapter III. Using this representation we showed that the number of feasible hybrid configurations is 16 for a 1-PG system, 2124 for a 2-PG system. So, if a computationally efficient control algorithm is used in the nested problem, we do not need to use an optimization algorithm to design $\mathbf{x}_c$ since enumeration over these finite choices is possible. Chapter V elaborates on this problem and finds solutions for some vehicles currently available in the market.

In the case of a multi-mode architecture design with given sizes, the problem formulation includes a larger $\mathbf{x}_c$, the size of which depends on N_{mode}, defining the configurations for all the modes in the architecture as design variables. More formally, the formulation becomes:

Muti-mode HEV Architecture Problem with Fixed Component Sizes

$$
\begin{aligned}
\min \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{p})) \\
\text{w.r.t.} \quad & \{\mathbf{x}_c(N_{mode})\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}), \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{p})) \leq \mathbf{0} \\
& g_{complex}(\mathbf{x}_c(N_{mode})) \leq \mathbf{0} \\
& N_{mode} \in \{1, 2, 3, 4, ...\} \\
& \mathbf{x}_c(N_{mode}) \text{ is feasible}
\end{aligned}
\tag{4.5}
$$

Note that since combining modes in the same architecture might result in high complexity, a constraint on the maximum complexity has to be part of the consideration here. In a multi-mode hybrid architecture, one of the modes has to be a hybrid mode in order to be able to charge the battery without any need for an external charger while the other modes can be either hybrid or pure electric modes. Including pure electric modes, the number of possible N_{mode} architectures is $16 \times 52^{N_{mode}-1}$ for 1-PG systems, $2124 \times 3420^{N_{mode}-1}$ for 2-PG systems and so on. As opposed to the single-mode architecture design, even though a computationally efficient control method is used, enumeration over all possible configurations is only possible for 1-PG systems. For a general multi-mode design, an optimization method is necessary to design $\mathbf{x}_c$ by searching in the design space efficiently. Assuming a Bond Graph representation is used, available derivative-free methods such as Genetic Algorithm or Pattern Search fail to find feasible configurations satisfying the feasibility constraints described in Chapter III. An alternative heuristic approach searching only in the feasible design space is proposed and details are given in Chapter VI.

4.2.2.2 Simultaneous HEV Architecture and Sizing Design

Finding good initial values for the sizing variables is not possible all the time. In that case, the problem formulation given in Section 4.2.2.1 either needs to be solved multiple times with different sizing parameters or the sizing problem needs to be solved as a nested problem problem. The former is called an iterative approach and the latter is called All-In-One (AIO). Both of the approaches are not computationally feasible efforts considering the problem size. An alternative AIO formulation can be solving the configuration and sizing problem simultaneously keeping both $\mathbf{x}_c$ and $\mathbf{x}_s$ design variables for a common objective. This approach suffers from the same problem mentioned in the previous section. Since there are many feasibility constraints on $\mathbf{x}_c$, the available derivative-free algorithms fail to find feasible designs. In order to address this combined sizing and configuration problem, a decomposition-based formulation is proposed in the present dissertation.

As descried in Chapter III, a 2×2 $\mathbf{C}_{mode}$ matrix which is a function of the configuration $\mathbf{x}_c$ and PG ratios $\boldsymbol{\rho}$ can be used to evaluate a configuration in a vehicle model. However, designing this $\mathbf{C}_{mode}$ matrix alone does not necessarily yield feasible configurations, i.e., not every $\mathbf{C}_{mode}$ can be realized by a feasible configuration. So, feasibility of the $\mathbf{C}_{mode}$ has to be considered in the optimization formulation.

Considering only PG ratios $\boldsymbol{\rho}$ and final drive ratio FR as the sizing variables, the proposed decomposition for a single-mode architecture design is depicted on Figure 4.2. A new configuration matrix combining $\mathbf{C}_{mode}$ and FR is defined as:

$$\mathbf{C}_{conf}(\mathbf{x}_c, \boldsymbol{\rho}, FR) = \begin{cases} \mathbf{C}_{mode} \cdot \begin{bmatrix} 1 & 0 \\ 0 & FR \end{bmatrix} & if \text{ hybrid mode} \\ \mathbf{C}_{mode} \cdot 1/FR & if \text{ pure electric mode,} \end{cases} \tag{4.6}$$

In the proposed decomposition, elements of the $\mathbf{C}_{conf}$ matrix is optimized at the

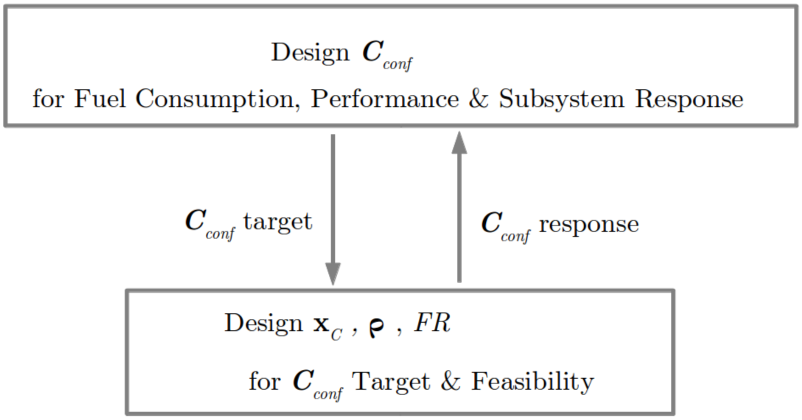

Design $\boldsymbol{C}_{conf}$
for Fuel Consumption, Performance & Subsystem Response
$\boldsymbol{C}_{conf}$ target
$\boldsymbol{C}_{conf}$ response
Design $\mathbf{x}_C$, $\boldsymbol{\rho}$, FR
for $\boldsymbol{C}_{conf}$ Target & Feasibility

Figure 4.2: Decomposition of combined single-mode architecture and gear ratio design

system-level to minimize the fuel consumption with performance constraints ignoring the feasibility of $\mathbf{C}_{conf}$. Then, the optimized $\mathbf{C}_{conf}$ is sent to the subsystem as a target to meet. The subsystem designs $\boldsymbol{\rho}$ and FR for all feasible configurations to the target as close as possible.Then, the subsystem sends the $\mathbf{C}_{conf}$ that matches the target closest to the system level. Then the system-level problem is optimized again based on that response. A penalty on the difference between target and response is added to the objective of the system-level problem. The process continues until the target from the upper system is met.

The formulation implementing this idea for the system-level problem becomes:

$$\begin{aligned}
\text{min} \quad & f_{cons}(\mathbf{C}^U_{conf}, \psi^*(t, \mathbf{C}^U_{conf}, \mathbf{p})) + \phi(\mathbf{C}^U_{conf} - \mathbf{C}^L_{conf}) \\
\text{w.r.t.} \quad & [C_{11}, C_{12}, C_{21}, C_{22}], \text{ where } \mathbf{C}^U_{conf} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{C}^U_{conf}, \psi^*(t, \mathbf{C}^U_{conf}, \mathbf{p})) \leq \mathbf{0} \\
& lb \leq \mathbf{C}^U_{conf} \leq ub \\
& |det(\mathbf{C}^U_{conf})| > 0
\end{aligned} \tag{4.7}$$

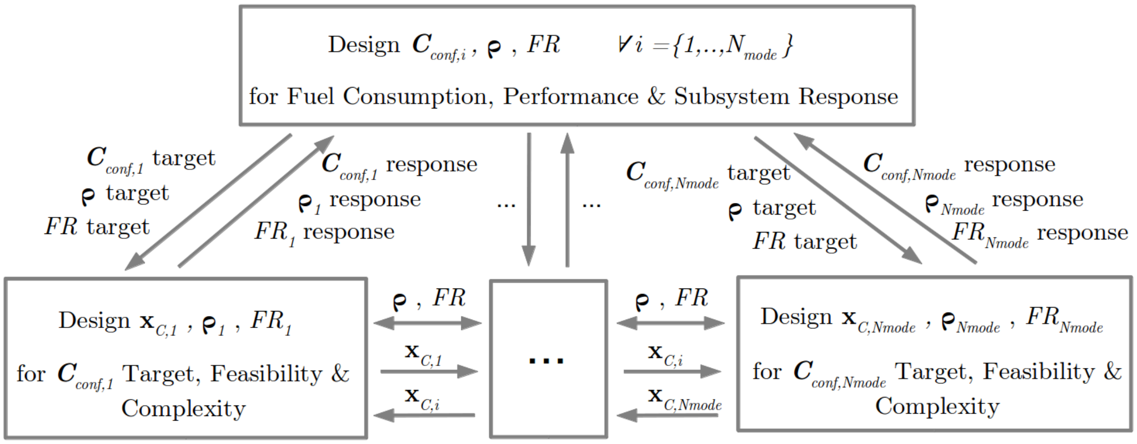

Design $C_{conf,i}$, ρ , FR $\forall i = \{1,..,N_{mode}\}$
for Fuel Consumption, Performance & Subsystem Response
$C_{conf,1}$ target
ρ target
FR target
$C_{conf,1}$ response
$ρ_1$ response
FR_1 response
...
...
$C_{conf,Nmode}$ target
ρ target
FR target
$C_{conf,Nmode}$ response
$ρ_{Nmode}$ response
FR_{Nmode} response
Design $\mathbf{x}_{C,1}$, $ρ_1$, FR_1
for $C_{conf,1}$ Target, Feasibility & Complexity
ρ , FR
$\mathbf{x}_{C,1}$
$\mathbf{x}_{C,i}$
...
ρ , FR
$\mathbf{x}_{C,i}$
$\mathbf{x}_{C,Nmode}$
Design $\mathbf{x}_{C,Nmode}$, $ρ_{Nmode}$, FR_{Nmode}
for $C_{conf,Nmode}$ Target, Feasibility & Complexity

Figure 4.3: Decomposition of combined multi-mode architecture and gear ratio design

where ϕ is the augmented Lagrangian penalty function (*Tosserams et al.*, 2006). Superscripts $(\cdot)^U$ and $(\cdot)^L$ indicate upper and lower system variables, respectively. Also the subsystem problem is formulated as:

$$\begin{aligned} \text{min} \quad & \phi(\mathbf{C}^U_{conf} - \mathbf{C}^L_{conf}) \\ \text{w.r.t.} \quad & \mathbf{x} = [\mathbf{x}_c, \mathbf{x}_s], \text{ where } \mathbf{x}_s = [\boldsymbol{\rho}, FR] \\ \text{s.t.} \quad & lb \le \mathbf{x}_s \le ub \\ & \mathbf{x}_c \text{ is feasible} \\ \text{where} \quad & \mathbf{C}^L_{conf} = f_{conf}(\mathbf{x}) \end{aligned} \tag{4.8}$$

Here f_{conf} is the function that calculates the $\mathbf{C}_{conf}$ given the configuration, PG ratios and final drive ratio. Details of the solution strategy using Analytical Target Cascading (*Kim*, 2001; *Michelena et al.*, 2003) for this decomposed problem is given in Chapter VII.

In case of a multi-mode architecture design, the main idea behind decomposition remains the same. However, the system- and subsystem-level formulations slightly differ from the single-mode design. The decomposition is depicted in Figure 4.3. In the proposed decomposition, the number of subsystems is equal to the number of modes communicating with a single system. Sizing variables are shared among subsystems since the PG ratios and final drive ratio must be the same among all the modes in a multi-mode architecture. The variables of the system-level problem are the elements of $\mathbf{C}_{conf}$ matrix of all the modes in the architecture and the shared variables of the subsystems. Due to these shared variables among subsystems, an additional penalty on the difference between the shared variables from the subsystems is part of the system objective in addition to the fuel consumption and the difference between $\mathbf{C}_{conf}$ targets and responses from the subsystems. The subsystems try to meet the $\mathbf{C}_{conf}$ and gear ratio targets coming from the system. There is an additional complexity constraint in the subsystems, which requires sharing the configurations $\mathbf{x}_c$ in order to calculate the complexity. The mathematical formulation for the system-level problem can be given as:

$$\begin{aligned}
\min \quad & f_{cons}(\mathbf{C}^U_{conf,i}, \psi^*(t, \mathbf{C}^U_{conf,i}, \mathbf{p})) + \sum_{i=1}^{N_{mode}} \phi(\mathbf{C}^U_{conf,i} - \mathbf{C}^L_{conf,i}) + \phi(\mathbf{x}^U_s - \mathbf{x}^L_{s,i}) \\
\text{w.r.t.} \quad & [C_{11,i}, C_{12,i}, C_{21,i}, C_{22,i}, \mathbf{x}^U_s], \text{ where } \mathbf{C}^U_{conf,i} = \begin{bmatrix} C_{11,i} & C_{12,i} \\ C_{21,i} & C_{22,i} \end{bmatrix} \ \forall i \in \{1, ...N_{mode}\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{C}^U_{conf,i}, \psi^*(t, \mathbf{C}^U_{conf,i}, \mathbf{p})) \leq \mathbf{0} \\
& lb_i \leq \mathbf{C}^U_{conf,i} \leq ub_i \\
& |det(\mathbf{C}^U_{conf,i})| > 0_i
\end{aligned} \tag{4.9}$$

where the subscript $(\cdot)_{\cdot,i}$ denotes each mode.

Also the formulation for a subsystem i becomes:

$$\begin{aligned}
\text{min} \quad & \phi(\mathbf{C}^U_{conf,i} - \mathbf{C}^L_{conf,i}) + \phi(\mathbf{x}^U_s - \mathbf{x}^L_{s,i}) \\
\text{w.r.t.} \quad & \mathbf{x}_i = [\mathbf{x}_{c,i}, \mathbf{x}^L_{s,i}], \text{ where } \mathbf{x}^L_{s,i} = [\boldsymbol{\rho}_i, FR_i] \\
\text{s.t.} \quad & g_{comp}(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}) \leq 0 \quad \forall j \in \{1, ...N_{mode} | j \neq i\} \\
& lb_i \leq \mathbf{x}^L_{s,i} \leq ub_i \\
& \mathbf{x}_{c,i} \text{ is feasible} \\
\text{where} \quad & \mathbf{C}^L_{conf,i} = f_{conf}(\mathbf{x}_i)
\end{aligned} \tag{4.10}$$

where $\mathbf{x}_s$ are the shared variables.

4.3 Summary

We gave an overview of the different intances of the general HEV architecture problem. We described general approaches to solve the combined design and control problems and proposed to use the nested formulation in order to be able to benefit from the existing HEV control strategies. We provided separate formulations for the HEV architecture design problems with single and multiple modes when the component sizes are given. Chapter V will describe the solution strategy for the single-mode architecture design problem with fixed component sizes and Chapter VI will discuss the multi-mode architecture design problem with the same assumptions for the component sizes.

Also we proposed a decomposition-based formulation to solve the combined configuration and component sizing design decisions simultaneously. Chapter VII will elaborate more on this formulation.

CHAPTER V

Single-mode Hybrid Electric Vehicle Architecture Design with Fixed Sizing

Chapter III demonstrated the representation of HEV architectures based on bond graphs and described a generation framework to find all feasible configurations (driving modes) given some powertrain components. Chapter IV formulated the general single and multi-mode HEV architecture design problems for fixed and variable component sizing. In this chapter we will discuss the solution of the single-mode HEV architecture design problem with given component sizes.

First we will describe the evaluation of the design alternatives in a general vehicle model. As discussed in Chapter IV, the evaluation of an architecture requires an optimal control strategy. We will use a nested formulation to solve the optimal control problem, i.e., each design candidate will be evaluated with the best control strategy. Then based on the evaluation of each design, we will select the best architecture design and perform a parametric study to show the effect of some key parameters on the design selection.

5.1 Problem Formulation

A formal mathematical formulation of the general architecture design problem was given in Chapter IV. The general formulation reduces down to the following from for the single-mode architecture design with given powertrain component sizes:

$$\begin{aligned} \text{min} \quad & f_{cons}(\mathbf{x}_c, \psi^*(t, \mathbf{x}_c, \mathbf{p})) \\ \text{w.r.t.} \quad & \{\mathbf{x}_c\} \\ \text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c, \psi^*(t, \mathbf{x}_c, \mathbf{p})) \leq \mathbf{0} \\ & \mathbf{x}_c \text{ is feasible} \end{aligned} \tag{5.1}$$

where f_{cons} is the fuel consumption of the vehicle under some given drive cycles, $\mathbf{g}_{perf}$ is the set of vehicle performance constraints which can be gradeability, top speed or acceleration requirements. $\mathbf{x}_c$ denotes the configuration, i.e., driving mode that should be selected from the feasible modes generated in Chapter III. $\mathbf{p}$ is the vector of vehicle parameters. Note that, since powertrain component sizes, $\mathbf{x}_s$ are assumed to be given in this chapter, they are also included in this vector of parameters. ψ is the supervisory control strategy that distributes the power demand from the driver to the powertrain components. Since we use a nested formulation to solve the control problem, the available optimal control strategies developed for a given design can be used. Section 5.3 describes these strategies in detail. ϕ^* denotes the optimal control strategy obtained in such a nested formulation.

Selection of feasible configuration from the generated designs for the single-mode architecture can be done using enumeration that is computationally feasible when the component sizes are given. We will present three case studies in Section 5.4 solving this design problem by evaluating each feasible design candidate separately.

5.2 Vehicle Model

We will describe a general vehicle model to evaluate an HEV configuration in terms of fuel consumption and vehicle performance under given vehicle parameters. A drive cycle, which is defined by a speed and road grade profile over a time horizon, is necessary to calculate the fuel consumption of a configuration. In civilian vehicle design, the road profile is usually assumed to be flat while in military applications both speed and road grade profile are included in the drive cycle. In HEV simulations, the fuel consumption calculation has to accommodate both engine consumption and battery consumption.

The model described here assumes that a drive cycle is given a priori and the vehicle follows the cycle exactly. Under that assumption, the rotational speed ω_{req} required from the transmission for the vehicle to follow the cycle is calculated as follows:

$$\omega_{req} = \frac{V_{cycle}}{R_{wheel}} FR, \tag{5.2}$$

where R_{wheel} is the wheel radius, FR is the final drive ratio and V_{cycle} is the speed of the vehicle following a drive cycle. In addition, the required torque T_{req} is given by

$$T_{req} = (F_{acc} + F_{aero} + F_{roll}) \frac{R_{wheel}}{\eta_{trans}\, FR}, \tag{5.3}$$

where η_{trans} is the transmission efficiency. In this equation, the rolling friction F_{roll} is calculated as

$$F_{roll} = M_{veh}\, g(\mu_{stat} + \mu_{dyn} V_{cycle}). \tag{5.4}$$

Here, μ_{stat} and μ_{dyn} are static and dynamic rolling friction coefficients, respectively; M_{veh} is the vehicle mass and g is the gravitational acceleration. The aerodynamic drag force F_{aero} is given by:

$$F_{aero} = 0.5\ \rho_{air}\ C_d\ A_{front}\ V_{cycle}^2, \tag{5.5}$$

where ρ_{air} is the air density, C_d is aerodynamic drag coefficient and A_{front} is the frontal area of the vehicle. Finally, the force required to accelerate, F_{acc} is simply,

$$F_{acc} = (M_{veh} + 4\ J_{wheel}/R_{wheel}^2)a_{cycle}. \tag{5.6}$$

This equation also accounts for the wheel inertia J_{wheel}. Here, a_{cycle} is the acceleration of the vehicle following a given drive cycle.

Given ω_{req} and T_{req}, a supervisory control strategy can determine the engine operating points, i.e., ω_{eng} and T_{eng} minimizing the fuel consumption. The details of the control strategies are discussed in Section 5.3. In the model here, a quasi-static approach described in Section 3.1.4.2 is utilized to calculate the speed and torque relationships among vehicle output, MGs and the engine. In summary, the relationships for a 2-dof hybrid mode with two MGs and an engine are as follows.

$$\begin{bmatrix} \omega_{MG1} \\ \omega_{MG2} \end{bmatrix} = \mathbf{C}_{mode} \begin{bmatrix} \omega_{eng} \\ \omega_{req} \end{bmatrix} \tag{5.7}$$

$$\begin{bmatrix} T_{MG1} \\ T_{MG2} \end{bmatrix} = -\mathbf{C}_{mode}^{-T} \begin{bmatrix} T_{eng} \\ -T_{req} \end{bmatrix} \tag{5.8}$$

In the case of pure electric modes, $\mathbf{C}_{mode,ev}$ matrix relates the MG speeds to vehicle output speed as follows:

$$\mathbf{C}_{mode,ev}\begin{bmatrix}\omega_{req}\\\omega_{MG1}\\\omega_{MG2}\end{bmatrix}=\mathbf{0} \tag{5.9}$$

The torque relationship for pure electric modes depends on the number of dof. Let $[\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3] = \mathbf{C}_{mode,ev}$. Then, for a 2-dof pure electric mode quasi-static torque relationship becomes:

$$-\mathbf{C}_1^T\begin{bmatrix}\mathbf{C}_2 & \mathbf{C}_3\end{bmatrix}^{-T}\begin{bmatrix}T_{MG1}\\T_{MG2}\end{bmatrix}=T_{req} \tag{5.10}$$

For a 1-dof pure electric mode the same torque relationship becomes:

$$\begin{bmatrix}T_{MG1}\\T_{MG2}\end{bmatrix}=-\begin{bmatrix}\mathbf{C}_2 & \mathbf{C}_3\end{bmatrix}\mathbf{C}_1^{-1}\,T_{req} \tag{5.11}$$

Note that a detailed mode switching model is not implemented in this vehicle model. Modes are switched by switching the speed and torque relatioships with the corresponding $\mathbf{C}_{mode}$ matrices. In order to prevent frequent switching, a penalty is imposed by the supervisory control strategy.

With ω_{eng} and T_{eng} given by the supervisory control strategy, the engine fuel consumption rate $\dot{m}_f(\omega_{eng}, T_{eng})$ can be looked up from an engine fuel consumption map. In order to calculate the battery consumption, the current demand from the battery needs to be calculated first. The model here, assumes a fixed battery DC voltage, V_{batt} and efficiency η_{batt}, for simplicity. However, this assumption can be relaxed and battery can also be modeled dynamically where the output voltage becomes a function of battery state of charge (SOC) and current demand. In the model here, the current demand by the i^{th} MG, I_{MGi}, is given by:

$$I_{MGi} = \frac{T_{MGi}\,\omega_{MGi}}{V_{batt}}\,\eta_{MGi}^{Ki} \qquad \forall i = 1, 2, \tag{5.12}$$

where $\eta_{MGi} = f(\omega_{MGi}, T_{MGi})$ denotes the efficiency of each MG which is looked up from an MG map and Ki is 1 when the corresponding MG is working as a generator and -1 when as a motor. Using the current demands from the MGs, the battery current I_{batt}, and battery power P_{batt}, are calculated as:

$$I_{batt} = \frac{I_{MG1} + I_{MG2}}{\eta_{batt}}, \tag{5.13}$$

$$P_{batt} = I_{batt}\,V_{batt}. \tag{5.14}$$

Finally, the SOC of a battery with capacity C_{batt} can be calculated by the following differential equation:

$$S\dot{O}C = -I_{batt}/C_{batt}. \tag{5.15}$$

The general vehicle model described here can be implemented in a simulation over discrete time steps until the end of the time horizon.

In order to calculate the fuel economy a control algorithm is necessary. As described in Chapter IV the control problem is solved in a nested formulation in this dissertation. The nested formulation provides the advantage of using the available control strategies developed for given HEV designs. The next section describes some of these available control strategies in detail.

5.3 Supervisory Control

The general control problem of finding the optimal engine operating points over a given drive cycle is formulated as follows:

$$
\begin{aligned}
min \quad & J = \int_{t_0}^{t_f} \dot{m}_f \cdot dt \\
subject\ to \quad & \omega_{min} \leq \omega_{eng} \leq \omega_{max} \\
& T_{min} \leq T_{eng} \leq T_{max} \\
& SOC_{min} \leq SOC(t) \leq SOC_{max}
\end{aligned} \tag{5.16}
$$

The objective is to minimize the total fuel consumption J measured from the beginning of the drive cycle t_0 to the end t_f. The constraints are the bounds on the engine speed denoted by ω_{min} and ω_{max}, on the engine torque denoted by T_{min} and T_{max}, and on the battery SOC denoted by SOC_{min} and SOC_{max}. In battery hybrid electric vehicle simulations where the vehicle has no external source to charge the battery, the fuel consumption is measured by the fuel consumption of the vehicle with initial and final battery SOC being the same. This is also called a charge-sustaining constraint:

$$SOC(t_f) = SOC(t_0). \tag{5.17}$$

The computational effort to solve this problem can be reduced by performing some of the calculations offline, namely, before starting the solution process. For an HEV mode, if the drive cycle is known a priori, at any time step all engine operating points and the corresponding MG operating points satisfying the drive cycle demand can be calculated offline. When these operating points are mapped to the 2D space of fuel consumption rate and battery power, the points minimizing the both the fuel consumption rate and battery power form a Pareto frontier as shown on Figure 5.1. Only these Pareto points along with the corresponding engine and motor operating points need to be saved for further use. The online simulation will use these Pareto curves as “lookup tables”. When solving the control problem given in Equation (5.16) and Equation (5.17), the mapping between $\dot{m}_f$ and P_{batt} on the Pareto frontier is used

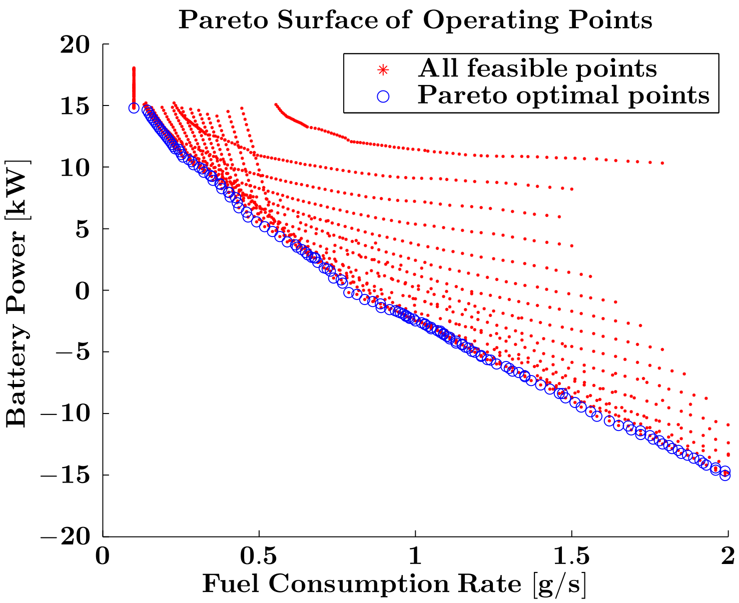


Figure 5.1: All engine and MG operating points satisfying the demand from the drive cycle form a Pareto surface on the space of $\dot{m}_f$ and P_{batt}

to reduce the number of independent variables to one for searching the optimal MG and engine operating points on the Pareto frontier.

In a pure-EV mode, since there is no engine, the offline calculation is performed for all possible MG operating points satisfying the drive cycle demand. In that case, since $\dot{m}_f = 0$, the Pareto frontier has only one point.

Three methods which can be used to solve this control problem given in Equation (5.16) and Equation (5.17) are (Dynamic Programming, Pontryagin's Minimum Principle and Equivalent Consumption Minimization Strategy). These methods are described in the subsequent sections.

5.3.1 Dynamic Programming

Dynamic Programming (DP) was first introduced by *Bellman* (1954). The formulation addresses a multi-stage decision problems described by a set of *state* and *decision* variables. Bellman's principle of optimality is an optimal decision policy at every stage based on the current state information only. The decision process works in a recursive way. The process of backward induction works as follows: Let s_n and d_n denote the state and decision variables at stage n and $c_n(s_n, d_n)$ be the resultant cost of that decision. Assume that the goal is to minimize the cost of going from state s_0 to s_N. If $\nu_{n+1}(s_{n+1})$ is the minimum cost of going from state s_{n+1} to s_N, then the decision at the state s_n is made as follows:

$$\nu_n(s_n) = min\{c_n(s_n, d_n) + \nu_{n+1}(s_{n+1})\} \tag{5.18}$$

The value of $\nu_0(s_0)$ obtained by this recursive process gives the minimum cost.

Dijkstra (1959) applied dynamic programming to finding the shortest path tree between two nodes in the most computationally efficient way. *Dijkstra's algorithm* has a computational cost on the order of $O(|E| + |V|log|V|)$ where $|E|$ denotes the number of edges (decision variables) and $|V|$ the number of vertices (state variables) in the graph.

DP has been applied to the HEV supervisory control problems before (*Lin et al.*, 2003; *Liu and Peng*, 2006). An analogy between the shortest path problem and the HEV supervisory control problem with a given drive cycle can be made. Let SOC describe the states and t describe the stages with a certain discretization. Considering the linear relationship between $S\dot{O}C$ and P_{batt} described in Section 5.2, ΔSOC which is the discretized version of $S\dot{O}C$ can be used as the only control variable in this formulation. If possible SOC values are represented by vertices on a graph and ΔSOC values by edges, the fuel consumption corresponding to each decision becomes the cost on the edges. These costs can be calculated using the mapping described by the Pareto curves calculated offline. Figure 5.2 depicts this idea for a simplified problem with three possible SOC values at every stage with three possible ΔSOC decisions. In the figure, the problem starts with the state s_0^0 denoting the SOC value of s^0 at the time t_0. Due to the charge sustaining constraint the SOC at the time t_N has to be s^0 which is denoted by s_N^0. The engine fuel consumption required to

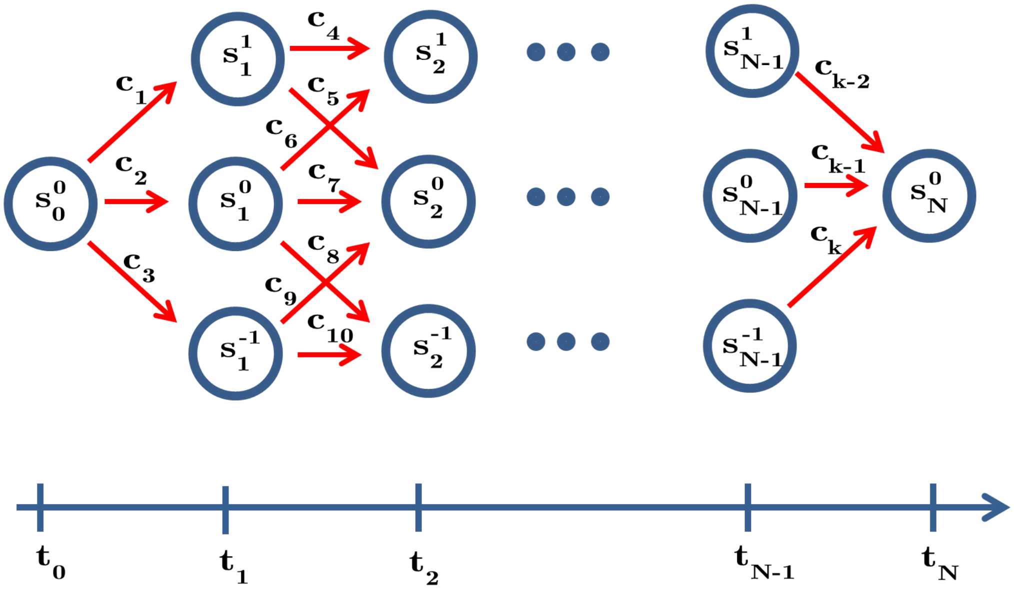


Figure 5.2: Representation of a fictitious control problem with an analogy to the shortest path problem starting with state s_0^0 and ending with s_N^0

move from one state to another is indicated by the cost variable c on the edges. With that representation, finding the shortest path problem between the nodes s_0^0 and s_N^0 is equivalent to the finding the minimum consumption starting and ending with the SOC level of s^0 between the times t_0 and t_N.

For a multi-mode HEV control problem, there is an additional state variable for the current mode and a decision variable for mode shifting. In that case the total number of states and decision variables is multiplied by the number of modes in the architecture. In addition to the cost of fuel, there is also an additional cost of shifting the mode in the multi-mode HEV control problem. A more detailed discussion on handling the mode shifting is given in Section 5.3.3.

Solution of the HEV control problem with DP starts with creating the above-mentioned graph first. Then, the recursive decision process of finding the minimum fuel consumption with a charge-sustaining constraint can be solved using Dijkstra's algorithm. As discussed by *Kim et al.* (2011) DP has been shown to find the global optimal control policy with a given discretization at the expense of a significant computational burden. The computational cost of the solution increases exponentially with the dimension of the problem. This issue makes DP a practically infeasible effort in a design process involving evaluations of a large number architecture alternatives as in the present dissertation.

5.3.2 Pontryagin's Minimum Principle

Pontryagin's Minimum Principle (PMP) was first introduced in 1956 by L. S. Pontryagin to provide necessary conditions for optimal control problems (*Pontryagin*, 1987). It has also been used for HEV control problems as an alternative to the computationally expensive dynamic programming method (*Delprat et al.*, 2002, 2004; *Kim et al.*, 2011). Considering the problem formulation given by Equation (5.16) where $\dot{m}_f$ is made a function of P_{batt} with offline calculations, the Hamiltonian function is defined as:

$$H(SOC, P_{batt}, t) = \dot{m}_f(P_{batt}, t) + \lambda(t) \cdot f(SOC(t), P_{batt}(t)) \tag{5.19}$$

where SOC is the state and λ is the co-state. Necessary conditions for optimal P^*_{batt} is such that Hamiltonian is minimized for all t as well as the following two equalities:

$$\begin{aligned} \dot{SOC}^*(t) &= \frac{\partial H(SOC^*, P^*_{batt}, t)}{\partial \lambda} = f(SOC^*(t), P_{batt}(t)) \\ \dot{\lambda}^*(t) &= -\frac{\partial H(SOC^*, P^*_{batt}, t)}{\partial SOC} = -\lambda^*(t)\frac{\partial f(SOC^*(t), P_{batt}(t))}{\partial SOC} \end{aligned} \tag{5.20}$$

Note that the optimal co-state is not necessarily constant. However, if $\dot{SOC}$ does not depend on SOC, then $\dot{\lambda}^*(t) = 0$ meaning that the optimal co-state remains constant. *Kim et al.* (2011) proved that under this assumption a globally optimal control policy can be obtained with PMP.

5.3.3 Equivalent Consumption Minimization Strategy

Equivalent Consumption Minimization Strategy (ECMS) was initially proposed as a heuristic method to minimize the sum of engine fuel consumption and fuel consumption equivalence of battery power (*Paganelli et al.*, 2002). If λ is a conversion factor from the battery power to fuel rate, the equivalent fuel consumption denoted by EFC is defined as:

$$EFC = \dot{m}_f + \lambda \, P_{batt} \tag{5.21}$$

Given a conversion factor, this strategy minimizes EFC subject to the same constraints given by (5.16) and (5.17) at every time step throughout the simulation. As a result of the offline calculations, similar to DP, ΔSOC that is a discretized version of $S\dot{O}C$ can be used as the only control variable to minimize the EFC. Available implementations of ECMS vary slightly in different applications. For instance, some formulations include an extra term in the objective to penalize the SOC going beyond the limits given in Equation (5.17). However, if the SOC window is not too narrow, the chances of hitting the SOC bounds are quite low which makes the extra term inactive throughout the simulation. Variations also include using multiple conversion factors for charging and depleting the battery or using a conversion factor as a function of battery SOC (*Serrao et al.*, 2009; *Liu and Peng*, 2008). In this section, we follow the implementation given by *Ahn et al.* (2008).

In a multi-mode architecture, each mode yields a different EFC value at a given point on the drive cycle. In that case, the mode with minimum EFC is preferred. However, this strategy typically results in frequent mode shifting that is not desired for driveability. A common approach to solve this problem involves assigning a penalty to shifting a mode. *Lin et al.* (2003) offer an example to such an approach. With the penalty term added, the modified objective becomes:

$$\widetilde{EFC} = \dot{m}_f + \lambda \, P_{batt} + \gamma \, \delta_{mode} \tag{5.22}$$

where γ is the penalty coefficient and $\delta_{mode} = 1$ when mode is shifted or $\delta_{mode} = 0$ otherwise. The value of γ impacts the mode shifting frequency and hence the fuel consumption. Deviation from optimality increases with γ. For the implementation used in the present dissertation, a value of γ is tuned experimentally and the same value is used for all architectures. A parametric study on the value of this coefficient is desirable future work.

In a multi-mode architecture problem, two control variables become ΔSOC and δ_{mode} while the states are SOC, driving mode and time. During the simulation, given the states, by using pre-calculated Pareto curves the two control variables minimizing $\widetilde{EFC}$ can be calculated. The value of λ in Equation (5.22) affects the fuel consumption and SOC trajectory significantly. For every architecture and drive cycle, the value of λ satisfying the charge-sustaining constraint is given by Equation (5.17). The relationship between λ and final SOC is nonlinear. Therefore, a solver for λ equalizing initial and final SOCs needs to be implemented within the controller. *The secant method* can be used to find a solution in a few iterations. Figure 5.3 shows an iteration of the secant method. The implementation starts with two initial guesses for λ, say, λ_1 and λ_2 and the corresponding final SOC values are calculated by solving the control problem using ECMS. Then, the final SOC is approximated as a linear function and solved for the λ_{new} corresponding to the target SOC. λ_{new} and the λ with the final SOC closer to the target from the previous iteration are passed to the next iteration. The same process continues until an SOC within an ϵ window of the target SOC is obtained.

As discussed by *Kim et al.* (2011) and *Serrao et al.* (2009), although ECMS is not an optimal control strategy for a generic model, it can provide results close to the PMP under the following conditions:

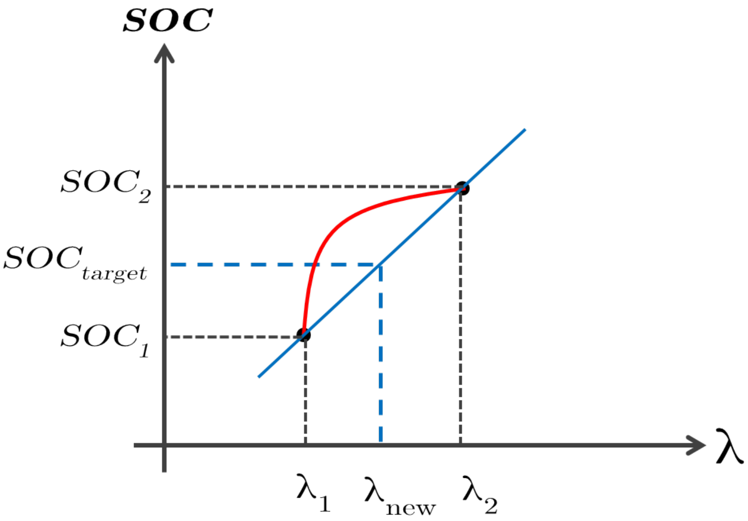


Figure 5.3: One iteration of the secant method to find the conversion factor corresponding to the target SOC

(i) The window of SOC_{min} and SOC_{max} is small such that the rate of change in the co-state is negligible.

(ii) The relationship between $S\dot{O}C$ and P_{batt} is close to linear.

Both conditions hold exactly for the model presented in Section 5.2 making ECMS and PMP solution equivalent. However, even if the modeling assumptions change in order to increase the model fidelity, this control algorithm can be favored when the computational burden of the solution becomes significant. In an optimal system design framework, a true optimality claim cannot be made with a non-optimal control decision. However, if the resultant control policy approximates the optimal control well enough, the design process can still identify promising design candidates, even if not necessarily the optimal one. For practical purposes, especially in combinatorial design problems such as studied here, identifying promising designs quickly rather than finding the optimum in a computationally expensive way can be preferred.

5.4 Design Selection

In Chapter III we described the generation of all power-split type 1-PG, 2-PG and 3-PG mode candidates in a systematic way using the bond graph representation. In this section we will discuss the mode selection for a single-mode architecture design given the mode alternatives and the design evaluation method described in the previous section in order to solve the problem formulated in Equation (5.1). We will discuss three case studies. In the first case study, we will design a 1-PG and a 2-PG single mode architecture for a Toyota Prius type of vehicle. In the second case study, we will perform the same for a Chevrolet Volt type of vehicle. In the third case study, we discuss the single-mode architecture design for a heavy-duty vehicle. For all cases, we assume that all powertrain component specifications, final drive and PG ratios are given beforehand. We calculate the fuel consumption by averaging the fuel consumption on city (UUDS) and highway (HWFET) cycles.

If the goal is to design a single-mode architecture (with given gear ratios), since the number of design alternatives is limited to a number that allows enumeration, we can evaluate each mode separately and select the best design among them. An advantage of enumeration based design search is the parallel computing capability. Since each design evaluation is independent from each other, as opposed to the algorithmic search methods, we can evaluate different design in parallel to reduce the computation time. For a 1-PG design, it is not necessary since there is only 16 design alternatives but since this number is larger for 2-PG and 3-PG architecture designs, we can benefit from parallelization significantly.

When we move from single-mode design to multi-mode design or include gear ratios in the design process, enumeration over all possible candidates is not possible any more. We need a smart way to search in the space in such cases. We will discuss multi-mode architecture design with given gear sizes in Chapter VI and the simultaneous configuration and gear size design in Chapter VII.

5.4.1 Illustration Case Study 1

The vehicle specifications used for this case study is given in Table 5.1. These specifications are similar to the Toyota Prius. Let's assume that PG ratio and final drive ratio to be used in the architecture is also the same as the Toyota Prius. Therefore, let $\rho = 2.6$ and $FR = 3.95$.

Table 5.1: Vehicle specifications used for Case Study 1

Specification	Value
Vehicle Body Mass	1400 [kg]
Tire Radius	0.3 [m]
Aerodynamic Drag Coefficient	0.29
Frontal Area	2 [m^2]
Battery Voltage	350 [V]
Battery Efficiency	92 [%]
Battery Capacity	6.5 [Ah]
Rated MG1 Power	42 [kW]
Rated MG2 Power	60 [kW]
Max MG1 / MG2 Speed	12000 [rpm]
Max MG1 / MG2 Torque	200 [Nm]
Rated Engine Power	43 [kW]
Max Engine Torque	102 [Nm]
Engine Displacement Size	1.5 [L]

In this case study we will two performance constraints that are 0-60 mph acceleration time and top speed of the vehicle requirements. 0-60 mph acceleration time of a vehicle is related to the maximum output torque capability of the architecture and top speed is related to the maximum speed the architecture can provide at the vehicle output shaft. We set these two constraints to 12 seconds and 105 mph, respectively. These values are set considering the true specifications of the Toyota Prius vehicle.

By evaluating each 16 1-PG configurations separately with $\rho = 2.6$ and $FR = 3.95$ we obtain the results shown on Figure 5.4. We sort the designs based on the feasibility and fuel consumptions. Feasible designs are indicated by black, the designs violating the 0-60 mph time constraint are by blue, the designs violating top speed constraint

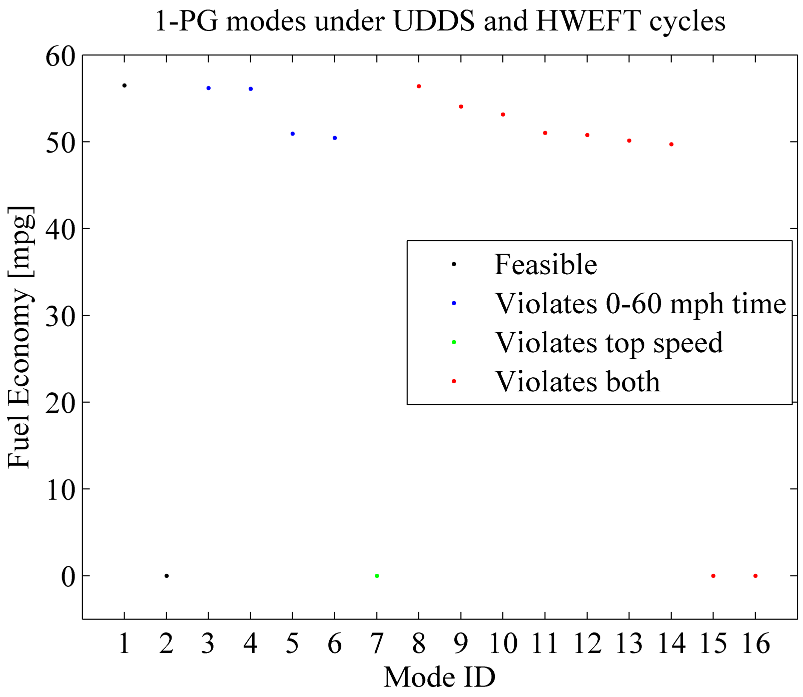


Figure 5.4: Simulation results for all 1-PG modes with $\rho = 2.6$ and $FR = 3.95$ using the vehicle specifications given in Table 5.1

are by green and the designs violating both are by red. As seen on the figure, only two modes are feasible and among them only one has a reasonable fuel economy. This design is shown on Figure 5.5. A 0 mpg fuel economy means that the design cannot sustain the battery charge at the initial SOC level while following the drive cycle. Even without the constraints, this design achieves the best average fuel consumption on city and highway cycles with 56.49 mpg. This is the architecture implemented in the Toyota Prius (*Sasaki*, 1998).

The detailed results of for the architecture given in Figure 5.5 is shown in Table 5.2.

A more interesting study can be designing a single-mode 2-PG architecture for the same vehicle. Assuming both PG ratios are the same, i.e., $\boldsymbol{\rho} = \begin{bmatrix} 2.6 \\ 2.6 \end{bmatrix}$ 2124 possible 2-PG hybrid configurations give 1214 unique $\mathbf{C}_{mode}$ matrices to select from. As discussed Chapter III, some distinct configurations might give the same kinematic

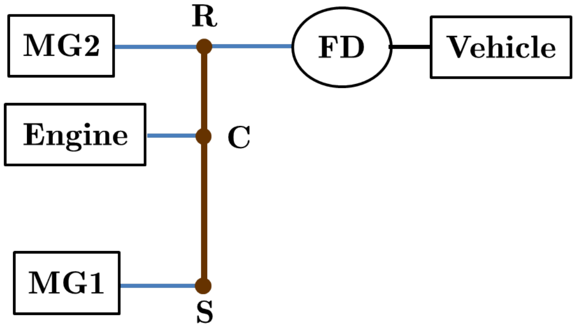


Figure 5.5: Optimal configuration obtained for the vehicle specifications given in Table 5.1 by enumerating all 1-PG designs

Table 5.2: Simulation results for the best 1-PG architecture designed for Case Study 1

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2.6$ $FR = 3.95$	$\begin{bmatrix} 3.6 & -2.6 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3.6 & -10.27 \\ 0 & 3.95 \end{bmatrix}$	56.49 mpg	109 mph	11.66 sec

matrix for given PG ratios. By evaluating each unique $\mathbf{C}_{mode}$ with $FR = 3.95$ we obtain the results shown on Figure 5.6. We performed a similar sorting in this case, too.

The top 3 designs are shown on Figure 5.7 and detailed simulation results are given on Table 5.3. As seen on Table 5.3, the $\mathbf{C}_{mode}$ matrices of the top three 2-PG designs are close to each other. $\mathbf{C}_{mode}$ matrix of the third best design is the same as the best 1-PG design. We call such designs as “2-PG equivalents” of 1-PG designs. In other words, the design space of 2-PG designs include both $\mathbf{C}_{mode}$ matrices which can be realized by 1-PG configurations and “2-PG exclusive” $\mathbf{C}_{mode}$ matrices. $\mathbf{C}_{mode}$ matrices of top 2 configurations cannot be realized by any 1-PG configuration, for instance.

The improvement from the best 1-PG design to best 2-PG design in this particular case is 2.37% in fuel economy. Increasing number of PGs in the architecture provides the opportunity to improve the fuel economy but it also increases the cost and volume

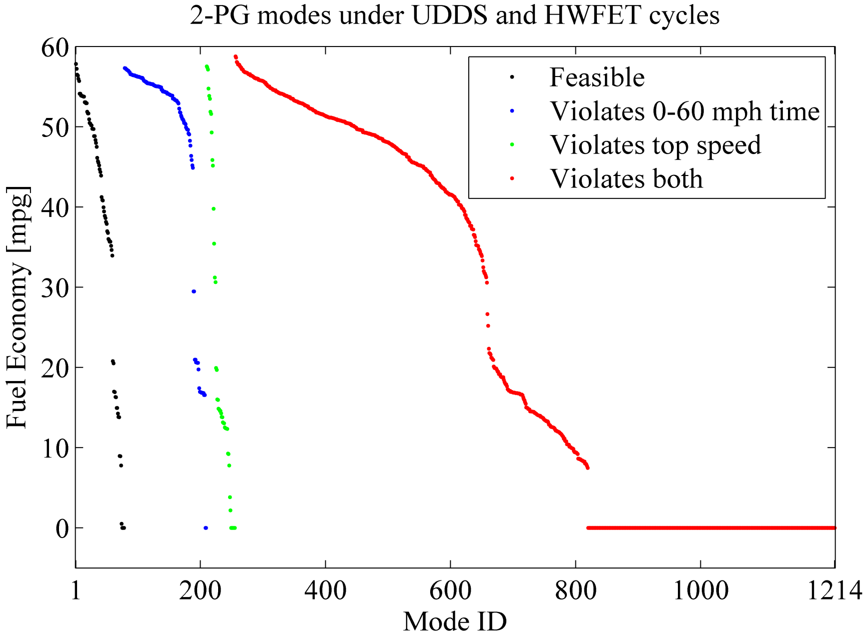


Figure 5.6: Simulation results for all 2-PG modes with $\boldsymbol{\rho} = [2.6; 2.6]$ and $FR = 3.95$ using the vehicle specifications given in Table 5.1

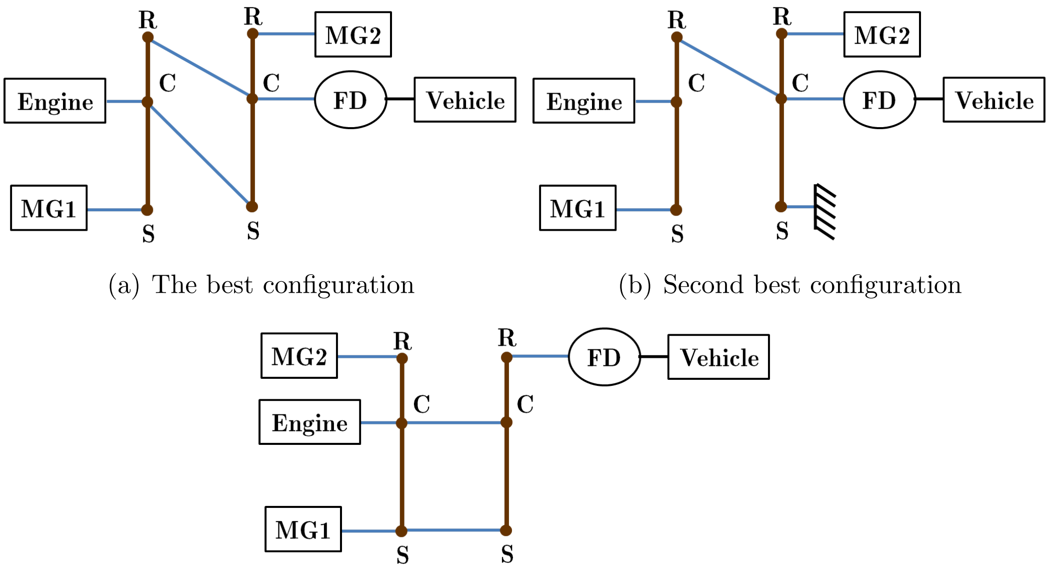
Table 5.3: Simulation results for the top three 2-PG architectures designed for Case Study 1

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho_1 = 2.6$ $\rho_2 = 2.6$ $FR = 3.95$	$\begin{bmatrix} 3.6 & -2.6 \\ -0.38 & 1.38 \end{bmatrix}$	$\begin{bmatrix} 3.6 & -10.27 \\ -0.38 & 5.47 \end{bmatrix}$	57.83 mpg	107 mph	11.81 sec
$\rho_1 = 2.6$ $\rho_2 = 2.6$ $FR = 3.95$	$\begin{bmatrix} 3.6 & -2.6 \\ 0 & 1.38 \end{bmatrix}$	$\begin{bmatrix} 3.6 & -10.27 \\ 0 & 5.47 \end{bmatrix}$	57.22 mpg	105 mph	9.75 sec
$\rho_1 = 2.6$ $\rho_2 = 2.6$ $FR = 3.95$	$\begin{bmatrix} 3.6 & -2.6 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3.6 & -10.27 \\ 0 & 3.95 \end{bmatrix}$	56.49 mpg	109 mph	11.66 sec

occupied by the PG system in the powertrain. The decision on the number of PGs to use should be made including such considerations, as well.

The effect of gear ratios on these optimal results will be discussed in Section 5.4.4.

Also among 2-PG modes, there are some designs which perform better than top



(a) The best configuration

(b) Second best configuration

(c) Third best configuration

Figure 5.7: Top three configurations obtained for the vehicle specifications given in Table 5.1 by enumerating all 2-PG designs

three modes in terms of fuel economy but violate the top speed and 0-60 mph time constraints. Such modes are not useful for single-mode architecture designs but can be useful when designing a multi-mode architecture. For instance, a design that does not satisfy the 0-60 mph time constraint but achieve a good fuel economy can be used together with a mode that has a good acceleration performance. A similar argument can be made for modes that do not satisfy the top speed constraint. They can be used as low-speed modes in addition to the modes that can achieve high speeds. Examples of such cases will be given in Chapter VI.

5.4.2 Illustration Case Study 2

In the second case study, we assume a vehicle with similar specifications to the Chevrolet Volt. The specifications are given in Table 5.4. The differences between previous case study are the vehicle weight and MG specifications. In this case study we also assume that the gear ratios are given as $\rho = 2.24$ and $FR = 2.16$. We design single-mode 1-PG and 2-PG architectures for this vehicle in the same way.

Table 5.4: Vehicle specifications used for Case Study 2

Specification	Value
Vehicle Body Mass	1700 [kg]
Tire Radius	0.3 [m]
Aerodynamic Drag Coefficient	0.29
Frontal Area	2 [m^2]
Battery Voltage	350 [V]
Battery Efficiency	92 [%]
Battery Capacity	6.5 [Ah]
Rated MG1 Power	55 [kW]
Rated MG2 Power	110 [kW]
Max MG1 Speed	6000 [rpm]
Max MG2 Speed	9500 [rpm]
Max MG1 Torque	200 [Nm]
Max MG2 Torque	370 [Nm]
Rated Engine Power	43 [kW]
Max Engine Torque	102 [Nm]
Engine Displacement Size	1.5 [L]

Evaluating all 1-PG designs by enumeration using the same 0-60mph time and top speed constraints of 12 seconds and 105 mph, respectively gives only one feasible design with a fuel economy of 39.23 mpg. Th results of the enumeration is given in Figure 5.8. The next two designs achieve 51.29 mpg and 50.42 mpg, respectively but they violate 0-60 mph time constraint. Figure 5.9 shows all three designs and Table 5.5. The mode shown in 5.14(b) is the hybrid mode used in the Chevrolet Volt. It is expected for this design to violate the 0-60 mph time constraint because the Chevrolet Volt was designed with a multi-mode architecture. As mentioned briefly in the previous section, in a multi-mode architecture not all the modes have to perform well under all driving conditions. While one mode can be used for acceleration, another mode can be used to save fuel after the vehicle accelerates. In other words, multi-mode architectures can benefit from the "division of labor" among the modes. The powertrain components and gear ratios in the Chevrolet Volt was designed to accelerate using pure EV modes in the architecture and save fuel in hybrid mode in

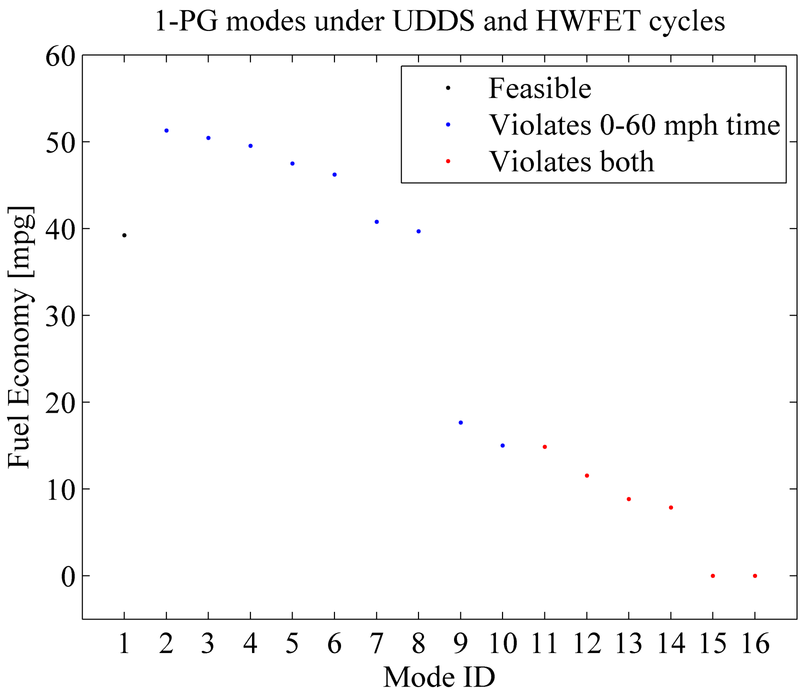


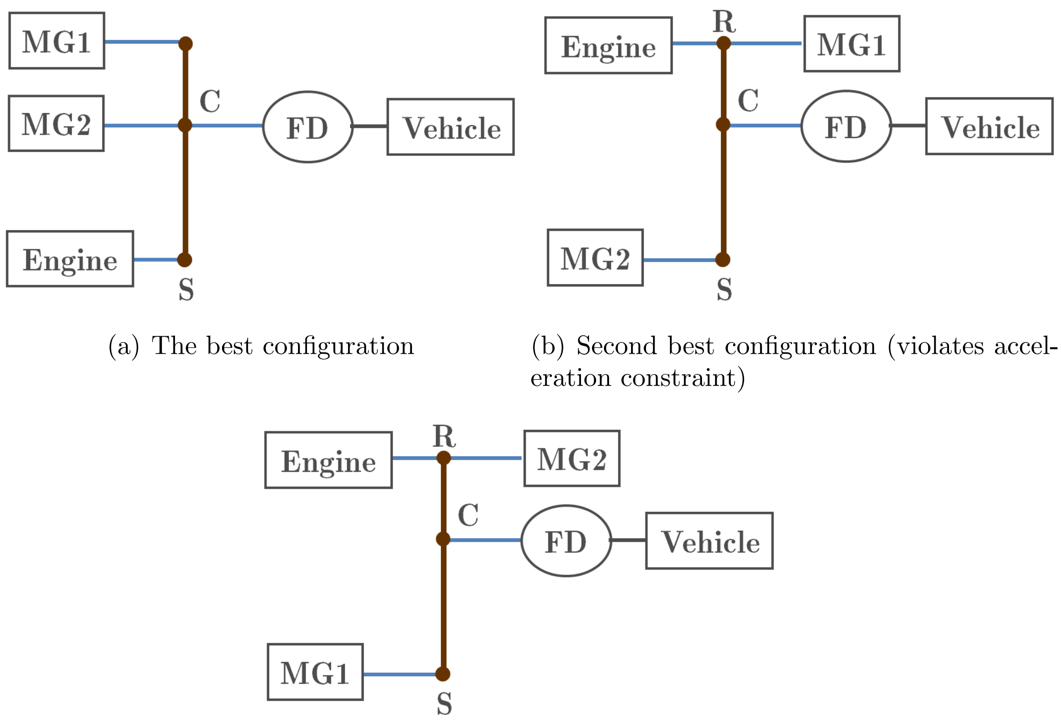
Figure 5.8: Simulation results for all 1-PG modes with $\rho = 2.24$ and $FR = 2.16$ using the vehicle specifications given in Table 5.4

highway speeds. So the mode in 5.14(b) was not designed to accelerate the vehicle.

Table 5.5: Simulation results for the top three 1-PG architectures designed for Case Study 2

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2.24$ $FR = 2.16$	$\begin{bmatrix} 0 & 1 \\ -0.45 & 1.44 \end{bmatrix}$	$\begin{bmatrix} 0 & 2.16 \\ -0.45 & 3.13 \end{bmatrix}$	39.23 mpg	140 mph	10.65 sec
$\rho = 2.24$ $FR = 2.16$	$\begin{bmatrix} 1 & 0 \\ -2.24 & 3.24 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ -2.24 & 7.00 \end{bmatrix}$	47.35 mpg	139 mph	**15.72 sec**
$\rho = 2.24$ $FR = 2.16$	$\begin{bmatrix} -2.24 & 3.24 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -2.24 & 7.00 \\ 1 & 0 \end{bmatrix}$	46.52 mpg	138 mph	**16.40 sec**

When we perform the same analysis among 2-PG designs we obtain the results on Figure 5.10. In this case we have more designs satisfying the constraints. Main reason is that we have more variety of $\mathbf{C}_{mode}$ matrices available among 2-PG modes. Detailed results of the top three designs, which are shown in Figure 5.11, are given



(a) The best configuration

(b) Second best configuration (violates acceleration constraint)

(c) Third best configuration (violates acceleration constraint)

Figure 5.9: Top three configurations obtained for the vehicle specifications given in Table 5.4 by enumerating all 1-PG designs

in Table 5.6.

Table 5.6: Simulation results for the top three 2-PG architectures designed for Case Study 2

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho_1 = 2.24$ $\rho_2 = 2.24$ $FR = 2.16$	$\begin{bmatrix} -2.24 & 3.24 \\ -2.24 & 0 \end{bmatrix}$	$\begin{bmatrix} -2.24 & 7.00 \\ -2.24 & 0 \end{bmatrix}$	51.95 mpg	134 mph	9.33 sec
$\rho_1 = 2.24$ $\rho_2 = 2.24$ $FR = 2.16$	$\begin{bmatrix} -2.24 & 3.24 \\ 3.24 & 0 \end{bmatrix}$	$\begin{bmatrix} -2.24 & 7.00 \\ 3.24 & 0 \end{bmatrix}$	50.73 mpg	122 mph	7.88 sec
$\rho_1 = 2.24$ $\rho_2 = 2.24$ $FR = 2.16$	$\begin{bmatrix} 0 & -2.24 \\ 0.69 & -0.69 \end{bmatrix}$	$\begin{bmatrix} 0 & -4.84 \\ 0.69 & -1.49 \end{bmatrix}$	49.92 mpg	117 mph	7.76 sec

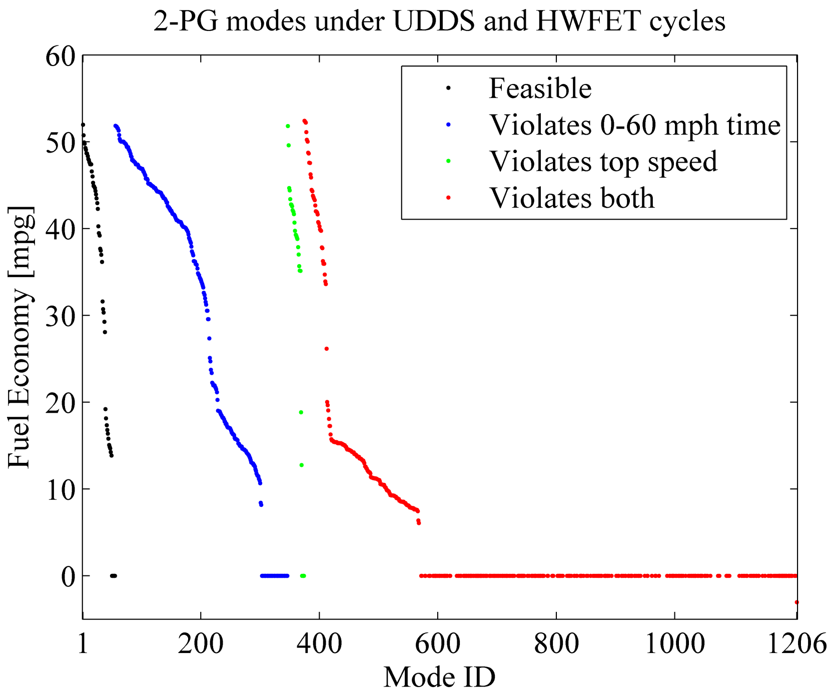
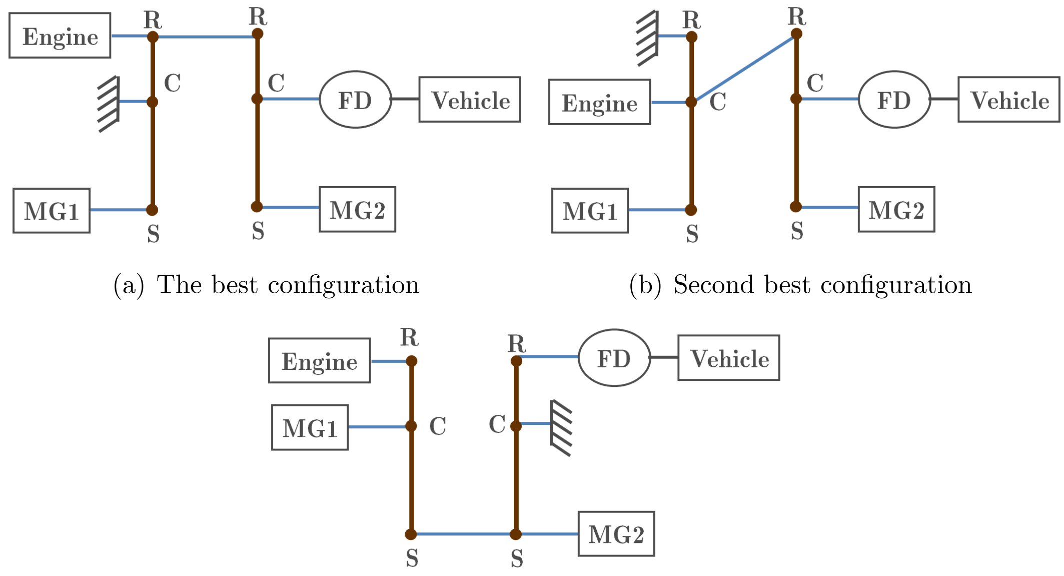


Figure 5.10: Simulation results for all 2-PG modes with $\boldsymbol{\rho} = [2.24; 2.24]$ and $FR = 2.16$ using the vehicle specifications given in Table 5.4

As seen in the top three designs, addition of a ground node to the designs usually increases the acceleration performance. Engine torque limit is generally the limiting factor during acceleration. In the acceleration test, the goal is to obtain maximum torque value at the vehicle output shaft. Having a ground node in the architecture allows larger torque output values in the vehicle shaft while staying within the torque limits of the engine. As will be discussed in Chapter VI grounding engine itself also increases the acceleration performance since we do not need to account for the engine torque limits in such as case.

5.4.3 Illustration Case Study 3

In this case study we design a single-mode power-split architecture for military type of vehicle. The vehicle and powertrain specifications for this case are given in Table 5.7. Military vehicles are usually tested on special drive cycles which also include road grade profiles in addition to the speed profiles (*Masrur et al.*, 2012).



(a) The best configuration

(b) Second best configuration

(c) Third best configuration

Figure 5.11: Top three configurations obtained for the vehicle specifications given in Table 5.4 by enumerating all 2-PG designs

However, for the sake of being consistent with the previous case studies, we will use the same city and highway cycles when selecting the architectures. Since there are no power-split type military examples in mass production, we assume some rough sizes for gear ratios. Let $\rho = 2$ and $FR = 5$.

Table 5.7: Vehicle specifications used for Case Study 3

Specification	Value
Vehicle Body Mass	2400$[kg]$
Tire Radius	0.4$[m]$
Aerodynamic Drag Coefficient	0.45
Frontal Area	3.2$[m^2]$
Battery Voltage	350$[V]$
Battery Efficiency	92[%]
Battery Capacity	13.8$[Ah]$
Rated MG1 / MG2 Power	75$[kW]$
Max MG1 / MG2 Torque	300$[Nm]$
Rated Engine Power	140$[kW]$
Max Engine Torque	470 [Nm]
Engine Displacement Size	6.5$[L]$

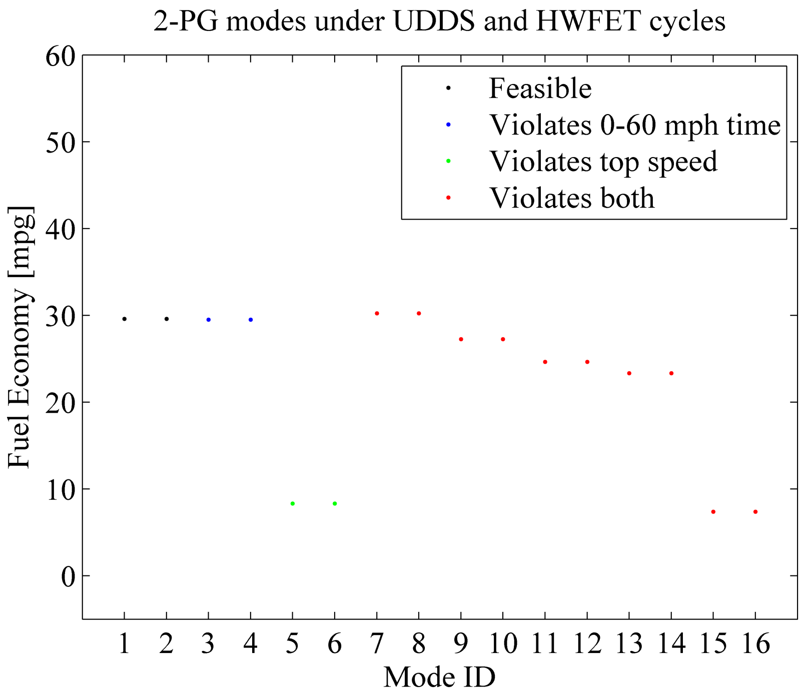


Figure 5.12: Simulation results for all 1-PG modes with $\rho = 2$ and $FR = 5$ using the vehicle specifications given in Table 5.7

Performing the same analysis we did for Case Study 1 and 2 using 1-PG modes gives the results shown in Figure 5.12. Only two 1-PG architectures are feasible and have the same fuel economy. This is mainly because we use two identical MGs and flipping the locations of the motors do not affect the fuel consumption. So the best 1-PG architecture is the same as the best 1-PG architecture for Case Study 1 which is given in Figure 5.5. The details of these two designs are given in Table 5.8.

Table 5.8: Simulation results for the two feasible 1-PG architectures designed for Case Study 3

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2$ $FR = 5$	$\begin{bmatrix} 3 & -2 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3 & -10 \\ 0 & 5 \end{bmatrix}$	29.60 mpg	130 mph	9.40 sec
$\rho = 2$ $FR = 5$	$\begin{bmatrix} 0 & 1 \\ 3 & -2 \end{bmatrix}$	$\begin{bmatrix} 0 & 5 \\ 3 & -10 \end{bmatrix}$	29.60 mpg	130 mph	9.40 sec

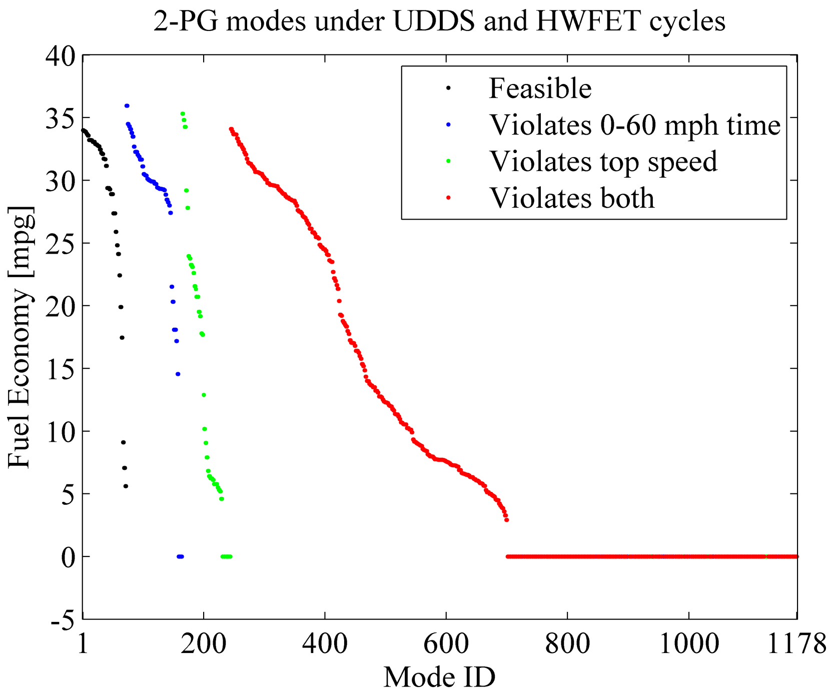
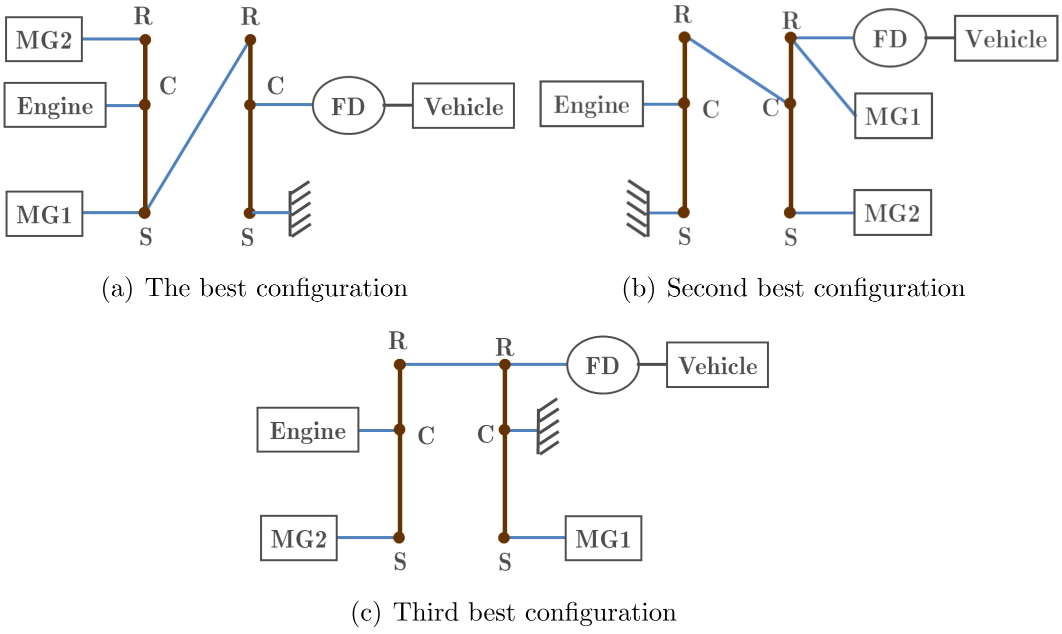


Figure 5.13: Simulation results for all 2-PG modes with $\boldsymbol{\rho} = [2; 2]$ and $FR = 5$ using the vehicle specifications given in Table 5.7

Enumeration over 2-PG designs gives the results in Figure 5.13. Similar to the 1-PG modes, since we use identical MGs, the designs appear in pairs having the same fuel economy and performance. Considering only one of the designs from each pair, top three designs are shown in Figure 5.14. Also the deatiled results for these three designs are given in Table 5.9.

Note that these architectures should be taken as reference to design an architecture for a military vehicle. As mentioned earlier, military vehicles should be tested on special drive cycles and a larger variety of mission scenarios. There should also be additional performance criteria from the architecture such as terrain capability, gradeability etc. The purpose here is to show that for different applications, different architectures need to be designed. One architecture designed for one application does not necessarily be optimal for another application. Also another observation to be made is the improvement in fuel economy from 1-PG designs to 2-PG designs.

In this chapter, we assumed some given gear ratios and powertrain component



(a) The best configuration

(b) Second best configuration

(c) Third best configuration

Figure 5.14: Top three configurations obtained for the vehicle specifications given in Table 5.7 by enumerating all 2-PG designs

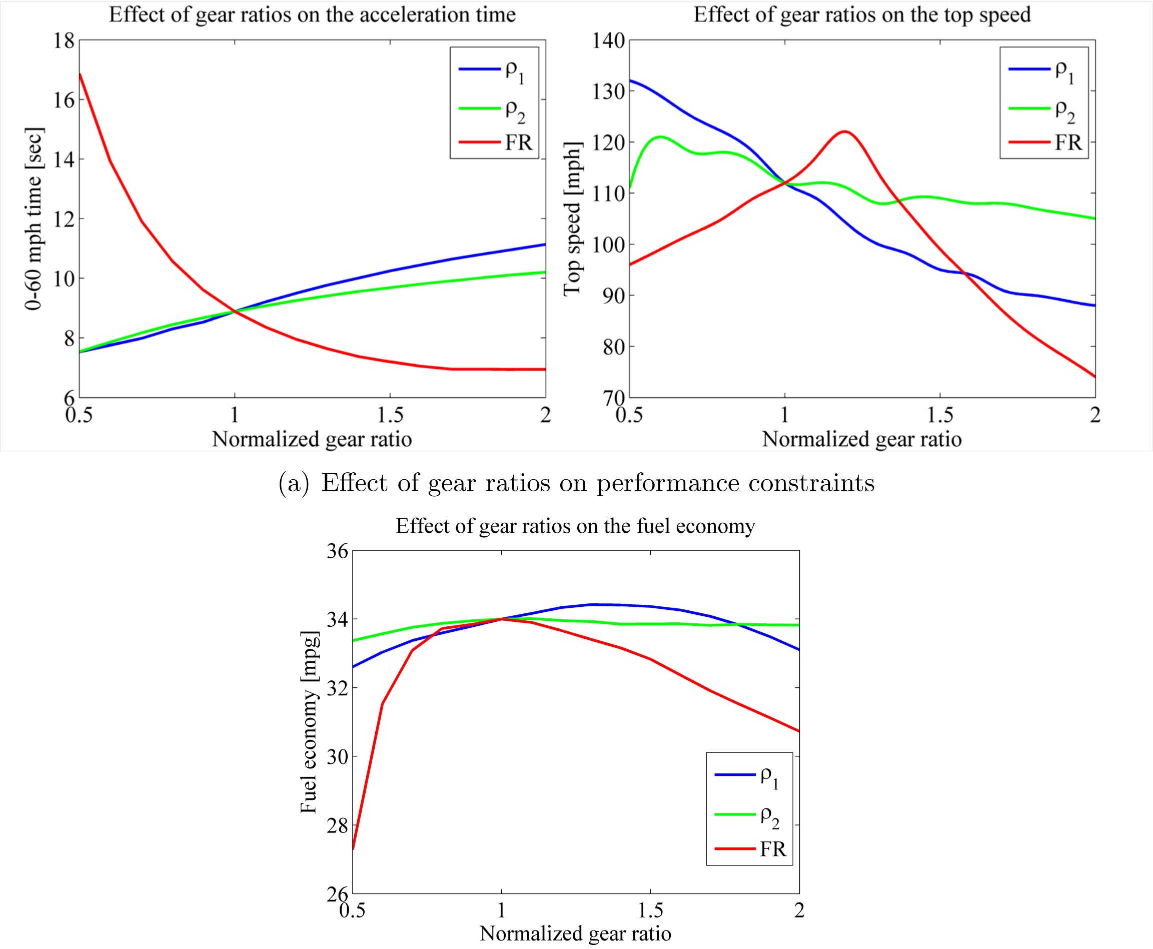
Table 5.9: Simulation results for the top three 2-PG architectures designed for Case Study 3

Given $\mathbf{x}_s$	Optimal $\mathbf{C}_{mode}$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho_1 = 2$ $\rho_2 = 2$ $FR = 5$	$\begin{bmatrix} 0 & 1.5 \\ 1.5 & -0.75 \end{bmatrix}$	$\begin{bmatrix} 0 & 7.5 \\ 1.5 & -3.75 \end{bmatrix}$	34.00 mpg	112 mph	8.89 sec
$\rho_1 = 2$ $\rho_2 = 2$ $FR = 5$	$\begin{bmatrix} 0 & 1 \\ 4.5 & -2 \end{bmatrix}$	$\begin{bmatrix} 0 & 5 \\ 4.5 & -10 \end{bmatrix}$	33.94 mpg	113 mph	11.12 sec
$\rho_1 = 2$ $\rho_2 = 2$ $FR = 5$	$\begin{bmatrix} 0 & -2 \\ 3 & -2 \end{bmatrix}$	$\begin{bmatrix} 0 & -10 \\ 3 & -10 \end{bmatrix}$	33.86 mpg	111 mph	7.74 sec

specifications before the design process. Powertrain components may be sized considering the power requirements from the architecture but gear ratio sizes are not available all the time, as it is the in the present case study. In the next section, we will show the effect of gear ratios on the fuel consumption with a parametric study. Also military vehicles should be designed considering different loading scenarios. They might carry payloads as heavy as the vehicle weight. The effect of vehicle weight will also be discussed in the next section.

5.4.4 Discussion

We showed the single-mode architecture design for three case studies assuming some gear ratios. We showed that different vehicle classes and powertrain component sizes require different architectures. We can also take a look at the effect of the gear ratio selections on the fuel economy with a parametric study. In order to show the significance of these values we choose the best 2-PG designed obtained in Case Study 3. Three parameters we can vary are ρ_1, ρ_2, FR. We vary only one parameter at a time. Figure 5.15 shows the effect of all three parameters on fuel economy, top speed and 0-60 mph acceleration time. The horizontal axis is the normalized parameter value where "1" represents the original values we used in the previous section which were $\rho_1 = 2$, $\rho_2 = 2$ and $FR = 5$. From the figure, we see that all these parameters have a significant impact on both the fuel economy and vehicle performance. Although our initial selection of gear ratios provide good fuel economy, there is some potential to further improve the fuel economy (for example, by using a larger ρ_1). An important point to be noted is that optimal gear ratios vary from one configuration to another. In that sense, evaluating all configurations with the same gear ratios is not a fair comparison unless the gear ratios are fixed by some tight constraints. Optimization of gear ratios can be performed for a particular configuration using a gradient-based algorithm such as Sequential Quadratic Programming. However, in order to make a fair comparison, optimizing the gear ratios for all configurations is not computationally tractable for systems with more than one PG. Chapter VII proposes an alternative formulation in order to make the configuration and gear ratio design decisions simultaneously.



(a) Effect of gear ratios on performance constraints

(b) Effect of gear ratios on fuel economy

Figure 5.15: Effects of the parameters on the simulation results obtained for the architecture in Figure 5.14(a), where"1" represents the original values $\rho_1 = 2$, $\rho_2 = 2$ and $FR = 5$.

However, such an approach require a different solution process that will be discussed in Chapter VII.

5.5 Summary

In this chapter we formulated the single-mode HEV architecture design problem with fixed component sizes. We presented a general vehicle model which can be used to evaluate the fuel consumption and performance of both single-mode and multi-mode architectures. The same models will also be used in Chapters VI and VII. Also a review of the supervisory control algorithms was given and ECMS was selected among them because of the computational efficiency. We described the advantages and limitations of this control strategy. Given the vehicle model and control strategy, we presented three case studies with different parameters and selected 1 and 2-PG architectures for each vehicle using enumeration.

In order to identify the key parameters, we performed a parametric study on gear ratios and final drive ratio. Since they have a significant impact on the fuel consumption, a better design study should include them as design variables. However, adding these design variables to the problem poses new challenges since the enumeration of all possible design will not be possible any more. Chapter VII proposes a decomposition-based formulation in order to design the configuration and component sizes simultaneously.

CHAPTER VI

Multi-mode Hybrid Electric Vehicle Architecture Design with Fixed Sizing

In Chapter III we presented a modified bond graph representation for HEV architectures and described a systematic process to generate all feasible configurations, i.e., driving modes, for any given set powertrain components connected through any number of planetary gears (PGs). Chapter IV described the general mathematical formulation of HEV architecture design problems for different scenarios utilizing the feasible driving modes we generated. In Chapter V we solved the architecture design problem for single-mode architectures for given powertrain component sizes and gear ratio selections. In this chapter we will solve the multi-mode architecture design problem with the same assumptions on the component sizes and gear ratios.

The main motivation to design multi-mode architectures is to benefit from the a variety of "specialized" modes each of which operate efficiently under different driving conditions. For instance, a mode that is designed to operate at highway driving conditions does not have to operate well for city driving conditions. However, it generally operates better than a mode designed considering all driving conditions.

The design of multiple modes in the same architecture poses new challenges. The difference between single-mode and multi-mode architecture mainly comes from the number of possible design alternatives. While the number of alternatives is small enough to allow for enumeration over all possible designs for a single-mode architecture design study, the same explicit enumeration cannot be applied to multi-mode architectures since the number of design alternatives is much larger. In this chapter we propose a different search method utilizing implicit enumeration. Additionally, when starting from the driving modes to create architecture which can be considered as a bottom-up approach, we need to consider the how to combine the modes in the same architecture. The connection differences between the modes to combine, require clutches that allow switching from one configuration to another. The modes that have very different connections require many clutches in order to be combined together. We penalize such cases by defining a *complexity* constraint for the design of multi-mode architectures.

6.1 Problem Formulation

A mathematical formulation for the general architecture design problem including component sizing as design decisions was given in Chapter IV. When the powertrain component sizes are given, this general formulation is reduced to a simpler form as follows:

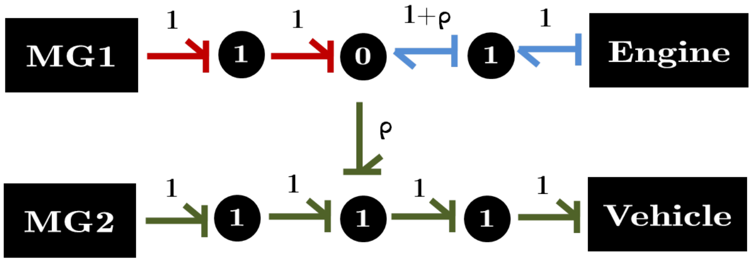
$$\begin{aligned}
\text{min} \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{p})) \\
\text{w.r.t.} \quad & \{\mathbf{x}_c(N_{mode})\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}, \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{p})), \mathbf{p}) \leq \mathbf{0} \\
& g_{complex}(\mathbf{x}_c(N_{mode})) \leq \mathbf{0} \\
& N_{mode} \in \{1, 2, 3, 4, ...\} \\
& \mathbf{x}_c(N_{mode}) \text{ is feasible}
\end{aligned} \tag{6.1}$$

where f_{cons} is the fuel consumption calculated using the optimal control policy ψ^* in a nested formulation. We gave an overview of the control strategies developed for HEV powertrains in Chapter V. Also, $\mathbf{g}_{perf}$ is the vector of vehicle performance constraints such as acceleration time, top speed, gradeability etc.; $\mathbf{x}_c$ is the vector representing the configuration of all modes in the architecture and the size of this vector depends on the number of modes, N_{mode}, in the architecture. The configurations must be selected from the feasible set of driving modes generated earlier. The given component sizes including the gear ratios and other vehicle parameters are represented by the parameter vector $\mathbf{p}$. Additionally, in the multi-mode architecture design, g_{comp} denotes the complexity constraint which is defined in Section 6.2.

At least one of the modes in a multi-mode HEV architecture must be a hybrid mode in order to be able to charge the battery to the initial state of charge (SOC) level at the end of the drive cycle and the other modes can be either hybrid or pure electric modes. Therefore, the number of possible multi-mode architecture design alternatives is $M \times M^{N_{mode}-1}$ where M denotes the number of feasible configurations to select from for a given number of PGs and a set of powertrain components. This number is much larger compared to the single-mode architecture design possibilities and explicit enumeration cannot be used to solve the problem in given in Equation (6.1). Section 6.3 describes the solution strategy we propose in this dissertation.

6.2 Architecture Complexity

A multi-mode architecture is created by combining two or more driving modes under a single arrangement of its components. Having multiple modes in the architecture provides flexibility to switch component connectivity during vehicle operation resulting in better performance and fuel efficiency under different driving conditions. An example for a commercially available architecture with multiple modes was given in Figure 3.1(a). Switching among the modes is achieved by engaging or disengaging clutches. A change in connectivity among the driving modes in the architecture requires one or more clutches, and a systematic way to evaluate the required clutches



	S1	R1	C1
Engine	0	0	1
Vehicle	0	1	0
MG1	1	0	0
MG2	0	1	0
Ground	0	0	0
S1	0	0	0
R1	0	0	0
C1	0	0	0

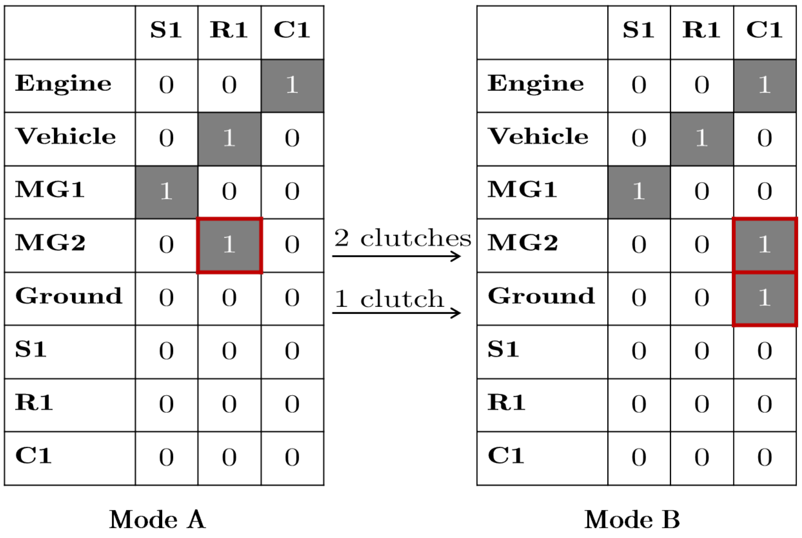
Figure 6.1: Example of a modified bond graph representation and its connectivity table

is a crucial element in the architecture design process. Having too many clutches in the architecture is not desired since it would result in increased complexity in clutch control, efficiency losses and cost. So, the number of clutches in an architecture can be used as a measure of *complexity* of the architecture. In this section, we adopt this measure of complexity and we describe the process to calculate the number of clutches needed to combine any given two modes. We then generalize the process to any N_{mode} modes. This complexity measure will be used as an additional design criterion in the proposed architecture design framework.

We start by introducing a binary connectivity table representing the connections among external components and PG nodes that is extracted from our modified bond graph representation. An example is given in Figure 6.1 that shows both the modified bond graph and connectivity table of the 1-PG Toyota Prius mode. For a general case, say n-PG mode with J^{ext} number of external components, the size of the connectivity table is $(J^{ext}+3n)\times 3n$ where the upper J^{ext} rows correspond to the external components and the remaining $3n$ rows and columns correspond to the sun, ring and carrier gears of each PG. Given a modified bond graph representing a driving mode, a table entry will have “1” when the corresponding two components or gears are connected and “0” otherwise. Also the connectivity table might have multiple “1”s in a row when an external component is connected a gear of a PG set and that gear is connected to another gear of a different PG set.

We explain the process of extracting this information from the modified bond graph using a color analogy. The weights around a 0-junction represents the kinematic relationships among sun, ring and carrier gears and every bond around 0-junction can be assigned to a different color representing each of these respective gears. In Figure 6.1, the edge weights representing sun, ring and carrier gear are colored by red, green and blue, respectively. As seen in this figure, since a 1-junction keep the same kinematic relationship, i.e. the speeds around the 1-junction are the same, the same coloring is kept around a 1-junction. When this process is repeated for all bonds we can see the connection information of each external component by checking the color of the bond connecting an external component to its corresponding external junction. In case of a 2-PG mode, we need 6 colors; for 3-PG modes we need 9 colors and so on. When multiple 0-junctions impose different colors on a bond, i.e. when colors are mixed on a bond, it means this bond is connected to multiple PGs. Such color mixtures give information on the internal connections among PGs.

Once the connectivity table is extracted from the modified bond graph, one can calculate the number of clutches needed by comparing the connectivity tables of the driving modes to be combined. Basically adding or removing a connection requires one clutch and changing a connection of an external component from one gear to another requires two. First let’s look at the case where we have two modes to be combined. Figure 6.2 shows a simple example that compares connectivity tables of two 1-PG modes and identify the clutches required. In the figure, Mode B has an additional ground connected to the carrier that doesn’t exist in Mode A. Adding this ground connection requires one clutch at the carrier gear. Also MG2 is connected to



	S1	R1	C1
Engine	0	0	1
Vehicle	0	1	0
MG1	1	0	0
MG2	0	1	0
Ground	0	0	0
S1	0	0	0
R1	0	0	0
C1	0	0	0

Mode A

	S1	R1	C1
Engine	0	0	1
Vehicle	0	1	0
MG1	1	0	0
MG2	0	0	1
Ground	0	0	1
S1	0	0	0
R1	0	0	0
C1	0	0	0

Mode B

2 clutches
1 clutch

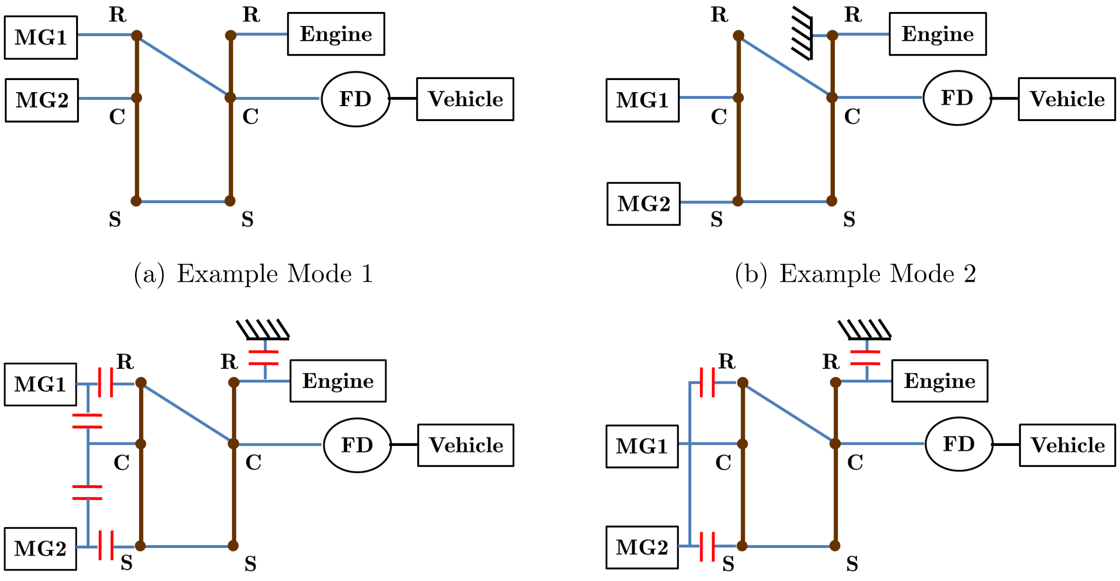
Figure 6.2: Two sample connectivity tables and the corresponding clutching solution indicated by red boxes

the ring gear in Mode A and connected to carrier gear in Mode B. Disconnecting MG2 from ring requires a clutch at the ring and connecting it to the carrier requires another clutch at the carrier gear. This process of comparing the tables can be automated.

When MGs used in the architecture are identical an extra step is necessary. In such a case, since the numbering of the MGs is arbitrary, multiple clutching solutions are possible. Figure 6.3 shows an example for a 2-PG system with an engine, two MGs and a ground. By swapping the definitions of MGs, a simpler and more preferred clutching solution can be found as shown on Figure 6.3(d). In terms of connectivity table, swapping the rows corresponding to the MGs will represent this solution as shown on Figure 6.4.

A similar idea applies to the case when the PGs used in the system are identical, i.e. have the same gear ratio. In that case, the rows and columns corresponding to the PGs should be swapped to check if a simpler clutching solution is possible.

In the general case with $N_{mode} > 2$ number of modes to be combined, the process described above can be applied sequentially. The connectivity tables of the modes can be compared pairwise and the union of the clutches found gives the final clutching



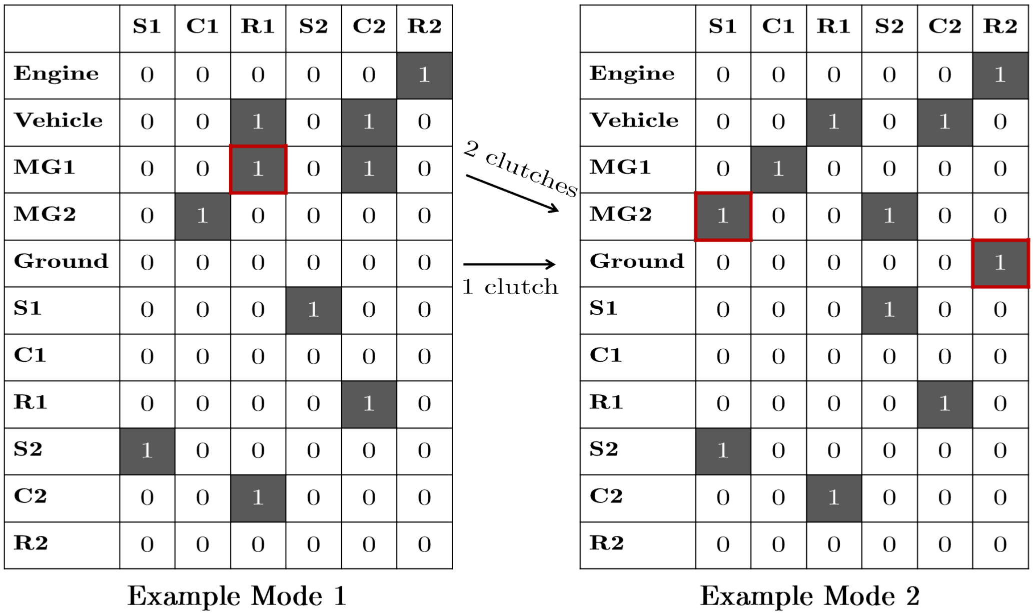
(a) Example Mode 1

(b) Example Mode 2

(c) Clutching Solution 1

(d) Clutching Solution 2

Figure 6.3: Multiple clutching solutions exist when MG1 and MG2 are identical


2 clutches
1 clutch

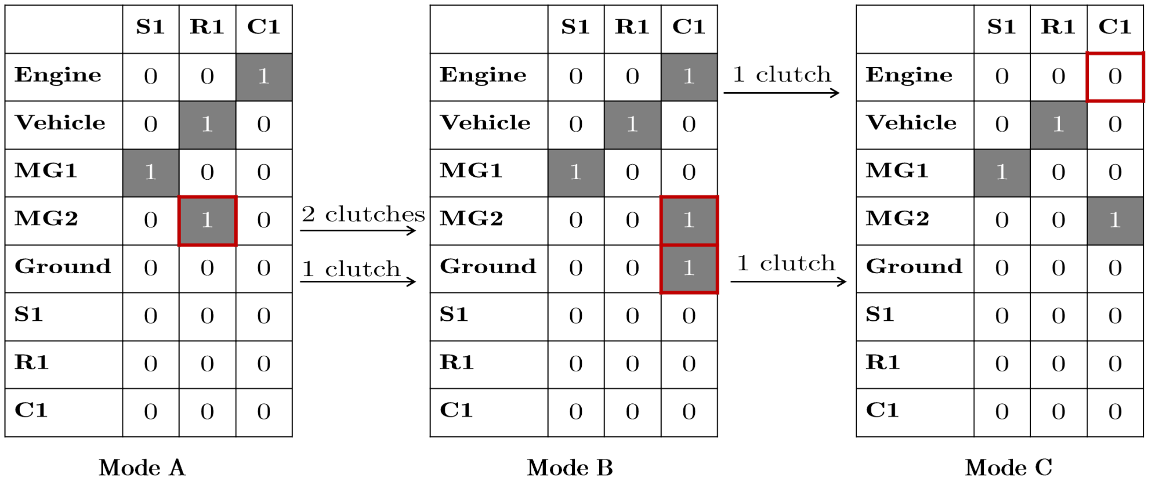
Example Mode 1

	S1	C1	R1	S2	C2	R2
Engine	0	0	0	0	0	1
Vehicle	0	0	1	0	1	0
MG1	0	0	1	0	1	0
MG2	0	1	0	0	0	0
Ground	0	0	0	0	0	0
S1	0	0	0	1	0	0
C1	0	0	0	0	0	0
R1	0	0	0	0	1	0
S2	1	0	0	0	0	0
C2	0	0	1	0	0	0
R2	0	0	0	0	0	0

Example Mode 2

	S1	C1	R1	S2	C2	R2
Engine	0	0	0	0	0	1
Vehicle	0	0	1	0	1	0
MG1	0	1	0	0	0	0
MG2	1	0	0	1	0	0
Ground	0	0	0	0	0	1
S1	0	0	0	1	0	0
C1	0	0	0	0	0	0
R1	0	0	0	0	1	0
S2	1	0	0	0	0	0
C2	0	0	1	0	0	0
R2	0	0	0	0	0	0

Figure 6.4: Connectivity tables for the example modes in Figure 6.3. The minimum number of clutches required is 3, when the two MGs are identical.



	S1	R1	C1
Engine	0	0	1
Vehicle	0	1	0
MG1	1	0	0
MG2	0	1	0
Ground	0	0	0
S1	0	0	0
R1	0	0	0
C1	0	0	0

Mode A

	S1	R1	C1
Engine	0	0	1
Vehicle	0	1	0
MG1	1	0	0
MG2	0	0	1
Ground	0	0	1
S1	0	0	0
R1	0	0	0
C1	0	0	0

Mode B

	S1	R1	C1
Engine	0	0	0
Vehicle	0	1	0
MG1	1	0	0
MG2	0	0	1
Ground	0	0	0
S1	0	0	0
R1	0	0	0
C1	0	0	0

Mode C

Figure 6.5: Three sample connectivity tables and the corresponding clutching solution indicated by red boxes

solution. Figure 6.5 shows an example case with 3 modes. In order to shift from Mode A to Mode B, three clutches are required: the first one is between MG2 and ring gear, the second one is between MG2 and carrier gear, and the third one is between ground and carrier gear. In order to shift from Mode B to Mode C, two clutches are required: the first one is between ground and carrier gear, and the second one is between engine and carrier gear. When take the union these clutches, we obtain four clutches since we count the clutch between ground and carrier gear twice. Note that this solution is independent from the order of the modes. If we follow the same steps switching from Mode A to Mode C and from Mode C to Mode B, we obtain the same set of clutches, for instance.

Finally, given the definition of architecture *complexity* measured by the number of clutches in the system and calculated by the process described above, a complexity design characteristic for an architecture can be formally defined. This characteristic should be minimized for each architecture, i.e., the simplest clutching solution should be selected and it should be bounded above in the architecture design optimization framework to eliminate designs beyond the desired complexity.

6.3 Architecture Search

Chapter III described an enumeration process to generate all possible driving modes given the external components. If the designs of interest are single-mode architectures only and all other parameters such as component sizes and gear ratios are fixed, all architecture alternatives can be evaluated one by one without any need for a sophisticated search method. However, in the case of multiple modes the architecture alternatives increase exponentially. For instance, the case reported in Chapter III had 3420 possible modes. A 2-mode hybrid architecture has approximately 7.2 million alternatives making explicit enumeration intractable. As a result, a search method capable of identifying promising multi-mode architectures is necessary.

This section describes a multi-mode architecture design framework shown in Figure 6.6. First, a general vehicle model is described. Next, power management strategies available in the literature for distributing the power demand among engine and MGs are explained. Finally, a search method to explore the design space of multi-mode architectures is shown.

6.3.1 Search Algorithm

For an n-mode architecture design, the goal of the search algorithm is to maximize the fuel economy by selecting the *feasible* n modes from the generated design space. *Feasibility* of a mode selection is defined by three criteria:

(i) At least one of the modes must satisfy the speed and torque requirements to follow the drive cycle at all time steps.

(ii) The architecture must have a charge-sustaining capability.

(iii) The complexity of the architecture must be less than a specified threshold.

The first criterion is a very conservative requirement. If the architecture cannot supply the demand even at a single time step, it is considered infeasible. The second

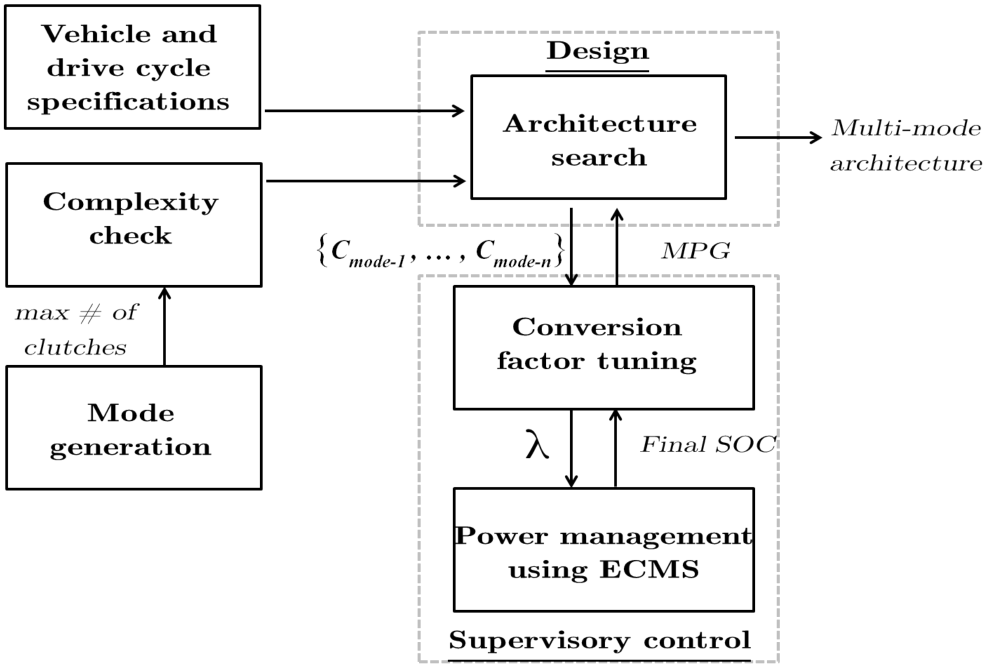


Figure 6.6: Flowchart of the dual-mode architecture design process

criterion simply constrains pure EV architectures: At least one engine is required in order to satisfy the charge-sustaining requirement as described in the previous section. This requirement can be relaxed if the designs of interest are pure EV architectures. The last criterion puts an upper bound on the number of clutches required to combine the modes. It is a practicality constraint as a large number of clutches in a system imposes additional complexity in the mode switching mechanism and cost.

The method proposed to search the design space of n modes is an application of the so-called *taxi cab method.* In this method, given an initial point in an n-dimensional space represented by n basis vectors, the next point is searched along one basis vector per iteration. The basis vector representing the search direction is switched at every iteration. In most engineering problems, the search in a direction starts from a point and continues until the objective stops improving. However, in the case of searching for modes, the search has to be performed over all elements in a direction since the modes are not ordered in a meaningful way.

More specifically, the search for an n-mode architecture starts with an initial set of n modes, $\mathbb{M}_0 = \{M_1, M_2, ..., M_{n-1}, M_n\}$. In the first iteration, keeping the first $n-1$ modes fixed, an n^{th} mode is searched among all feasible modes to maximize the fuel economy. When the search for the n^{th} mode is complete and the mode set is updated as $\mathbb{M}_1$, a search for $(n-1)^{th}$ mode starts keeping all other modes fixed. Since the mode set $\mathbb{M}_1$ is among the alternatives evaluated in this iteration, the fuel economy has to improve or remain the same as in the previous iteration. The search algorithm terminates when the same set of modes is selected (i.e., no improvement in the fuel economy) in two consecutive iterations. Since there is a finite number of mode selections, the algorithm is guaranteed to converge in a finite number of iterations. The findings of the method depend on the initial mode selection. So, different initial points should be tested to get reliable results. For demonstration purpose only, Figure 6.7 visually depicts the iterations of the search algorithm for a fictitious case with two modes with 9 possible modes. In the figure, the search starts with the mode selection $\mathbb{M}_0 = \{8, 1\}$ and searches in the direction of the second mode. At the end of the first iteration it finds $\mathbb{M}_1 = \{8, 7\}$ and starting from that point, it searches in the direction of the first mode. Second iteration finds $\mathbb{M}_2 = \{3, 7\}$. Searching the direction of the second mode again, gives $\mathbb{M}_3 = \{3, 4\}$. Searching in the direction of the first mode does not improve the result giving the same mode selection $\mathbb{M}_4 = \{3, 4\}$ and the search terminates.

6.4 Case Study Illustration Results

This section gives the architecture design results for three case studies in order to demonstrate the capability of the process described in Section 6.3. For demonstration, we use the same three applications we discussed in Chapter V.

In addition to these specifications, the controller initial battery *SOC* is set to 60% and *SOC* is allowed to vary between 40% and 80%. In the multi-mode HEV control,

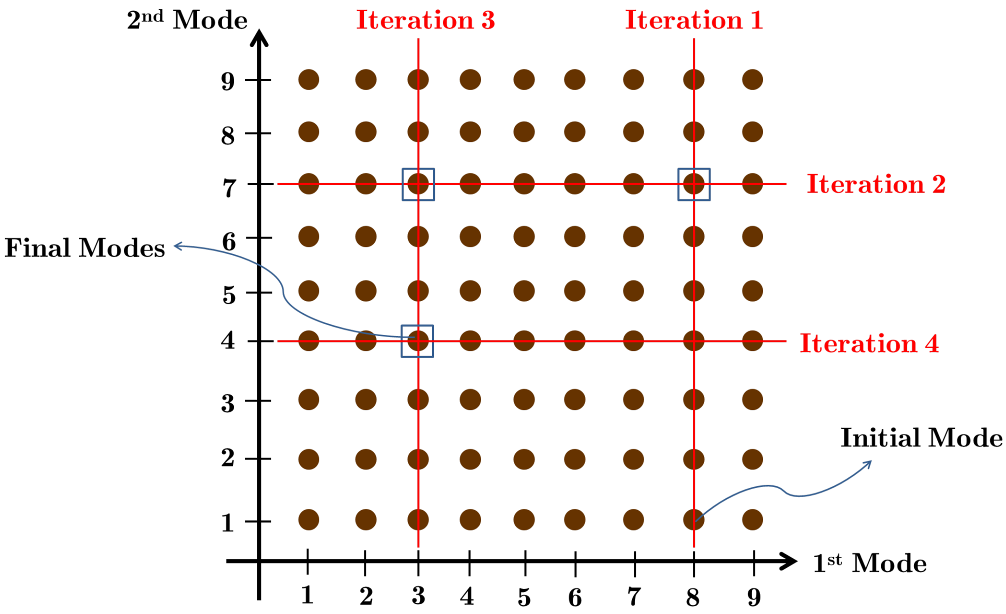


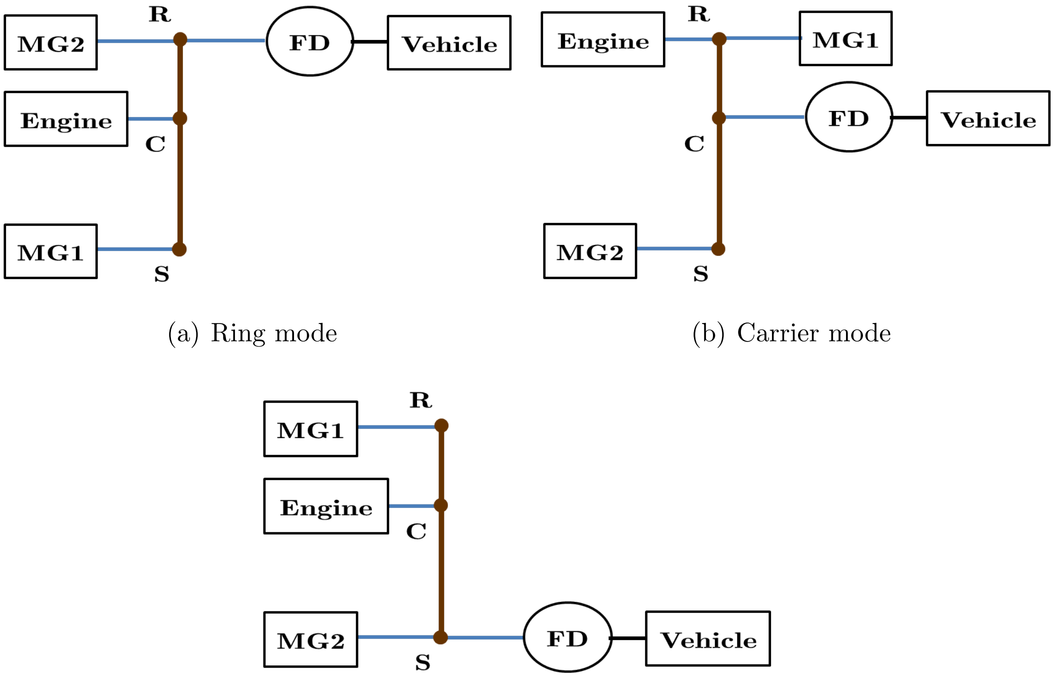
Figure 6.7: Iterations of the search algorithm for a fictitious problem starting with $\mathbb{M}_0 = \{8, 1\}$ and converging to $\mathbb{M}_4 = \{3, 4\}$ in four iterations. Minimum of each iteration is denoted by a square

Table 6.1: Architecture specifications for all three case studies

Specification	Value
Number of PGs	1 or 2
External components	1 Engine 2 MGs Vehicle output 1 (optional) ground
Mode type	Hybrid or pure-EV
Number of modes	2
Max. number of clutches	2

the mode shifting is managed using a shifting penalty as described in Chapter V. The mode shifting penalty is experimentally tuned to 0.1 in order to obtain a reasonable mode shifting strategy and it is fixed for all architecture alternatives. Clutch and gear losses are not included in the model.

The objective of the case studies is to maximize the fuel economy with vehicle performance constraints in addition to the *complexity* as described in the previous



(a) Ring mode

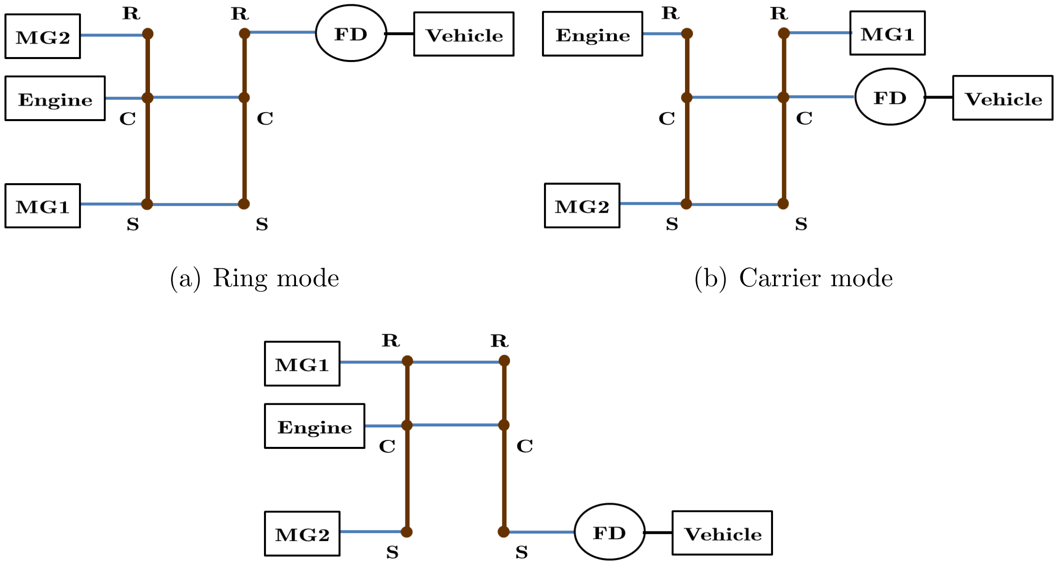
(b) Carrier mode

(c) Sun mode

Figure 6.8: Initial modes for 1-PG architecture design studies

section. The design variables are the selections of feasible modes with 1 or 2 PGs connecting an engine, 2 MGs, a vehicle output and an optional ground. Table 6.1 summarizes the specifications of the architecture alternatives considered. The design process described in Section 6.3 requires initial modes to start the search. We assume that the output shaft remains fixed, i.e., we do not allow any clutch to be connected to the vehicle output shaft. In order to search different types of modes where the vehicle output shaft is connected to different gears of the PG sets, we choose three initial designs for both 1-PG and 2-PG design studies. The same initial modes are used for three case studies and they are given on Figures 6.8 and 6.9.

For each case study a different set of fixed gear ratios are used and we assume that these values are given beforehand.



(a) Ring mode

(b) Carrier mode

(c) Sun mode

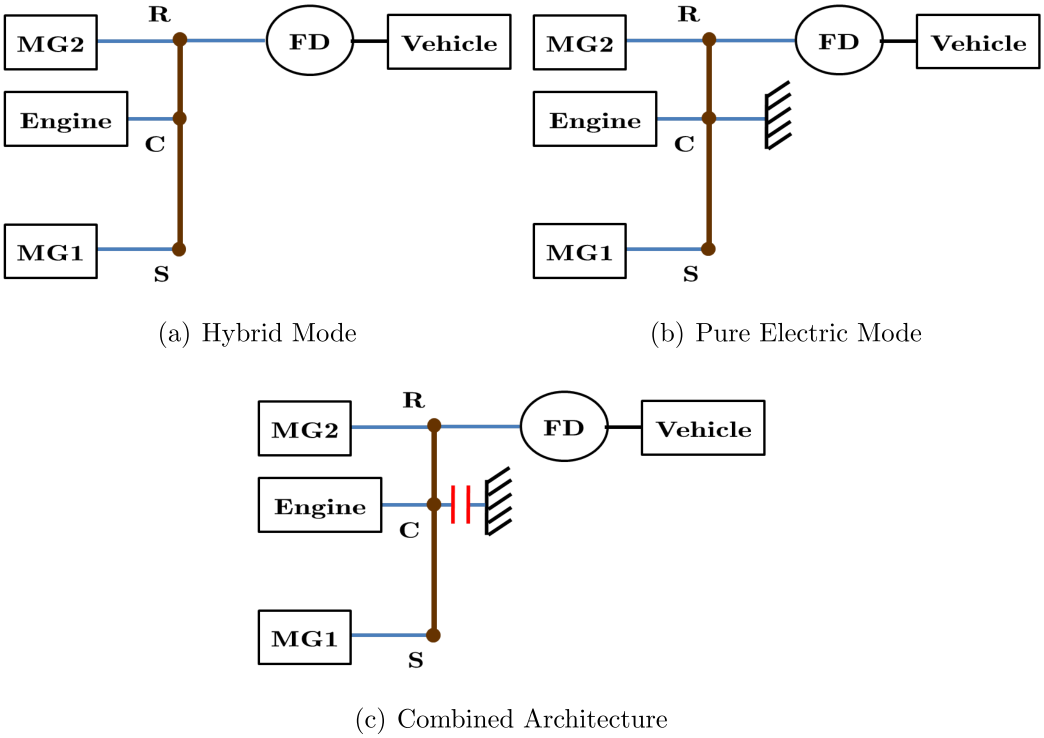
Figure 6.9: Initial modes for 2-PG architecture design studies

6.5 Case Study 1

The vehicle and powertrain component specifications used for this case study is given in Tabletbl:ch6case1specs.

Table 6.2: Vehicle specifications used for Case Study 1

Specification	Value
Vehicle Body Mass	1400 [kg]
Tire Radius	0.3 [m]
Aerodynamic Drag Coefficient	0.29
Frontal Area	2 [m^2]
Battery Voltage	350 [V]
Battery Efficiency	92 [%]
Battery Capacity	6.5 [Ah]
Rated MG1 Power	42 [kW]
Rated MG2 Power	60 [kW]
Max MG1 / MG2 Speed	12000 [rpm]
Max MG1 / MG2 Torque	200 [Nm]
Rated Engine Power	43 [kW]
Max Engine Torque	102 [Nm]
Engine Displacement Size	1.5 [L]



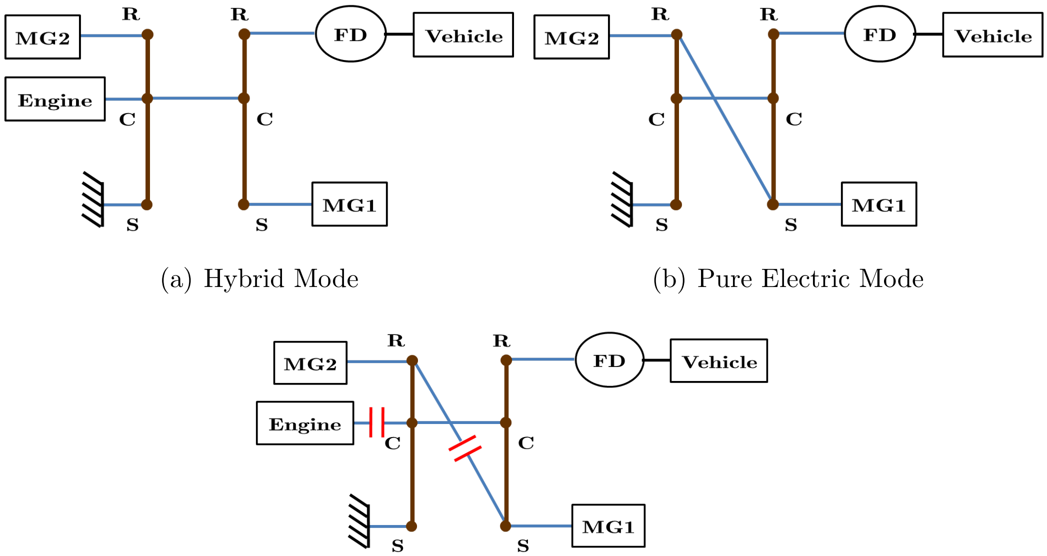
(a) Hybrid Mode

(b) Pure Electric Mode

(c) Combined Architecture

Figure 6.10: Best 1-PG architecture found for Case Study 1

For the 1-PG architecture design study we assume that the gear ratios are given as $\rho = 2.6$ and $FR = 3.95$. Starting from the three initial designs given in Figure 6.8 we obtain the 3 different dual-mode architectures with fuel economies of 58.13 mpg, 57.69 mpg, and 52.64 mpg. The design with the best fuel economy is obtained with starting from the mode shown in Figure 6.8(a) and the best architecture obtained is shown in Figure 6.10. As it is seen the initial hybrid mode and the hybrid mode in the best design are the same. Since there are only 16 hybrid configurations and 36 pure electric configurations, there are not many good options to change the initial design. Recall from Chapter V that this mode is the best configuration selected for the single-mode architecture design for this case study. This is also the mode implemented in the Toyota Prius. The detailed results for this multi-mode architecture is given in Table 6.3. The search method finds an additional pure electric mode that improves the single mode architecture capabilities. For instance, by adding this pure electric mode to the single-mode architecture of Chapter V, the fuel economy improved from 57.83 mpg to



(a) Hybrid Mode

(b) Pure Electric Mode

(c) Combined Architecture

Figure 6.11: Best 2-PG architecture found for Case Study 1

59.50 mpg. A similar conclusion was made by *Zhang et al.* (2012). Additionally, the 0-60 mph acceleration time dropped from 11.81 seconds to 7.6 seconds in this case.

When we perform the same search for 2-PG designs using the initial modes given in Figure 6.9 we converge to another three different designs with fuel consumptions of 60.81 mpg, 58.85 mpg and 52.63 mpg, respectively. We select the best one which was obtained starting from the design in Figure 6.9(a). No the difference between initial and final hybrid modes. It shows that the method can modify the initial design to improve the results. However, this capability is problem dependent. A counter example is given in Case Study 2. The detailed results for this architecture is given in Table 6.3.

The improvement in all three specifications going from 1-PG to 2-PG multi-mode design can be observed on the table. These results might further be improved by testing multiple initial points.

Table 6.3: Simulation results of the selected designs for Case Study 1

Given $\mathbf{x}_s$	Optimal Design	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2.6$ $FR = 3.95$	Figure 6.10	58.13 mpg	109 mph	7.6 sec
$\rho_1 = 2.6$ $\rho_2 = 2.6$ $FR = 3.95$	Figure 6.11	60.81 mpg	127 mph	6.8 sec

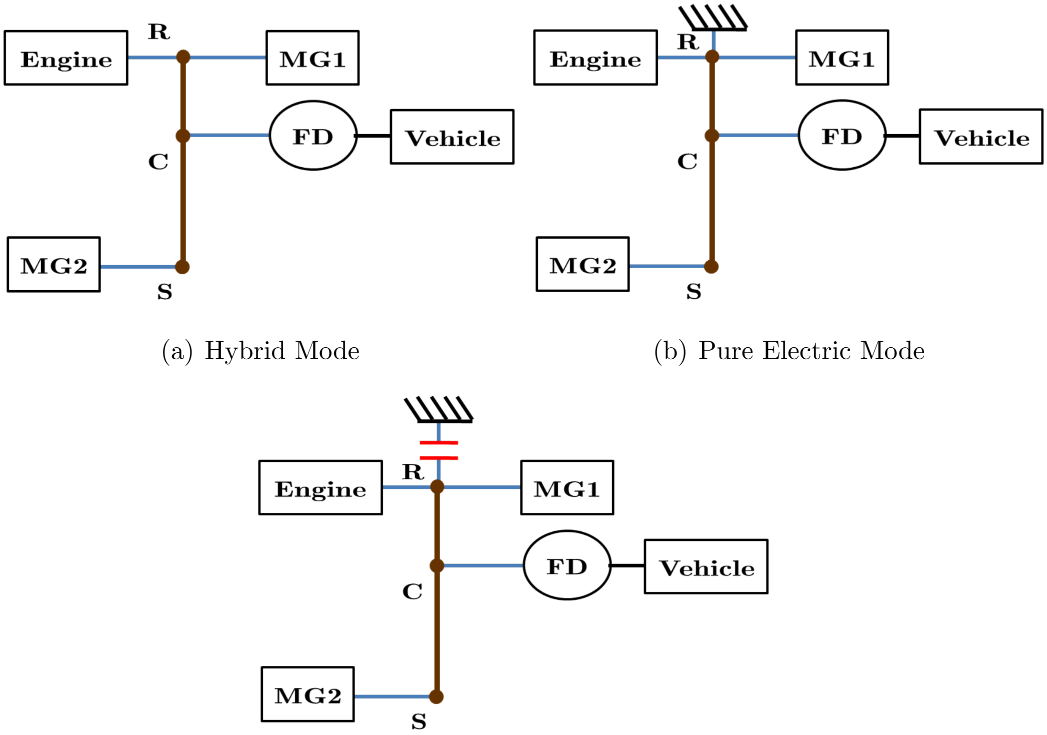
6.6 Case Study 2

We test the design framework in another case study with the following specifications:

Table 6.4: Vehicle specifications used for Case Study 2

Specification	Value
Vehicle Body Mass	1700 [kg]
Tire Radius	0.3 [m]
Aerodynamic Drag Coefficient	0.29
Frontal Area	2 [m^2]
Battery Voltage	350 [V]
Battery Efficiency	92 [%]
Battery Capacity	6.5 [Ah]
Rated MG1 Power	55 [kW]
Rated MG2 Power	110 [kW]
Max MG1 Speed	6000 [rpm]
Max MG2 Speed	9500 [rpm]
Max MG1 Torque	200 [Nm]
Max MG2 Torque	370 [Nm]
Rated Engine Power	43 [kW]
Max Engine Torque	102 [Nm]
Engine Displacement Size	1.5 [L]

Using the same 1-PG initial designs as we used in Case Study 1, we obtain three architectures with the fuel economies of 46.57 mpg, 48.35 mpg and 40.39 mpg, respectively. The design with the best fuel economy is shown in Figure 6.10. This design



(a) Hybrid Mode

(b) Pure Electric Mode

(c) Combined Architecture

Figure 6.12: Best 1-PG architecture found for Case Study 2

consists of two modes of the Chevrolet Volt architecture. Recall that Chevrolet Volt has one more pure electric mode and one more series hybrid mode. Also comparing this result with the single-mode architecture results, we can see that the hybrid mode we have found for the multi-mode architecture appeared to be an infeasible design due to the 0-60 mph time constraint. However, in this multi-mode architecture, the additional pure electric mode is used to accelerate vehicle and the hybrid mode becomes useful to improve the fuel economy.

Similarly when the same design problem is solved for 2-PG multi-mode architectures, the three architectures we converge have the fuel economy of 46.57 mpg, 48.35 mpg, 40.50 mpg. The best design is obtained by starting from the design in Figure 6.9(b) and depicted in Figure 6.13. As it is seen in the figure, the hybrid mode is the same as the initial hybrid mode selected. In this problem, the performance constraints prevent the method to modify the initial design and converges to the same

Table 6.5: Simulation results of the selected designs for Case Study 2

Given $\mathbf{x}_s$	Optimal Design	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2.24$ $FR = 2.16$	Figure 6.12	48.35 mpg	139 mph	7.67 sec
$\rho_1 = 2.24$ $\rho_2 = 2.24$ $FR = 2.16$	Figure 6.13	48.35 mpg	139 mph	7.67 sec

hybrid mode with an additional pure EV mode. This design is the 2-PG equivalent of the design shown in Figure 6.12. The performance results for this design is given in Table 6.5.

As it was pointed in the previous section, these results can be improved by testing more initial points.

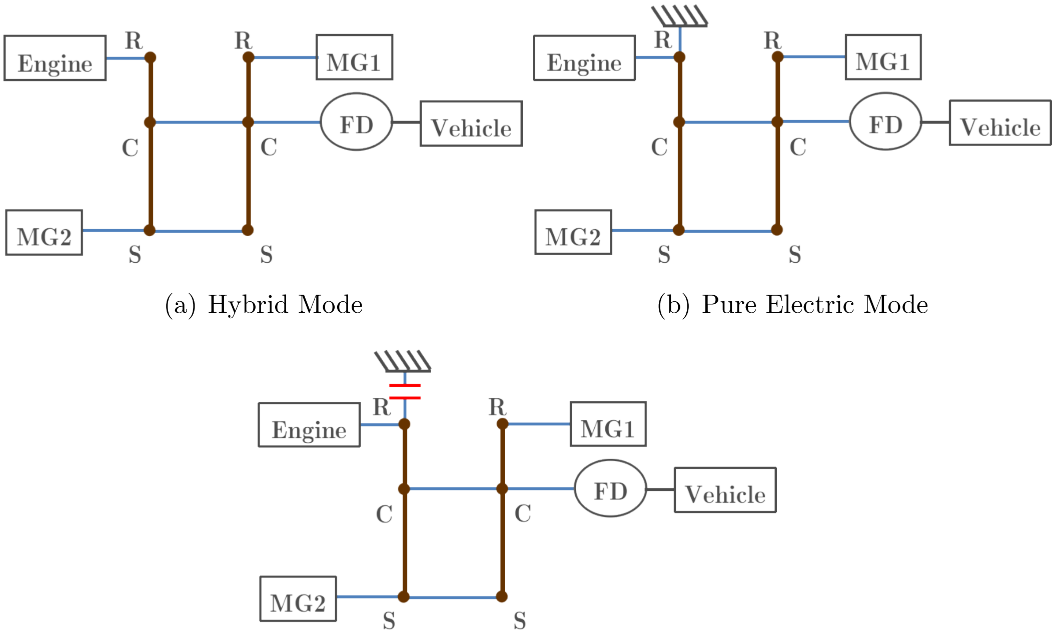
6.7 Case Study 3

The specifications of the vehicle used for this case study is given on Table 6.6.

Table 6.6: Vehicle specifications used for Case Study 3

Specification	Value
Vehicle Body Mass	$2400[kg]$
Tire Radius	$0.4[m]$
Aerodynamic Drag Coefficient	0.45
Frontal Area	$3.2[m^2]$
Battery Voltage	$350[V]$
Battery Efficiency	$92[\%]$
Battery Capacity	$13.8[Ah]$
Rated MG1 / MG2 Power	$75[kW]$
Max MG1 / MG2 Torque	$300[Nm]$
Rated Engine Power	$140[kW]$
Max Engine Torque	470 [Nm]
Engine Displacement Size	$6.5[L]$

Following the same methodology used for Case Study 1 and 2, when we solve



(a) Hybrid Mode

(b) Pure Electric Mode

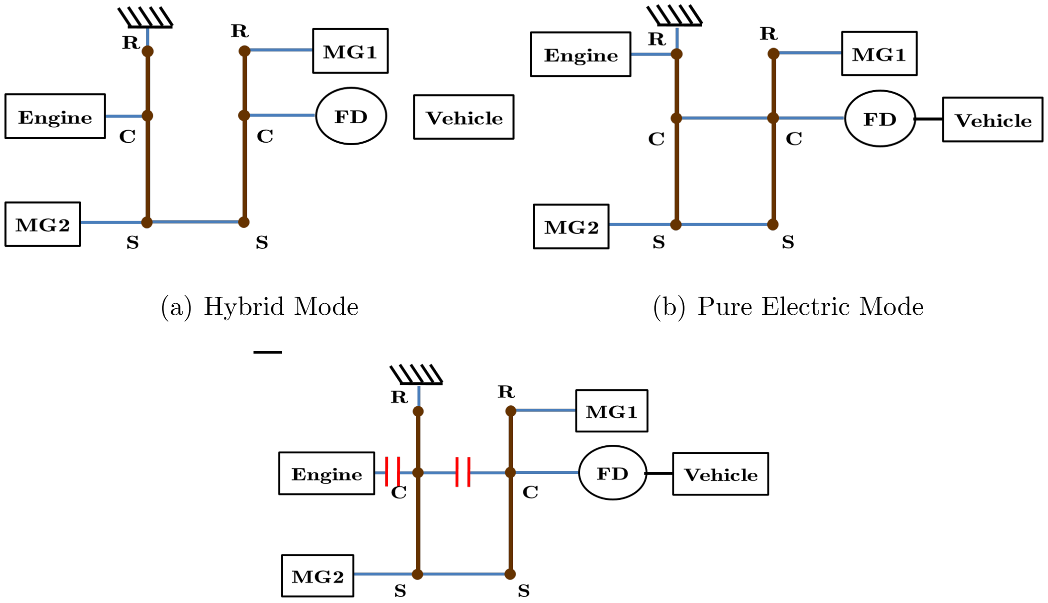
(c) Combined Architecture

Figure 6.13: Best 2-PG architecture found for Case Study 2

the problem for 1-PG configurations starting from the same initial points as before, only the first test starting from the configuration shown in Figure 6.9(a) converges to an architecture satisfying the performance constraints. This architecture with fuel economy of 34.2 mpg is shown in Figure 6.10. Note that this is the same design we found for Case Study 1. Recall from Chapter V that the same case study had only two feasible hybrid configurations. So difficulty in finding feasible designs in this case study was an expected outcome.

Table 6.7: Simulation results of the selected designs for Case Study 3

Given $\mathbf{x}_s$	Optimal Design	Fuel Consumption	Top Speed	0-60 mph time
$\rho = 2$ $FR = 5$	Figure 6.10	34.2 mpg	130 mph	7.5 sec
$\rho_1 = 2$ $\rho_2 = 2$ $FR = 5$	Figure 6.14	35.1 mpg	111 mph	11.52 sec



(a) Hybrid Mode

(b) Pure Electric Mode

(c) Combined Architecture

Figure 6.14: Best 2-PG architecture found for Case Study 3

When we solve the same problem for 2-PG configurations where we have more alternative configurations, three architectures we find starting from the initial configurations given in Figure 6.9, have fuel economies of 34.2 mpg, 35.1 mpg, 33.5 mpg. The best design is given in Figure 6.14. The results for the best 1-PG and 2-PG architecture designs are also shown in Table 6.7.

The main challenge with this type architecture design problems is the combinatorial nature. In order to provide more efficient solutions to this type of problems, a continuous representation of the problem needs to be found. Chapter VII includes the gear ratio design to the problem and proposes a continuous problem formulation that can be addressed by the available optimization algorithms in a more efficient way.

6.8 Summary

In this chapter, we discussed the design of multi-mode HEV architectures with given component sizes. We created multi-mode architectures by combining two or more modes using the configurations we generated in Chapter III. When combining modes, in order to prevent the use of a large number of clutches we defined an architecture complexity design characteristic measured by the minimum number of clutches needed to combine the modes. We showed how to calculate this complexity measure with examples. In the design process we constrained the complexity to filter the designs with many clutches. As opposed to the single-mode design, the number of architecture alternatives was large enough to make the enumeration computationally intractable. The results from the search method we introduced highly depends on the initial mode selection and it needs to be tested with many initial designs in order to find the global optimum. However, if the purpose is to improve a given design the method was shown to be useful.

We showed case studies designing dual-mode architectures. However, the methods described in Sections 6.3 and 6.2 are general enough to account for larger number of modes as well.

The same comment about fixed component sizing that we made for single-mode architecture design can be made here, too. There is some more potential to improve the results by making configuration and component sizing design decisions simultaneously. However, such an approach requires a new methodology. Chapter VII gives a mathematical formulation and proposes a partitioning and coordination strategy to solve this combined problem.

CHAPTER VII

Simultaneous Architecture and Sizing Design

In Chapters V and VI, the HEV architecture design problems for fixed gear ratios were introduced and solution strategies were provided using the representation and generation framework proposed in Chapter III. In this chapter we extend the problem to the case where component design, specifically gear ratios, are also part of the design problem definition. We formulate the combined configuration and component design problem and propose a decomposition-based solution approach for this combined problem.

When there are no tight constraints on the values of gear ratios and no good values are known for them to treat gear ratios as parameters, the gear ratios must be treated as decisions (design variables) in the problem formulation. Solving the configuration and gear ratio design problems separately as proposed by *Liu* (2007) ignores the coupling between them. Gear ratio selection and configuration decisions are coupled and have significant impact on the fuel economy. Solving the gear ratio design as a nested problem within the configuration design (for example, through a parametric optimization for each configuration) is only possible when the number of configurations is small such as 1-PG configurations. When the designs of interest are 2-PG and 3-PG systems, a new methodology to handle configuration and gear ratio design simultaneously is necessary.

7.1 Problem Formulation

In Chapter IV we provided an overview of the general HEV powertrain architecture design problem under different scenarios. A simplified problem when the component sizes and gear ratio selections are given was discussed in Chapters V and VI. In this section we revisit the general formulation and propose a solution strategy to the problem combining the configuration and component design decisions denoted by $\mathbf{x}_c$ and $\mathbf{x}_s$, respectively. The general mathematical formulation can be given as follows:

$$
\begin{aligned}
\text{min} \quad & f_{cons}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \\
\text{w.r.t.} \quad & \{\mathbf{x}_c(N_{mode}),\ \mathbf{x}_s\} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{x}_c(N_{mode}), \mathbf{x}_s, \psi^*(t, \mathbf{x}_c(N_{mode}), \mathbf{x}_s, \mathbf{p})) \leq \mathbf{0} \\
& g_{complex}(\mathbf{x}_c(N_{mode})) \leq \mathbf{0} \\
& lb \leq \mathbf{x}_s \leq ub \\
& N_{mode} \in \{1, 2, 3, 4, ...\} \\
& \mathbf{x}_c \text{ is feasible}
\end{aligned}
\tag{7.1}
$$

where f_{cons} is the fuel consumption, $\mathbf{g}_{perf}$ is the vector of vehicle performance constraints and $g_{complex}$ is the architecture complexity function defined in Chapter VI which applies only to multi-mode architectures. Note that in $\mathbf{x}_s$ only PG ratios $\boldsymbol{\rho}$ and final ratio FR are considered as sizing variables in this study. Further, $\mathbf{x}_c$ is a function of N_{mode} (modes in the architecture). Also $\mathbf{p}$ denotes the vehicle and other powertrain parameters which are assumed to be given a priori. This formulation solves the HEV supervisory control problem in a nested formulation. Therefore ψ^* denotes the optimal control strategy obtained by the nested control problem solution for a given design. A review of available control strategies to compute ψ^* was given Chapter V.

Assume that $\mathbf{x}_c$ is represented by a modified bond graph, i.e., a vector consisting of elements of the adjacency matrix, junction types, causalities and bond weights. One way to approach this problem is to define elements of $\mathbf{x}_c$ as integer design variables together with $\mathbf{x}_s$ as continuous design variables and solve the problem using a derivative-free algorithm. However, not all modified bond graphs correspond to feasible HEV configurations. This is the reason for including the feasibility constraints on $\mathbf{x}_c$ in the formulation given by Equation (7.1). These constraints were described in detail in Chapter III. The feasibility constraints define not only a small design space but also a very dispersed (disjoint) one. In such a case, derivative-free algorithms have difficulty even finding feasible designs, let alone optimizing. Therefore, they cannot be used with this formulation.

Another way to approach the problem given in Equation (7.1) is to enumerate the feasible $\mathbf{x}_c$ and solve the optimization problem with respect to $\mathbf{x}_s$ for all feasible $\mathbf{x}_c$. This approach is possible only when the number of feasible $\mathbf{x}_c$ is small. For instance for 1-PG designs, the number of feasible configurations is 16 and enumeration is possible in this case. However, for 2-PG systems the number of feasible configurations becomes 2244 and even larger for more PGs. Since fuel consumption calculations that require solving a nested control problem are computationally expensive, optimizing each configuration one by one with respect to $\mathbf{x}_s$ is not computationally tractable.

An alternative approach is to use another representation that consists of continuous variables combining both configuration and gear ratios as proposed by *Cheong et al.* (2011). In Chapters III and V the use of a 2 × 2 kinematic matrix $\mathbf{C}_{mode}$ was described in detail. This kinematic matrix used in vehicle simulations depends on both configuration and PG ratios but is independent of the final drive ratio. In order to include the final drive ratio as well, we define the following new matrix:

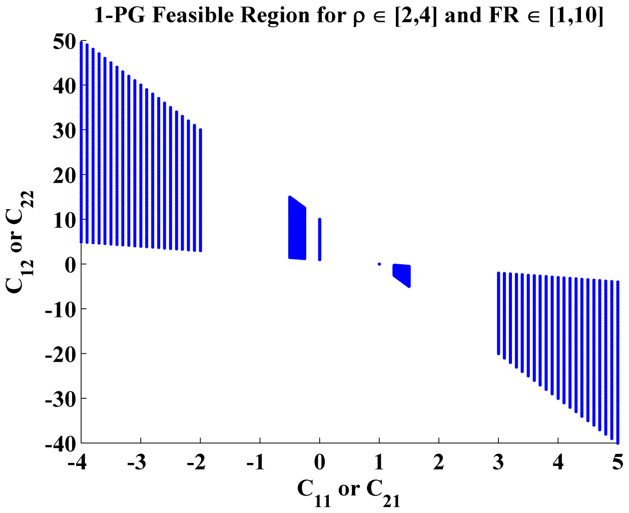
$$\mathbf{C}_{conf}(\mathbf{x}_c, \mathbf{x}_s) = \begin{cases} \mathbf{C}_{mode} \cdot \begin{bmatrix} 1 & 0 \\ 0 & FR \end{bmatrix} & if \text{ hybrid mode} \\ \mathbf{C}_{mode} \cdot 1/FR & if \text{ pure electric mode,} \end{cases} \tag{7.2}$$

Here $\mathbf{C}_{conf}$ is a function of $\mathbf{x}_c$ and $\mathbf{x}_s$ where $\mathbf{x}_s = \begin{bmatrix} \boldsymbol{\rho} \\ FR \end{bmatrix}$. The elements of $\mathbf{C}_{conf}$ could be optimized to maximize fuel economy subject to vehicle performance constraints. However, contrary to the case discussed in the study by *Cheong et al.* (2011) not every $\mathbf{C}_{conf}$ can be realized as a feasible configuration in real design. For instance, Figure 7.1 shows the projections of the 4-dimensional feasible design space of $\mathbf{C}_{conf}$ which can be obtained by 16 1-PG configurations.

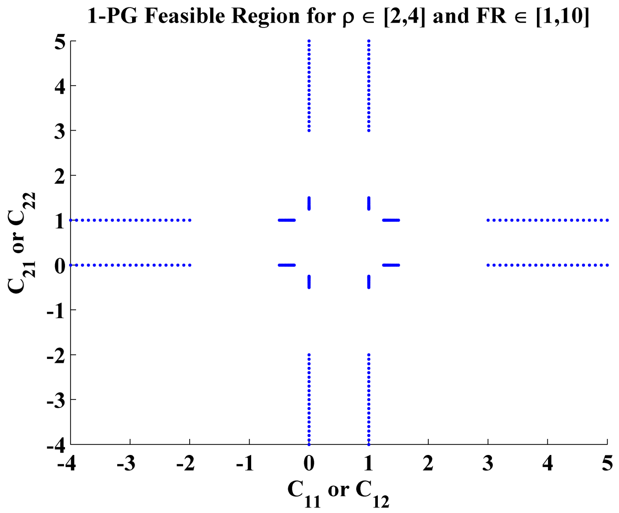
As seen in Figure 7.1, there are many disjoint feasible regions in the design space. When designing this $\mathbf{C}_{conf}$ matrix, such feasibility constraints must be taken into account. Expressing the feasibility constraints on $\mathbf{C}_{conf}$ becomes more complicated as the number of PGs in the system increases. In what follows, we propose a decomposition-based strategy to design $\mathbf{C}_{conf}$ taking these feasibility constraints into account. The next section elaborates on this decomposition-based strategy.

7.2 Decomposition-Based Solution Strategy

Decomposition-based design optimization methods have been developed for problems consisting of several subproblems interacting with each other. Solving the overall problem all together as a single problem is referred to as *All-In-One* (AIO) approach. When AIO is possible, then it should be preferred. However, in large scale problems or problems already decomposed into subproblems each of which can come from different disciplines, a single optimization method may not be capable of solving the



(a) C_{11} vs C_{12} (or C_{21} vs $C22$)



(b) C_{11} vs C_{21} (or C_{12} vs $C22$)

Figure 7.1: Projection of the 4D feasible region to 2D planes

overall problem. In such cases a coordination strategy accounting for the interactions among subsystems is necessary. Partitioning and coordination techniques to for such systems have been studied in Multidisciplinary Design Optimization (MDO) area. *Martins and Lambe* (2013) give a through review of recent available MDO methods.

As an MDO strategy, Analytical Target Cascading (ATC) was introduced by *Kim* (2001) and *Michelena et al.* (2003). ATC translates the system-level targets to the subsystem specifications preserving consistency at the system level. The theoretical convergence of this method was shown by *Michelena et al.* (2003) when all the subproblems are convex. Augmented Lagrangian penalty function was applied by *Tosserams et al.* (2006) to extend its applicability to a larger variety of problems. Available implementations of ATC to problems from several disciplines are given by *Kang et al.* (2012); *Li et al.* (2008); *Allison et al.* (2006); *Choudhary et al.* (2005); *Kokkolaras et al.* (2004); *Blouin et al.* (2004); *Kim et al.* (2003a,b, 2002).

Since an AIO formulation of the problem given in Equation (7.1) is difficult to solve, we propose to decompose the problem into two levels using the ATC framework. The system-level problem optimizes both configuration and sizing using the $\mathbf{C}_{conf}$ as the representation without considering feasibility. The system-level objectives are fuel consumption and (desirable) subsystem responses while the constraints are the vehicle performance requirements. The optimized $\mathbf{C}_{conf}$ values are sent to the subsystem as targets to be met. The subsystem tries to achieve these targets by designing the gear ratios for each feasible configuration and sends to the upper level the $\mathbf{C}_{conf}$ response matching the targets as close as possible. The process continues until subsystem responses system-determined targets match within a tolerance error.

An important point that needs to be noted is that optimizing the gear ratios in each feasible configuration one by one for minimum fuel consumption is computationally expensive and generally not feasible for systems having more than 1 PG. However, same optimization is possible if the objective is to meet the system level $\mathbf{C}_{conf}$ targets because the computational cost of calculating the $\mathbf{C}_{conf}$ given a configuration and gear ratios is much smaller compared to the fuel consumption calculations. This is what makes the decomposition-based approach attractive.

We apply this formulation to a single-mode architecture design and a multi-mode architecture design separately. Sections 7.2.1 and 7.2.2 give the mathematical formulations for the decomposed problems for these two cases.

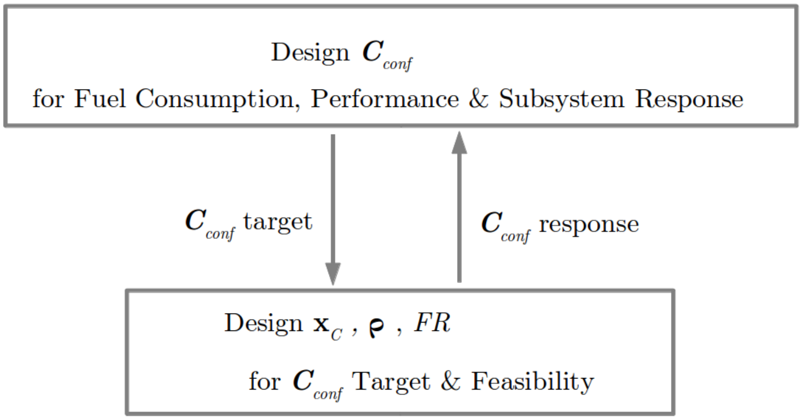

Design $\boldsymbol{C}_{conf}$
for Fuel Consumption, Performance & Subsystem Response
$\boldsymbol{C}_{conf}$ target
$\boldsymbol{C}_{conf}$ response
Design $\mathbf{x}_C$, $\boldsymbol{\rho}$, FR
for $\boldsymbol{C}_{conf}$ Target & Feasibility

Figure 7.2: Decomposition of combined single-mode architecture and gear ratio design

7.2.1 Single-mode HEV Architecture Design Problem

The general problem formulation given in Equation (7.1) is reformulated in two levels for a single mode architecture design using $\mathbf{C}_{conf}$ as the representation of the mode. Figure 7.2 depicts the two levels of the decomposition. The system-level formulation is as follows:

$$\begin{aligned}
\min \quad & f_{cons}(\mathbf{C}^U_{conf}, \psi^*(t, \mathbf{C}^U_{conf}, \mathbf{p})) + \phi(\mathbf{C}^U_{conf} - \mathbf{C}^L_{conf}) \\
\text{w.r.t.} \quad & [C_{11}, C_{12}, C_{21}, C_{22}], \text{ where } \mathbf{C}^U_{conf} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \\
\text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{C}^U_{conf}, \psi^*(t, \mathbf{C}^U_{conf}, \mathbf{p})) \leq \mathbf{0} \\
& lb \leq \mathbf{C}^U_{conf} \leq ub \\
& |det(\mathbf{C}^U_{conf})| > 0
\end{aligned} \tag{7.3}$$

where ϕ is the augmented Lagrangian penalty function (*Tosserams et al.*, 2005). Superscripts $(\cdot)^U$ and $(\cdot)^L$ indicate upper and lower system variables, respectively. In this formulation, the elements of $\mathbf{C}_{conf}$ are continuous design variables and all invertible $\mathbf{C}_{conf}$ are assumed to be feasible. If $\mathbf{C}_{conf}$ is singular, it does not correspond to any 2-dof design. So we include invertibility as constraint at the system level. When an optimal set of design variables is found, they are passed to the subsystem level as targets. The subsystem problem addresses feasibility of $\mathbf{C}_{conf}$ based on the feasible configurations we generated earlier. The subsystem problem is stated as follows:

$$\begin{aligned} \text{min} \quad & \phi(\mathbf{C}^U_{conf} - \mathbf{C}^L_{conf}) \\ \text{w.r.t.} \quad & \mathbf{x} = [\mathbf{x}_c, \mathbf{x}_s], \text{ where } \mathbf{x}_s = [\boldsymbol{\rho}, FR] \\ \text{s.t.} \quad & lb \leq \mathbf{x}_s \leq ub \\ & \mathbf{x}_c \text{ is feasible} \\ \text{where} \quad & \mathbf{C}^L_{conf} = f_{conf}(\mathbf{x}) \end{aligned} \tag{7.4}$$

Here f_{conf} is the function that calculates $\mathbf{C}_{conf}$ given the configuration $\mathbf{x}_c$, PG ratios $\boldsymbol{\rho}$ and final drive ratio FR. This subsystem problem can be optimized with respect to $\mathbf{x}_s$ for each feasible $\mathbf{x}_c$ separately. As discussed earlier, since evaluation of f_{conf} is almost instantaneous, enumeration over all feasible configurations does not have a large computational cost. Once the problem is solved for all feasible configurations, the $\mathbf{C}_{conf}$ that matches the target closest is sent back to the system level.

An important aspect of this formulation is its applicability to any number of PGs without requiring much modification. Any 2-dof hybrid system can be represented by 2×2 $\mathbf{C}_{conf}$ matrix. Therefore, the system-level problem is exactly the same regardless of the number of PGs in the architecture. The subsystem-level problem has a different numbers of feasible $\mathbf{x}_c$ for different number of PGs used in the architecture. The size of $\mathbf{x}_c$ also increases with the number of PGs but the formulation and the solution strategy remain the same. In Section 7.3 two case studies are presented to show the implementation of this idea.

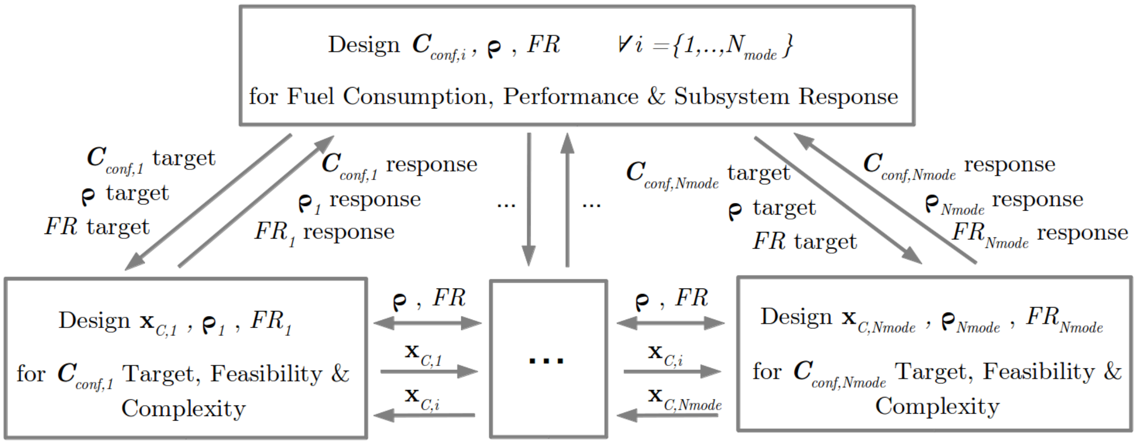

Design $C_{conf,i}$, ρ , FR $\forall i = \{1,..,N_{mode}\}$
for Fuel Consumption, Performance & Subsystem Response
$C_{conf,1}$ target
ρ target
FR target
$C_{conf,1}$ response
ρ_1 response
FR_1 response
...
...
$C_{conf,Nmode}$ target
ρ target
FR target
$C_{conf,Nmode}$ response
ρ_{Nmode} response
FR_{Nmode} response
Design $\mathbf{x}_{C,1}$, ρ_1 , FR_1
for $C_{conf,1}$ Target, Feasibility & Complexity
ρ , FR
$\mathbf{x}_{C,1}$
$\mathbf{x}_{C,i}$
...
ρ , FR
$\mathbf{x}_{C,i}$
$\mathbf{x}_{C,Nmode}$
Design $\mathbf{x}_{C,Nmode}$, ρ_{Nmode} , FR_{Nmode}
for $C_{conf,Nmode}$ Target, Feasibility & Complexity

Figure 7.3: Decomposition of combined multi-mode architecture and gear ratio design

7.2.2 Multi-mode HEV Architecture Design Problem

When there are multiple modes in the architecture, both system level and subsystem level formulations differ from the single-mode architecture design formulation. There are multiple ways to decompose the problem in this case. For brevity, only one formulation is given in this section and alternative approaches are briefly mentioned. One decomposition is given in Figure 7.3. In this situation, there is a single system and as many subsystems as the number of modes in the architecture. If there are N_{mode} modes in the architecture, an N_{mode} number of $\mathbf{C}_{conf}$ matrices must be designed at the system level. Also the PG ratios and final drive ratios cannot be different for different modes in the same architecture. Hence, $\boldsymbol{\rho}$ and FR must be shared variables among the subsystems and they appear as design variables in the system-level formulation.

In the muti-mode architecture design, there is also an additional complexity constraint which was described in Chapter VI. This constraint that depends only on $\mathbf{x}_c$ is taken into account in the subsystem where we design each individual configuration.

The formulation at the system level is given as follows:

$$\begin{aligned} \min \quad & f_{cons}(\mathbf{C}^U_{conf,i}, \psi^*(t, \mathbf{C}^U_{conf,i}, \mathbf{p})) + \sum_{i=1}^{N_{mode}} \phi(\mathbf{C}^U_{conf,i} - \mathbf{C}^L_{conf,i}) + \phi(\mathbf{x}^U_s - \mathbf{x}^L_{s,i}) \\ \text{w.r.t.} \quad & [C_{11,i}, C_{12,i}, C_{21,i}, C_{22,i}, \mathbf{x}^U_s], \text{ where } \mathbf{C}^U_{conf,i} = \begin{bmatrix} C_{11,i} & C_{12,i} \\ C_{21,i} & C_{22,i} \end{bmatrix} \ \forall i \in \{1, ...N_{mode}\} \\ \text{s.t.} \quad & \mathbf{g}_{perf}(\mathbf{C}^U_{conf,i}, \psi^*(t, \mathbf{C}^U_{conf,i}, \mathbf{p})) \leq \mathbf{0} \\ & lb_i \leq \mathbf{C}^U_{conf,i} \leq ub_i \\ & |det(\mathbf{C}^U_{conf,i})| > 0_i \end{aligned} \tag{7.5}$$

where the subscript $(\cdot)_{\cdot,i}$ denotes each individual mode in the architecture. Each $\mathbf{C}_{conf,i}$ optimized at the system level is sent as target to the corresponding subsystems. Also the same $\mathbf{x}_s$ is sent to every subsystem as it is the vector of shared variables. The formulation for a subsystem i becomes:

$$\begin{aligned} \min \quad & \phi(\mathbf{C}^U_{conf,i} - \mathbf{C}^L_{conf,i}) + \phi(\mathbf{x}^U_s - \mathbf{x}^L_{s,i}) \\ \text{w.r.t.} \quad & \mathbf{x}_i = [\mathbf{x}_{c,i}, \mathbf{x}^L_{s,i}], \text{ where } \mathbf{x}^L_{s,i} = [\boldsymbol{\rho}_i, FR_i] \\ \text{s.t.} \quad & g_{comp}(\mathbf{x}_{c,i}, \mathbf{x}_{c,j}) \leq 0 \quad \forall j \in \{1, ...N_{mode} | j \neq i\} \\ & lb_i \leq \mathbf{x}^L_{s,i} \leq ub_i \\ & \mathbf{x}_{c,i} \text{ is feasible} \\ \text{where} \quad & \mathbf{C}^L_{conf,i} = f_{conf}(\mathbf{x}_i) \end{aligned} \tag{7.6}$$

Recall that we use the number of clutches in the architecture as a measure of complexity. Here g_{comp} calculates the number of clutches necessary to combine two modes. Instead of constraining the total number of clutches in the architecture, we can restrict the clutches needed to shift from one mode to another. Such a constraint requires only $\mathbf{x}_c$ from the next subsystem, and the order of the subsystems becomes important. As mentioned earlier, there are multiple ways to define the design problem for multi-mode architectures. The total number of clutches can also be constrained, but in that case each individual subsystem requires $\mathbf{x}_c$ from all subsystems.

An alternative formulation can define all subsystems as a single subsystem. In that case the system-level problem does not include the shared variables as there is only one subsystem. The design variables become only the elements of $\mathbf{C}_{conf,i}$ for all modes in the architecture. Only the optimized $\mathbf{C}_{conf,i}$ matrices are passed to the subsystem. The subsystem tries to meet all the targets at once by designing $\mathbf{x}_{c,i}$, $\boldsymbol{\rho}$ and FR. There are pros and cons of this approach compared to the one described earlier. In the former approach the problem at the subsystem level is simpler but the coordination of the shared variables make the overall ATC problem challenging. In the latter approach, the overall ATC problem is very similar to the single mode-architecture design problem but the subsystem level problem is much bigger as the number of feasible configurations increases exponentially with the number of modes. A more detailed analysis of the alternative formulations is left as future work.

7.3 Results

In this section we present two case studies to design a single-mode architecture with 1 and 2 PGs. Both case studies use the same vehicle and powertrain components, similar to the Toyota Prius. Key specifications for the vehicle and powertrain components are given in Table 7.1.

In both case studies fuel consumption is calculated as the average of city (UDDS) and highway (HWFET) cycle consumptions. The HEV supervisory control problem is solved with a nested formulation using ECMS as described in Chapter V. In both case studies, 0-60 mph acceleration time, t_{60mph} and top speed, V_{top} are used as performance requirements. While 0-60 mph time is related to the maximum torque capability of the architecture, top speed is set by the maximum rotary speed of the powertrain components.

Table 7.1: Vehicle specifications used for the case studies

Specification	Value
Vehicle Body Mass	$1400[kg]$
Tire Radius	$0.3[m]$
Aerodynamic Drag Coefficient	0.29
Frontal Area	$2[m^2]$
Battery Voltage	$350[V]$
Battery Efficiency	$92[\%]$
Battery Capacity	$6.5[Ah]$
Rated MG1 Power	$42[kW]$
Rated MG2 Power	$60[kW]$
Max MG Speed	12000 [rpm]
Max MG Torque	200 [Nm]
Rated Engine Power	$43[kW]$
Max Engine Torque	$102[Nm]$
Engine Displacement Size	$1.5[L]$

In the case studies, the top speed requirement is set to be at least 105 miles per hour while 0-60 mph time is required to be less than 12 seconds. The PG ratios are allowed to vary between 2 and 4 and the final drive ratio between 1 and 10.

7.3.1 Single-mode 1-PG Architecture Design

For 1-PG systems, since there are only 16 feasible hybrid configurations, we can optimize each configuration with respect to PG ratios and final ratio to maximize the fuel economy subject to vehicle performance constraints. Recall that such an approach is not computationally feasible for a larger number of PGs. We use Sequential Quadratic Programming (SQP) with multiple initial points to solve this problem. After solving the problem with four different initial points, among 16 configurations, only six of them converged to feasible design solutions, i.e. satisfied the vehicle performance constraints. We select the feasible configuration with the best fuel consumption as the solution. All of four tests converged to the same configuration with similar design variable values. Table 7.2 shows the optimal solutions with their

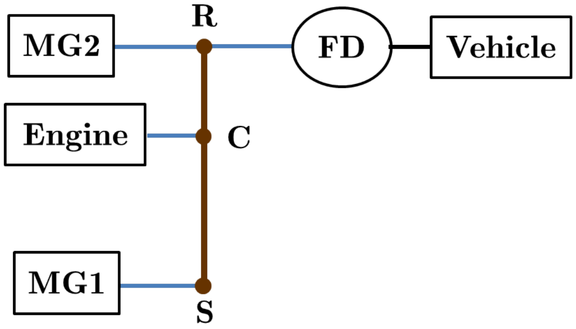


Figure 7.4: Optimal 1-PG architecture obtained by ATC. This is the architecture used in the Toyota Prius

corresponding fuel consumption and performance values. The optimal configuration is the same as the Toyota Prius power-split configuration shown in Figure 7.4.

Table 7.2: Results for for single-mode 1-PG architecture design by optimizing each configuration separately

Initial $\mathbf{x}_s$	Optimal $\mathbf{x}_s$	Optimal $\mathbf{C}_{conf}$	Fuel Consumption	Top Speed	0-60 mph time
$\rho_0 = 2$ $FR_0 = 1.5$	$\rho^* = 2.01$ $FR^* = 4.27$	$\begin{bmatrix} 3.01 & -8.59 \\ 0 & 4.27 \end{bmatrix}$	56.81 mpg	108 mph	11.66 sec
$\rho_0 = 2.5$ $FR_0 = 2.5$	$\rho^* = 2.00$ $FR^* = 4.25$	$\begin{bmatrix} 3.00 & -8.50 \\ 0 & 4.29 \end{bmatrix}$	56.80 mpg	108 mph	11.74 sec
$\rho_0 = 2.5$ $FR_0 = 4.5$	$\rho^* = 2.00$ $FR^* = 4.15$	$\begin{bmatrix} 3.00 & -8.30 \\ 0 & 4.15 \end{bmatrix}$	56.77 mpg	107 mph	11.99 sec
$\rho_0 = 3$ $FR_0 = 3.5$	$\rho^* = 2.00$ $FR^* = 4.29$	$\begin{bmatrix} 3.00 & -8.57 \\ 0 & 4.29 \end{bmatrix}$	56.87 mpg	108 mph	11.64 sec

The results shown on Table 7.2 are very similar to each other. Some of them hit the lower bound of the PG ratio which makes this bound an active constraint. Although there is some potential to improve the fuel consumption results by relaxing the bound, we are limited by the practicality constraints on the PG designs. The smallest pinion gear size limits the PG ratio to 2 (*Antony and Pantelides*, 2006). We take these optimal results as a reference point in order to evaluate the solutions obtained by ATC.

ATC formulation allows us to use different solution algorithms for the system- and subsystem-level problems. This flexibility offers the opportunity to test multiple optimization solvers and find the suitable one for each problem. The combined configuration and sizing problem at the system level is highly non-linear. On the contrary, the subsystem-level problem for each configuration can be expressed analytically by extracting the $\mathbf{C}_{conf}$ matrices from the modified bond graph representation. The subsystem-level problem can be successfully solved using SQP. For the system-level problem we tested SQP, interior point method, and GA. The results of these tests are given in Tables 7.3, 7.4 and 7.5, respectively.

Using the bounds on ρ and FR we can calculate some reasonable upper and lower bounds on the system-level design variables as it can be seen on Figure 7.1.

Table 7.3: ATC results for single-mode 1-PG architecture design using SQP for the system level problem

Initial $\mathbf{C}_{conf}$	Optimal $\mathbf{C}_{conf}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 3.9 & -12.2 \\ 0 & 4.2 \end{bmatrix}$	$\begin{bmatrix} 3.51 & -11.07 \\ 0 & 4.40 \end{bmatrix}$	$\rho^* = 2.51$ $FR^* = 4.4$	56.40 mpg	111 mph	10.67 sec
$\begin{bmatrix} 0 & 4 \\ -0.4 & 5.6 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 0 \\ -0.25 & 4.08 \end{bmatrix}$	$\rho^* = 4$ $FR^* = 3.26$	48.7 mpg	**94 mph**	**16.38 sec**
$\begin{bmatrix} 0 & 3 \\ -2.2 & 9.6 \end{bmatrix}$	$\begin{bmatrix} 0 & 2.33 \\ -2 & 7.00 \end{bmatrix}$	$\rho^* = 2$ $FR^* = 2.33$	48.36 mpg	111 mph	**15.78 sec**
$\begin{bmatrix} -0.5 & 5.7 \\ 0 & 3.8 \end{bmatrix}$	$\begin{bmatrix} -0.25 & 7.82 \\ 0 & 6.26 \end{bmatrix}$	$\rho^* = 4$ $FR^* = 6.26$	**0 mpg**	**80 mph**	6.71 sec
$\begin{bmatrix} -2.2 & 9.6 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} -0.35 & 4.63 \\ 0 & 3.41 \end{bmatrix}$	$\rho^* = 2.79$ $FR^* = 3.41$	**0 mpg**	109 mph	8.68 sec

From Table 7.3, the differences between initial and optimal $\mathbf{C}_{conf}$ values are small. Also 3 out of 5 cases converge to infeasible designs in terms of vehicle performance and one case satisfying the vehicle performance requirements cannot recharge the battery at the initial SOC level while following the drive cycle. Violated constraints are highlighted in bold in the table. Also the designs that cannot follow the drive cycle or cannot sustain the battery charge at the end of the cycle are assigned to 0 mph.

Table 7.4: ATC results for single-mode 1-PG architecture design using interior point method for the system level problem

Initial $\mathbf{C}_{conf}$	Optimal $\mathbf{C}_{conf}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 3.9 & -12.2 \\ 0 & 4.2 \end{bmatrix}$	$\begin{bmatrix} 3.48 & -10.53 \\ 0 & 4.23 \end{bmatrix}$	$\rho^* = 2.48$ $FR^* = 4.23$	56.53 mpg	110 mph	11.08 sec
$\begin{bmatrix} 0 & 4 \\ -0.4 & 5.6 \end{bmatrix}$	$\begin{bmatrix} -0.29 & 5.32 \\ 0 & 4.13 \end{bmatrix}$	$\rho^* = 3.45$ $FR^* = 4.12$	**0 mpg**	**101 mph**	7.7 sec
$\begin{bmatrix} 0 & 3 \\ -2.2 & 9.6 \end{bmatrix}$	$\begin{bmatrix} 0 & 3.07 \\ -2.06 & 9.38 \end{bmatrix}$	$\rho^* = 2.06$ $FR^* = 3.07$	51.37 mpg	123 mph	11.92 sec
$\begin{bmatrix} -0.5 & 5.7 \\ 0 & 3.8 \end{bmatrix}$	$\begin{bmatrix} -0.49 & 6.02 \\ 0 & 4.04 \end{bmatrix}$	$\rho^* = 2.04$ $FR^* = 4.04$	33.97 mpg	105 mph	7.63 sec
$\begin{bmatrix} -2.2 & 9.6 \\ 0 & 3 \end{bmatrix}$	$\begin{bmatrix} -2 & 9.63 \\ 0 & 3.21 \end{bmatrix}$	$\rho^* = 2$ $FR^* = 3.2$	51.37 mpg	120 mph	11.51 sec

We further tested another gradient-based method, interior point (*Byrd et al.*, 1999, 2000), using the same initial points to solve the system level problem. The results for this test are shown on Table 7.4. Similar to the results with SQP, the solution changes the changes the initial designs by a small amount. These methods give results similar to the ones we obtained by optimizing each configuration separately only when we start from a very similar initial point. In order to ensure the reliability of the results we need to test these methods with several initial points.

An alternative idea is to use a method which utilizes multiple initial points such as GA. If we don't provide any initial designs, GA randomly generates an initial population. However, this approach is not useful since the random population will mostly have infeasible designs. A better approach is to generate some feasible designs to provide as initial population to GA. A variety in the initial population help us to make the results more reliable. Since we have only 16 configuration possibilities, we can generate some feasible designs from each configuration and randomly select some feasible designs from each configuration to provide some variety to the initial population. As noted earlier, evaluating the 0-60mph time and the top speed of a given design is much faster than evaluating the fuel consumption over a drive cycle since these performance measures do not require solving a nested optimal control problem. This way we can find feasible designs from only 6 configurations out of 16 that satisfy the vehicle performance requirements. Randomly selecting 100 designs in total from each of 6 configurations and setting the generation number to 5 gives the results shown in Table 7.5.

Table 7.5: ATC results for single-mode 1-PG architecture design using GA with 5 generations for the system level problem

Optimal $\mathbf{C}_{conf}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 3.67 & -11.46 \\ 0 & 4.28 \end{bmatrix}$	$\rho^* = 2.67$ $FR^* = 4.28$	56.37 mpg	111 mph	10.73 sec
$\begin{bmatrix} 3.84 & -12.00 \\ 0 & 4.21 \end{bmatrix}$	$\rho^* = 2.85$ $FR^* = 4.21$	56.24 mpg	111 mph	11.98 sec
$\begin{bmatrix} 4.07 & -11.58 \\ 0 & 3.77 \end{bmatrix}$	$\rho^* = 3.07$ $FR^* = 3.77$	56.35 mpg	109 mph	11.57 sec

The configurations for these results are all the same as the one shown in Figure 7.4. The results from using this specially-seeded GA for the system-level problem are more consistent and close to the results obtained from optimizing each configuration one by one. Both approaches converge to the same configuration with some difference in the gear ratios. After the ATC process, we can use a local search on the gear ratios using the final configuration to fine tune the results. This will decrease the difference between the gear ratios obtained from both approaches.

The consistency in ATC comes mainly from the variety in the initial population.

As long as the initial population contains designs from different configurations, GA is expected to find similar results. In a 1-PG system, since the number of feasible configurations is small, it is not challenging to select the initial population.

In the next section, another case study where we have more than 100 configurations containing feasible designs is discussed.

7.3.2 Single-mode 2-PG Architecture Design

In the previous section we described how SQP, interior point and a specially-designed GA perform to solve the system-level problem. When we increase the number of PGs in the architecture, the system-level problem does not change but the subsystem problem has more design variables and more configurations to search. In a 2-PG architecture design, we select the GA approach again, as it provides the most reliable results. Unfortunately, in this case study it is not possible to optimize each configuration one by one to maximize the fuel economy with performance constraints because of the computational cost. However, we can use the lessons learned from the 1-PG case study to increase the chance of finding good designs.

We already discussed that we need to provide a variety of designs in the initial population of GA. In the case of 2-PG configurations, we have 2124 possible choices and more than 100 of them have feasible designs. In this case we can group the designs based on the sign of the elements of $\mathbf{C}_{conf}$ matrix. Since this matrix has four elements and each element can be either positive, negative or zero, we can have 81 possible types of $\mathbf{C}_{conf}$ matrices. Regardless of the number of PGs in the system, the number of types of the elements cannot be more than 81. In addition, not all 81 groups have feasible designs in them because most of these groups will correspond to non-invertible matrices. For instance, a group which has both C_{11} and C_{12} being zero is always non-invertible and cannot correspond to meaningful designs.

For this specific case study, only 15 groups have feasible designs. By selecting

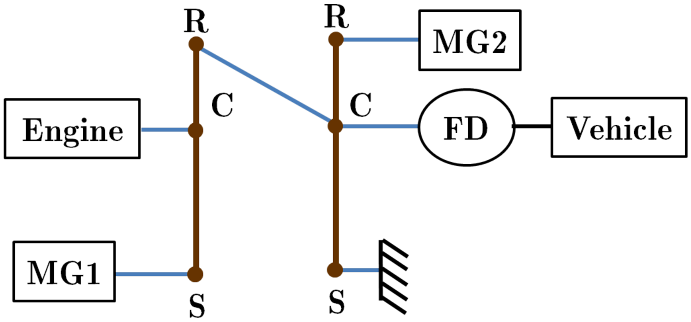


Figure 7.5: Optimal 2-PG architecture obtained by ATC

random 100 designs in total from each of these groups we tested 3 initial populations with 5 generations. The results are shown in Table 7.6. As it is seen in the table, the results consistently converged to similar optimal designs. The configuration that realizes these $\mathbf{C}_{conf}$ is shown in Figure 7.5.

Table 7.6: ATC results for single-mode 2-PG architecture design using GA with 5 generations for the system level problem

Optimal $\mathbf{C}_{conf}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 4.33 & -11.48 \\ 0 & 4.62 \end{bmatrix}$	$\rho_1^* = 3.33$ $\rho_2^* = 2.94$ $FR^* = 3.44$	57.56 mpg	105 mph	11.44 sec
$\begin{bmatrix} 4.56 & -11.94 \\ 0 & 4.5 \end{bmatrix}$	$\rho_1^* = 3.55$ $\rho_2^* = 2.94$ $FR^* = 3.35$	57.51 mpg	105 mph	11.52 sec
$\begin{bmatrix} 3.98 & -11.28 \\ 0 & 5.33 \end{bmatrix}$	$\rho_1^* = 2.45$ $\rho_2^* = 2.98$ $FR^* = 3.78$	57.81 mpg	105 mph	10.65 sec

In order to see the effect of the generation number on the results, we increase it to 10. Table 7.7 shows the results with the same initial populations to the previous test. By comparing Table 7.6 and Table 7.7 we can see that the effect of the generation number on the optimal fuel consumption is small.

Table 7.7: ATC results for single-mode 2-PG architecture design using GA with 10 generations for the system level problem

Optimal $\mathbf{C}_{conf}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 4.29 & -11.38 \\ 0 & 4.73 \end{bmatrix}$	$\rho_1^* = 2.71$ $\rho_2^* = 3.29$ $FR^* = 3.46$	57.69 mpg	105 mph	11.31 sec
$\begin{bmatrix} 4.27 & -11.54 \\ 0 & 4.93 \end{bmatrix}$	$\rho_1^* = 3.27$ $\rho_2^* = 2.53$ $FR^* = 3.53$	57.75 mpg	105 mph	11.01 sec
$\begin{bmatrix} 4.35 & -11.67 \\ 0 & 4.84 \end{bmatrix}$	$\rho_1^* = 2.55$ $\rho_2^* = 3.36$ $FR^* = 3.47$	57.74 mpg	105 mph	11.11 sec

7.3.3 Multi-mode 1-PG Architecture Design

In this section, we demonstrate the decomposition-based design framework for a dual-mode architecture design. The general problem formulation given in Equations (7.5) and (7.5) is solved for 1-PG designs using the same solution strategy described in Sections 7.3.1 and 7.3.2. System-level problem is solved using GA with a special seeding of feasible configurations and subsystem-level problems are solved using SQP. The optimal dual-mode architecture obtained in this case is shown in Figure 7.6. Note that this is the same architecture obtained for Case Study 1 in Chapter VI. Also note that this dual-mode architecture is obtained by adding a pure-electric mode to the optimal architecture we found for single-mode 1-PG design example. The detailed results for this case with three different initial populations are given in Table 7.8.

Similar to the single-mode design cases, the results from three initial populations are similar to each other.

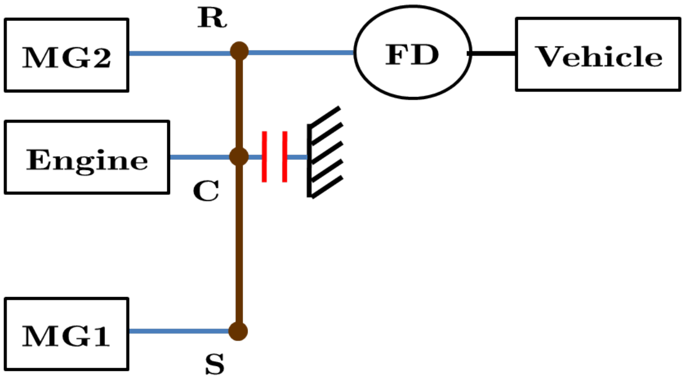


Figure 7.6: Optimal dual-mode 1-PG architecture obtained by ATC

Table 7.8: ATC results for dual-mode 1-PG architecture design using GA with 5 generations for the system level problem

Optimal $\mathbf{C}_{conf,1}$	Optimal $\mathbf{C}_{conf,2}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} 3.00 & -9.37 \\ 0 & 4.68 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.21 \\ 0.21 & 0.43 \end{bmatrix}$	$\rho^* = 2.00$ $FR^* = 4.68$	58.46 [mpg]	110 [mph]	7.09 [sec]
$\begin{bmatrix} 3.08 & -10.25 \\ 0 & 4.92 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.20 \\ 0.20 & 0.41 \end{bmatrix}$	$\rho^* = 2.08$ $FR^* = 4.93$	59.97 [mpg]	112 [mph]	7.04 [sec]
$\begin{bmatrix} 3.03 & -9.60 \\ 0 & 4.72 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.21 \\ 0.21 & 0.43 \end{bmatrix}$	$\rho^* = 2.03$ $FR^* = 4.72$	58.83 [mpg]	111 [mph]	7.11 [sec]

7.3.4 Multi-mode 2-PG Architecture Design

Similar to the previous section, here, we design a dual-mode architecture for the same vehicle application using 2-PG configurations. A similar special seeding strategy for GA is followed in this example with a small difference. If we follow the same $\mathbf{C}_{conf}$ grouping strategy to select feasible designs for initial population, we have 81 × 81 groups for dual-mode architectures. In this case, 100 initial population is not enough to ensure variety. We run the three optimization studies with 300 initial population and the results are given in Table 7.9.

As seen in the table, the first two runs converged to the same architecture while the third run converged to an infeasible design with a different architecture. The

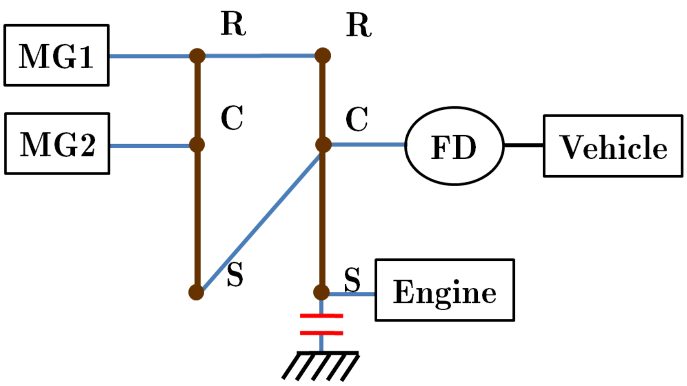


Figure 7.7: Optimal dual-mode 2-PG architecture obtained by ATC

Table 7.9: ATC results for dual-mode 2-PG architecture design using GA with 5 generations for the system level problem

Optimal $\mathbf{C}_{conf,1}$	Optimal $\mathbf{C}_{conf,2}$	Optimal $\mathbf{x}_s$	Fuel Consumption	Top Speed	0-60 mph time
$\begin{bmatrix} -0.50 & 5.63 \\ -0.34 & 5.04 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.20 \\ 0.26 & -0.30 \end{bmatrix}$	$\rho_1^* = 2.17$ $\rho_2^* = 2.00$ $FR^* = 3.75$	75.22 [mpg]	108 [mph]	7.12 [sec]
$\begin{bmatrix} -0.60 & 5.91 \\ -0.40 & 5.16 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.18 \\ 0.27 & -0.31 \end{bmatrix}$	$\rho_1^* = 2.48$ $\rho_2^* = 2.00$ $FR^* = 3.68$	73.79 [mpg]	106 [mph]	6.87 [sec]
$\begin{bmatrix} -2.79 & 6.77 \\ 7.40 & -11.44 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.09 \\ 0.56 & 0.33 \end{bmatrix}$	$\rho_1^* = 2.65$ $\rho_2 * = 2.79$ $FR^* = 1.78$	56.38 [mpg]	117 [mph]	**27.81 [sec]**

best architecture obtained in the first two runs is shown in Figure 7.7. Optimality of these results are questionable due to the dependency on the initial population. A discussion on these results are given in Section 7.3.5.

7.3.5 Discussion

In general, since we use GA to solve the system-level problem, we rely on some random processes, and so every time we run the optimization we get slightly different results. However, the difference among different tests in both 1-PG and 2-PG architecture design cases is only in the gear ratios while the converged configuration remains the same. As mentioned earlier, after obtaining the configuration and gear ratio results from ATC, we can use a local search method to obtain more precise gear ratios to further improve the results.

Comparing the best results of 1-PG and 2-PG architecture optimization from Tables 7.5 and 7.7, the similarity between the optimal $\mathbf{C}_{conf}$ values should be noted. Since the system level problems are the same in both cases, the true system-level optimal without considering feasibility must be the same. The subsystems try to meet this system-level optimal $\mathbf{C}_{conf}$ targets during the iterations of ATC. 1-PG and 2-PG subsystems have different responses but the goal is to come close to the system-level optimal as close as possible. This is the underlying reason to have similar $\mathbf{C}_{conf}$ results in the end. Also in this particular case, there is 2.55% improvement in fuel consumption by going from 1 PG to 2 PGs. It is expected to improve the fuel consumption further by increasing the number of PGs in the architecture as more various $\mathbf{C}_{conf}$ become feasible with more PGs. However, additional PGs increase the production cost as well as the space required for the HEV powertrain. Depending the cost constraints and available space, a choice on 1-PG or 2-PG design might be preferable.

Comparing Tables 7.5 and 7.8 we can see that adding a pure-electric mode to the single-mode architecture improves, both fuel economy and 0-60 mph acceleration performance. Also, the optimal gear ratio values are different for single-mode and multi-mode architectures. It shows the necessity to design component sizes for single-mode and multi-mode architectures separately. Using the gear ratios of a single-mode architecture in a multi-mode architecture even for the same vehicle application is not optimal as seen in this example.

The results for the dual-mode 2-PG architecture design show that this case is more sensitive to the initial population selection. Considering the number of possible architecture candidates for this case, it is expected outcome. Also one can argue that some of the good designs might not be included in any of the initial populations, resulting in only local optimal results. The dependency of the results on the initial population can be reduced by increasing the initial population size and generation number further. That way more reliable results and possibly better architectures can be obtained in the design process. These results also give some idea of the challenges for designing architectures with higher number of modes. Such analyses are left to a future study.

7.4 Summary

In this chapter we discussed the solution of the combined configuration and sizing problem where only gear ratios are considered as sizing variables. We proposed a decomposition-based approach using ATC. The main reason to decompose the problem is the difficulty in solving the all-in-one formulation. Separate decomposition strategies were given for the single-mode and multi-mode architecture design problems. The single-mode and multi-mode architecture design problems were solved using ATC for case studies with a different number of PGs in order to show the versatility of the formulation.

In the case studies, results using only GA for the system level problem and SQP for the subsystem-level problem were presented. While the subsystem problem is simple enough to be solved by SQP, the system-level problem is challenging due to high non-linearity. A better approach to solve the system-level problem can be a combination of global and local search methods.

Although only gear ratios were included as sizing variables, the sizing of the powertrain components can also be included in the system level problem as long as parametric models for the powertrain components are available. Sizing variables of the powertrain components do not affect the subsystem formulation because the subsystem checks the feasibility of the configuration only. As discussed in Section 7.3, the system-level problem is highly non-linear in its present form. Addition of the powertrain component sizing to the problem will pose further solution challenges.

CHAPTER VIII

Conclusions

8.1 Summary

Planetary gear systems have been widely used in the design of HEV architectures since they provide a variety of speed ratios that increases the overall efficiency of the powertrain without requiring an additional transmission. These systems also provide many architecture alternatives to select from. The architecture design alternatives include single-mode architectures that are in a fixed configuration, and multi-mode architectures that changes the configuration during the vehicle operation. Multi-mode architectures provide additional efficiency and performance compared to the single-mode architecture designs since they include multiple configurations each of which performs well under different driving conditions. Multi-mode architectures also introduce additional complexity and cost to the design. Based on the design constraints on the cost and available space for the powertrain a single-mode or a multi-mode architecture might be preferred.

For a given vehicle application, selection of a good single-mode or multi-mode architecture accounting for the fuel economy and vehicle performance is a challenging problem because of its combinatorial nature. It is also a combined design and control problem where evaluation of each architecture design alternative under a drive cycle requires a control strategy to calculate the fuel consumption. Including the powertrain component design decision to the configuration selection problem poses additional challenges as it adds new continuous variables to the discrete problem. The goal of this dissertation was to solve this problem in a computationally tractable way.

In Chapter III, we introduced a modified bond graph representation of HEV architectures that was general enough to account for any given set powertrain components and PGs. Using the proposed representation, we formulated the problem of generating all feasible configurations in a general way. As an advantage of using a graphical representation mainly based on the concepts from bond graphs, the kinematic relationships among the powertrain components can be extracted from our representation. These equations were used in the evaluation of fuel economy and performance of the architectures. The configurations generated can be used alone in a single mode architecture or they can be combined together in a multi-mode architecture.

In Chapter IV we gave an overview of different aspects of the general HEV architecture design problem. We described the mathematical formulations and the proposed solution strategies for the case where the component sizes were given and for the case that includes component sizes as design decisions. We also discussed the solution of the combined design and control problem and proposed to use the nested formulation that solves the control problem for each design alternative considered during the design process. This approach can utilize the available control strategies developed for any given HEV design.

Chapter V described the design of a single-mode architecture design for given component sizes. Since the number of design alternatives was small enough to allow evaluation of all feasible alternatives separately, we preferred to use enumeration. We also identified the PG ratios and final drive ratio as the significant contributors of the fuel consumption and vehicle performance.

In Chapter VI we addressed the multi-mode architecture design problem with the same assumptions on the component sizes. A multi-mode architecture was created by combining two or more configurations in the same architecture. In order to switch from one configuration to another, we needed to add a certain number of clutches to the architecture. However, adding clutches to the architecture introduces additional complexity in the clutch control and as well as additional cost. In order to prevent an undesired number of clutches in the architecture, we introduced a complexity constraint measured by the number of clutches required to combine the selected modes. In the multi-mode architecture design, enumeration of all feasible designs was not computationally tractable and we introduced a search method capable of identifying good design alternatives.

In Chapter VII we extended the design problem to the case that includes the components sizes as design decisions. In Chapter V, we already showed the effect of gear ratio selections on the fuel economy and vehicle performance. If there was no tight constraints on the values of the gear ratios or no good values were available, the gear ratios should be designed together with the configuration. We proposed a decomposition-based formulation in order to solve the combined problem.

8.2 Contributions

Three major contributions of this dissertation are as follows

(i) A new representation of hybrid electric vehicle (HEV) architectures was created based on bond graphs that is general enough to represent architectures with any number of powertrain components connected through any number of planetary gears (PGs). Using this representation, a general formulation was derived to generate all feasible configurations, i.e., driving modes, for any given number of powertrain components and PGs.

(ii) A new, combined HEV architecture design and control problem formulation was derived and solution strategies for single and multi-mode architecture design problems for given powertrain component sizes and gear ratio selections were demonstrated.

(iii) A general decomposition-based formulation for the combined HEV architecture design and control problem was developed for the more general case that included component sizing as design decisions. This formulation included a partitioning model and a coordination strategy using analytical target cascading.

In order to demonstrate the capabilities of the methods proposed, we presented the generation results for all feasible power-split type 1-PG, 2-PG and 3-PG driving modes with one engine, two MGs and one ground. Then, using these feasible driving modes, we showed the design of 1 and 2-PG single and multi-mode architectures for different vehicle applications by considering fuel economy, vehicle performance and architecture complexity. Finally, for the design process of combined architecture and component sizing, we presented separate decompositions for single and multi-mode architectures. We demonstrated the capability by designing 1 and 2-PG single-mode and multi-mode architectures and gear ratios simultaneously for a particular vehicle application.

8.3 Limitations and Future Work

In this dissertation, we omitted the series and parallel type of configurations and considered only 2-dof power-split types as they provide the largest variety of architecture design alternatives. Series hybrid configurations have a very limited number of applications and generally a power-split configuration provides most of the functionality that a series hybrid configuration offer. For instance, *Zhang et al.* (2012) showed that removing the series hybrid mode together with one of the EV modes from the Chevrolet Volt design does not reduce the fuel economy significantly.

A parallel type configuration can be obtained by flipping the causality of the engine and considering kinematically 1-dof power-split type configurations (i.e. only one independent node in the PG system) during the configuration generation process. However, these designs do not provide any variety in terms of speed ratio and have to rely on an additional transmission. A comparison of the parallel configurations and power-split types can be made as a future study.

The search algorithm we introduced in Chapter VI was not robust enough and has to be tested with many initial designs. A better tractable method which can search the design space effectively requiring less human input should be studied.

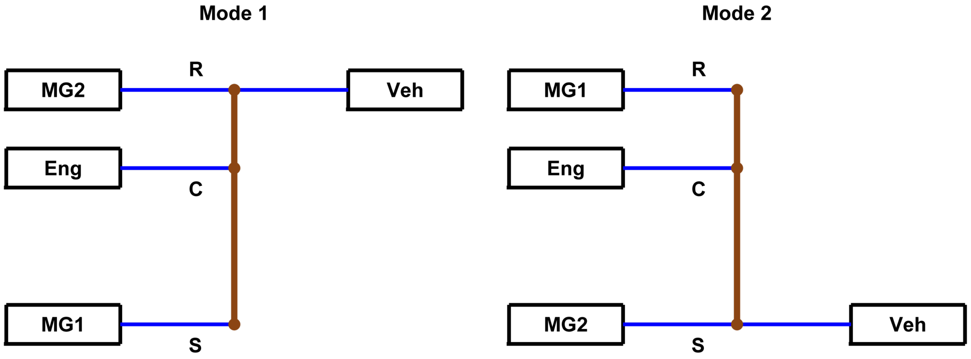
The problem formulation given in Chapter VII was solved considering only gear ratios as component sizing decisions. The formulation is general enough to account for all powertrain component sizes. However, adding these variables poses new challenges with the solution of the decomposed problem. The solution of this more general problem is necessary since fixing sizes of the powertrain components might favor some of the configurations during the design process. Also additional design freedom in the general problem gives the opportunity to improve the optimal results further. These considerations should be investigated as a future study.

Since the modified bond graphs used in this dissertation are general tools that can also be used to model and represent other types of systems besides automotive powertrain configurations, there is some potential to apply the proposed design framework to mechatronic system designs. In a general case, similar to the automotive powertrain architecture design, the first step is the generation all feasible configurations using the modified bond graph representation before the design process. Then, a system matrix consisting of continuous elements that define the configuration and component sizes must be created. A system level formulation designing the elements of this matrix and a subsystem level formulation that includes the feasibility based on previously generated configurations can be used to design general mechatronic systems. Such possible generalizations of the proposed methods to other types of systems will be investigated as future study.

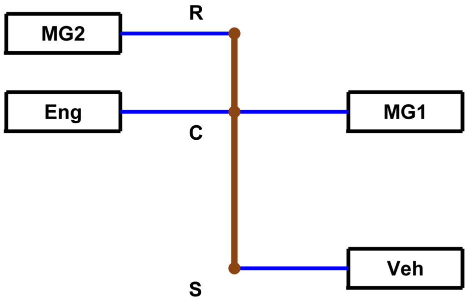
APPENDICES

APPENDIX A

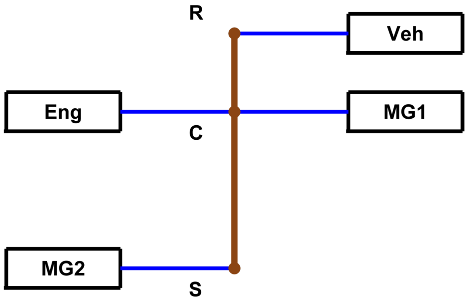
1-PG Hybrid and Pure Electric Modes



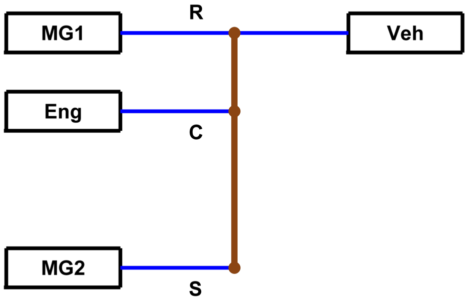
Mode 3



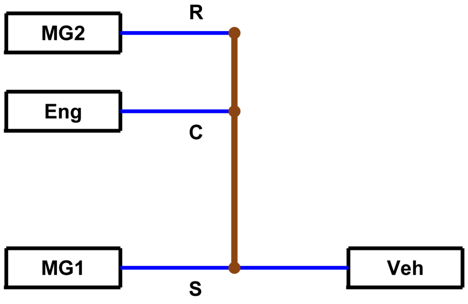
Mode 4



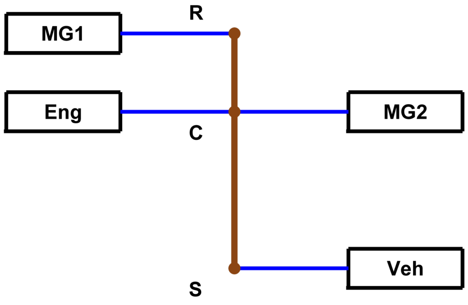
Mode 5



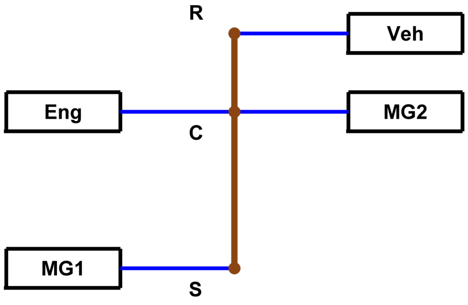
Mode 6



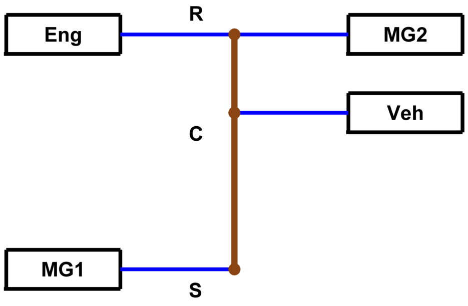
Mode 7



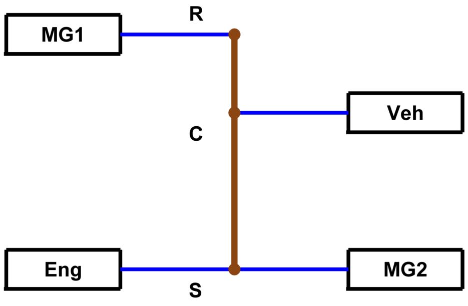
Mode 8



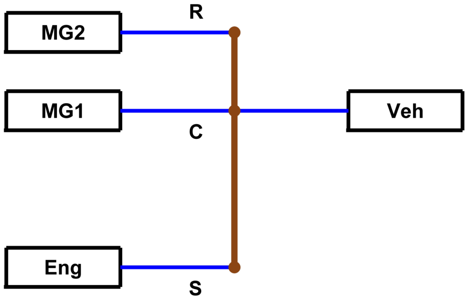
Mode 9



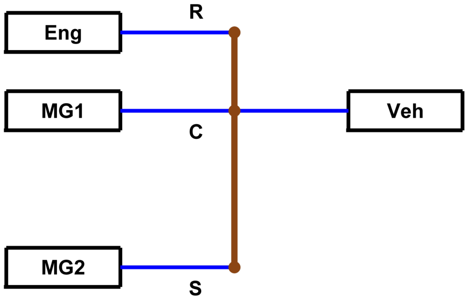
Mode 10



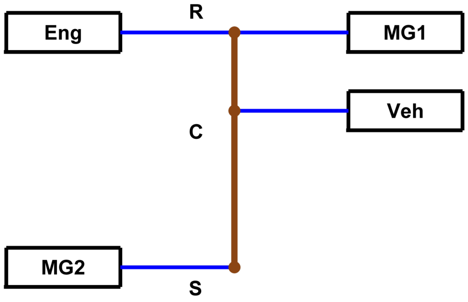
Mode 11



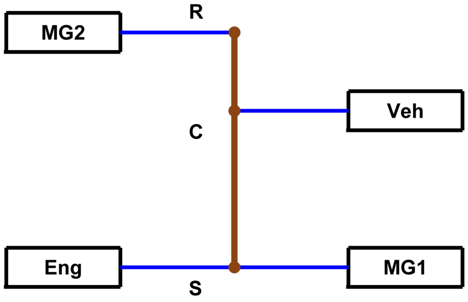
Mode 12



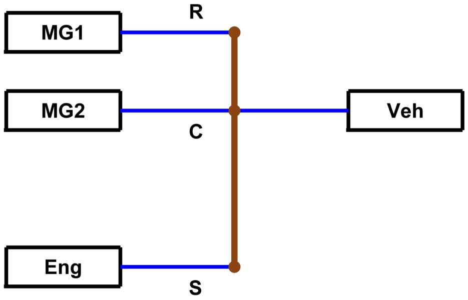
Mode 13



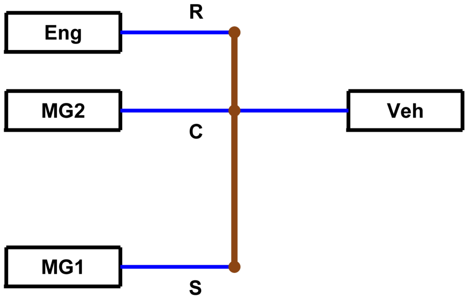
Mode 14



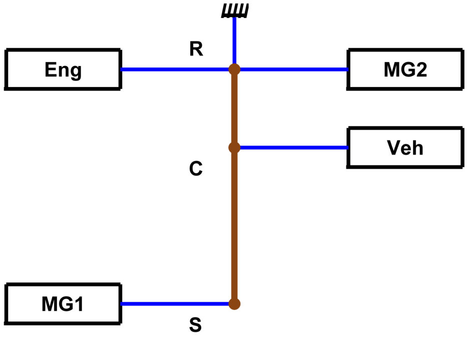
Mode 15



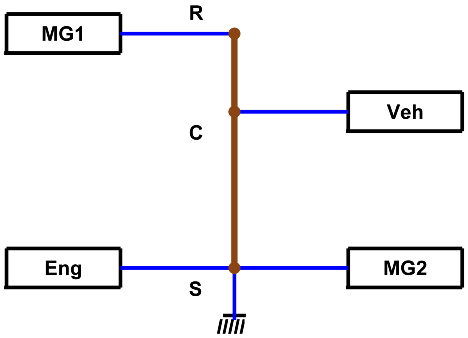
Mode 16



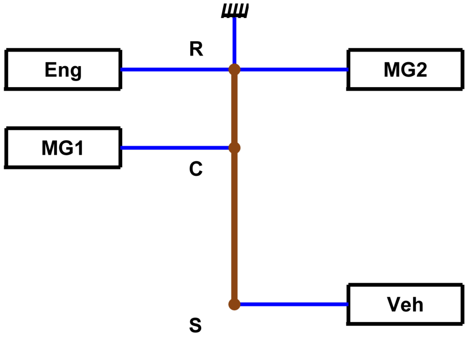
Mode 17



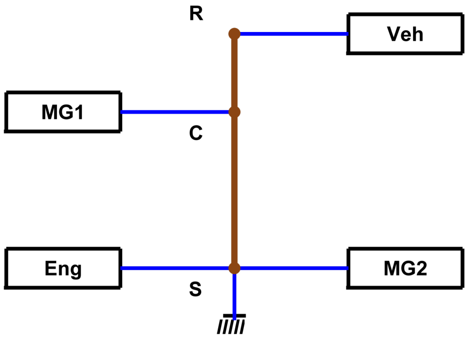
Mode 18



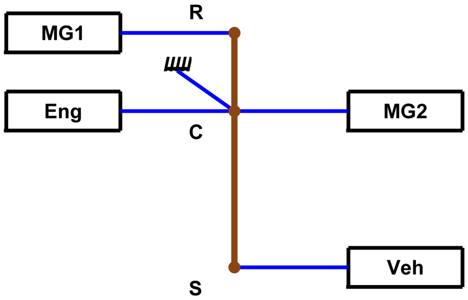
Mode 19



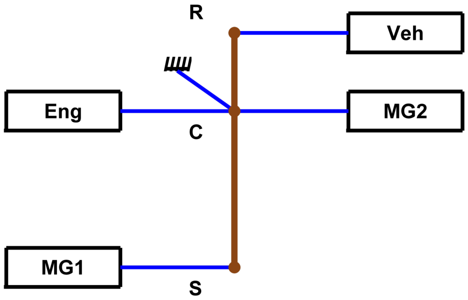
Mode 20



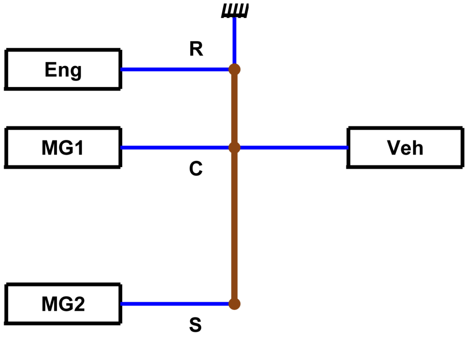
Mode 21



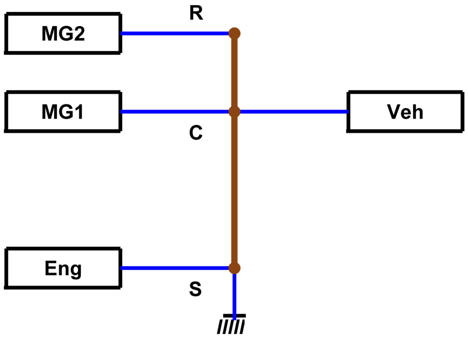
Mode 22



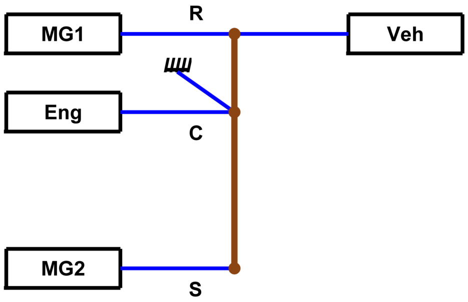
Mode 23



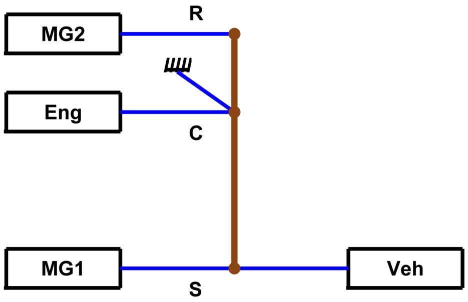
Mode 24



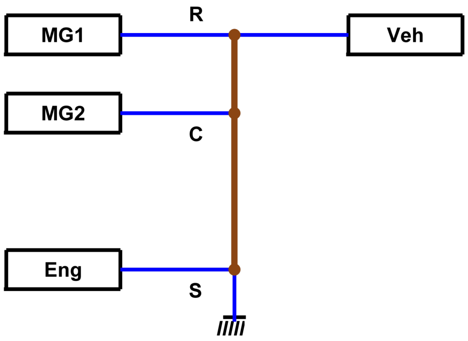
Mode 25



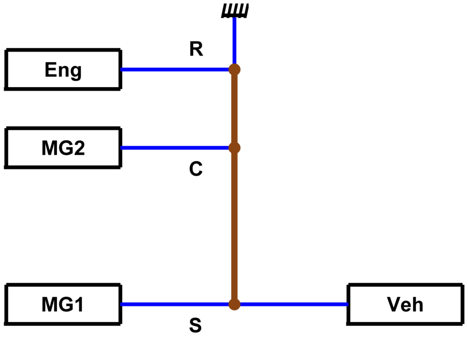
Mode 26



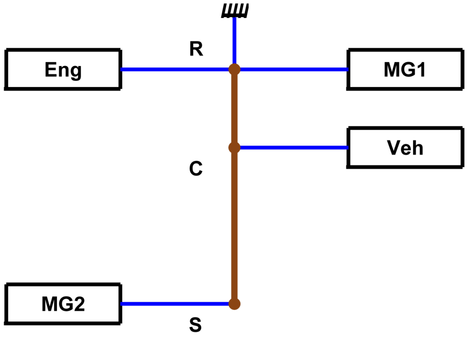
Mode 27



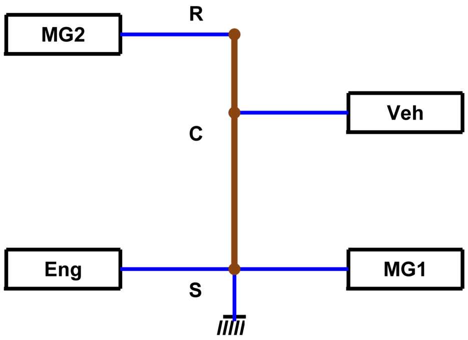
Mode 28



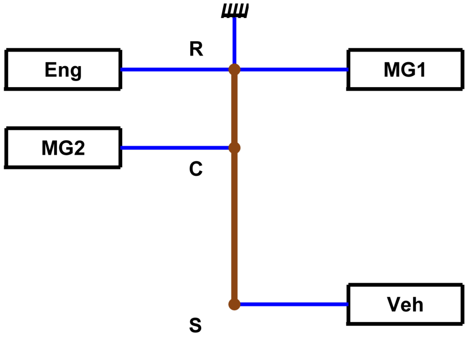
Mode 29



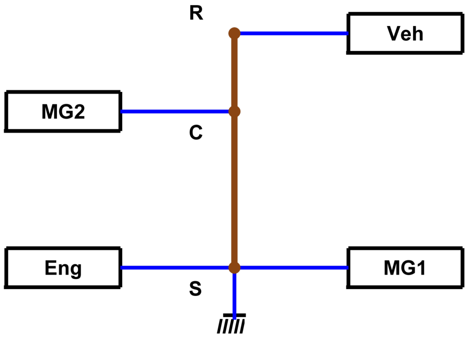
Mode 30



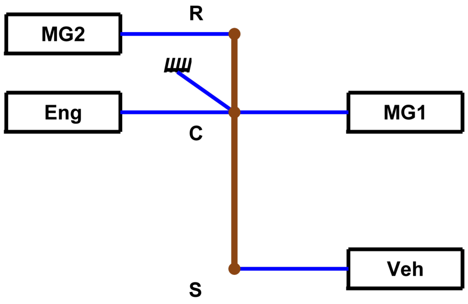
Mode 31



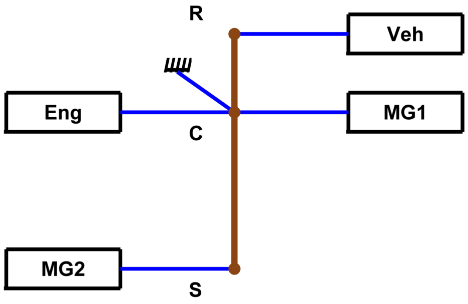
Mode 32



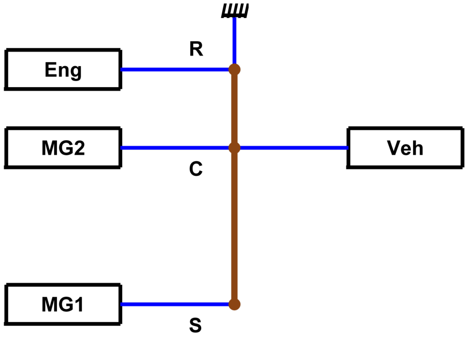
Mode 33



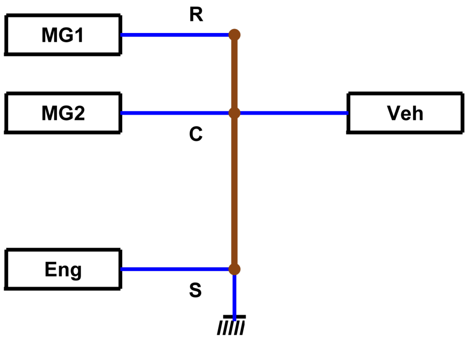
Mode 34



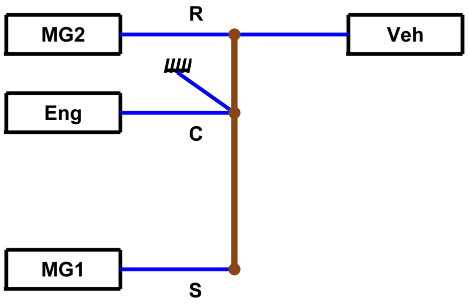
Mode 35



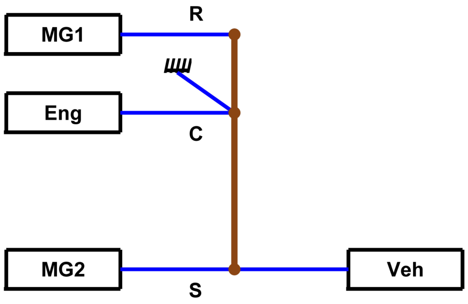
Mode 36



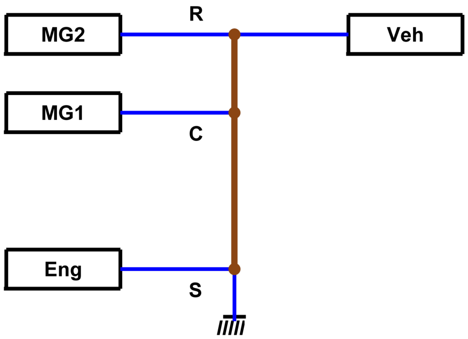
Mode 37



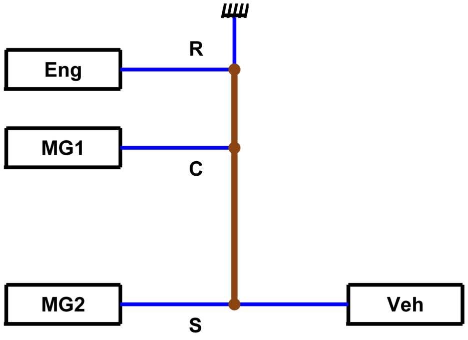
Mode 38



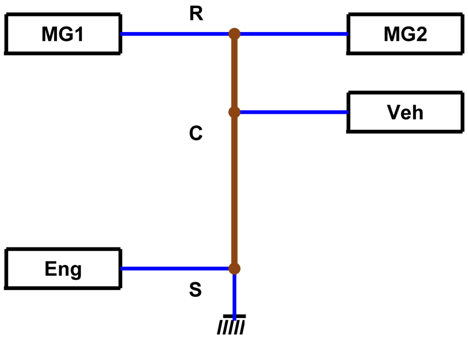
Mode 39



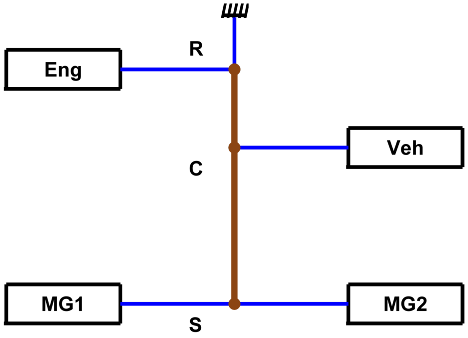
Mode 40



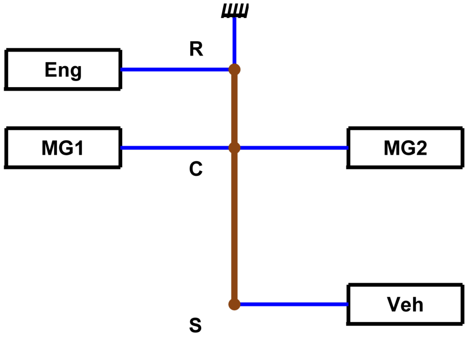
Mode 41



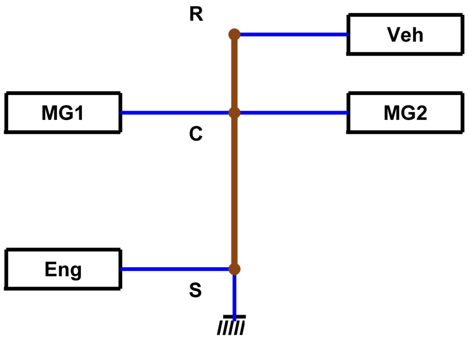
Mode 42



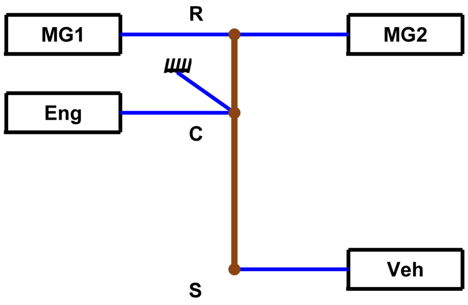
Mode 43



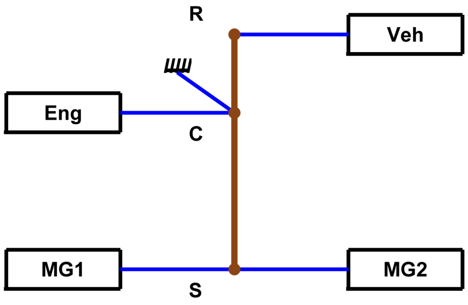
Mode 44



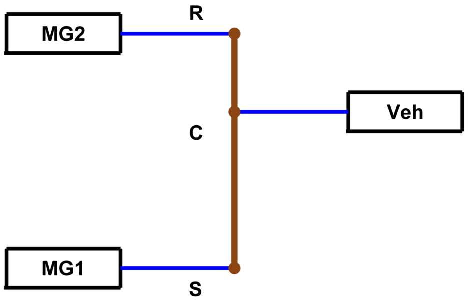
Mode 45



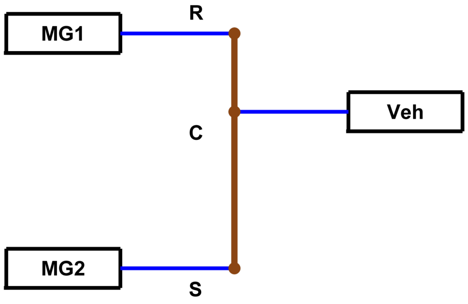
Mode 46



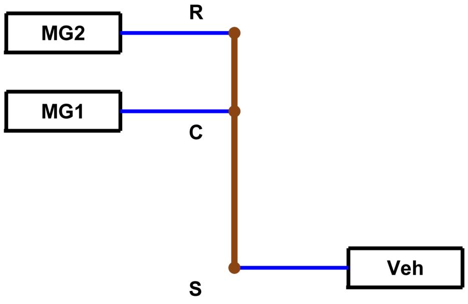
Mode 47



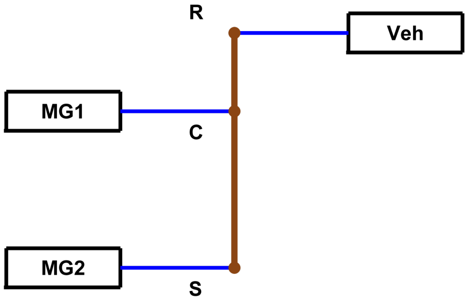
Mode 48

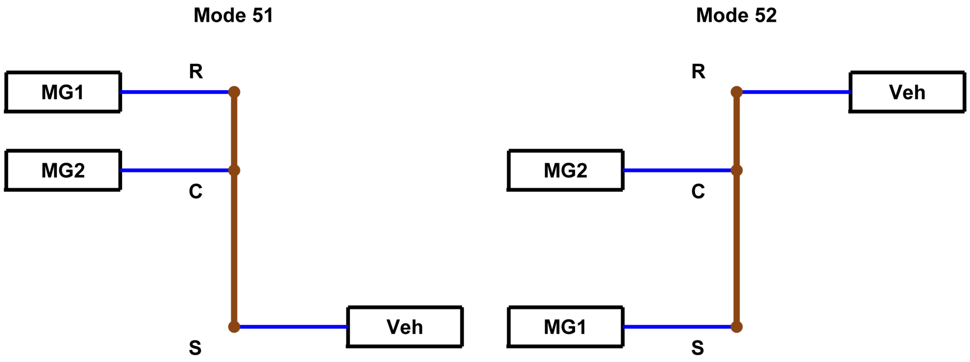


Mode 49



Mode 50



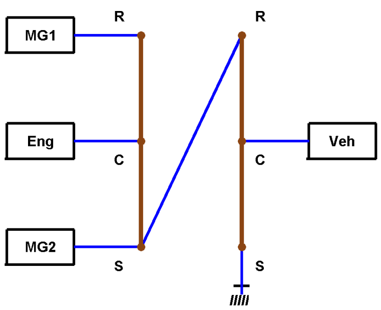

Mode 51
R
MG1
MG2
C
S
Veh
Mode 52
R
Veh
MG2
C
MG1
S

APPENDIX B

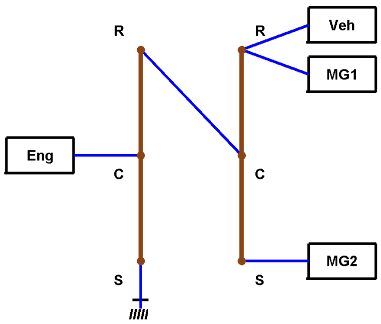
Selected 2-PG Hybrid and Pure Electric Modes

The following are the top 15 feasible 2-PG hybrid modes selected from Case Study 3 in Chapter V. The remaining are the pure electric modes which can be combined with the first 15 hybrid modes using at most two clutches.

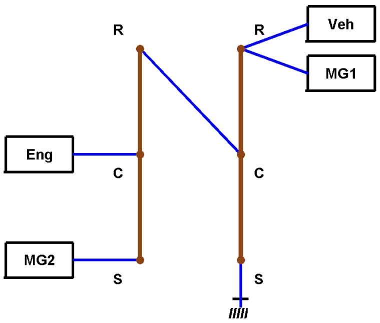
Mode 1



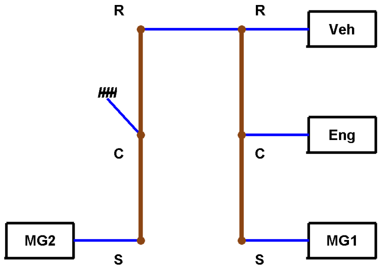
Mode 2



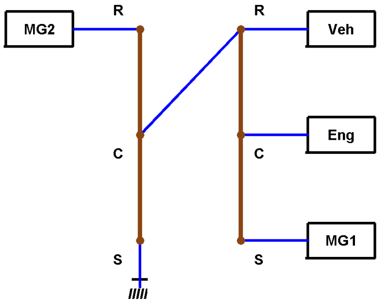
Mode 3



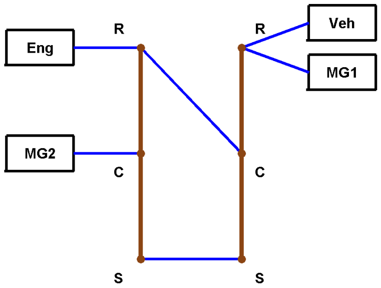
Mode 4



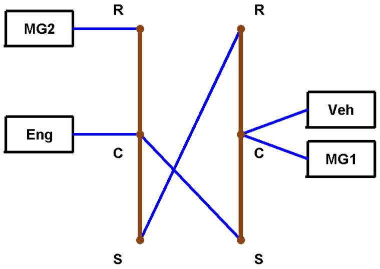
Mode 5



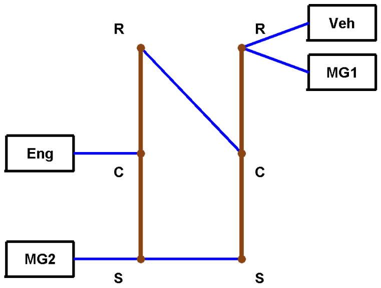
Mode 6



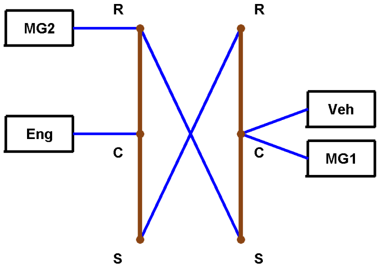
Mode 7



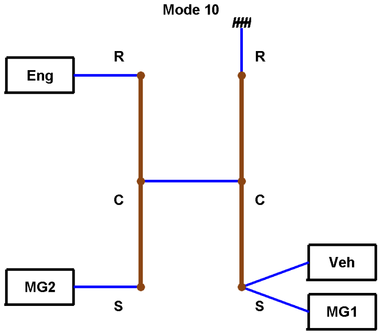
Mode 8



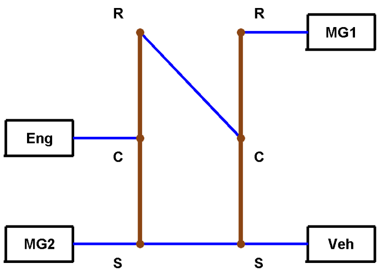
Mode 9



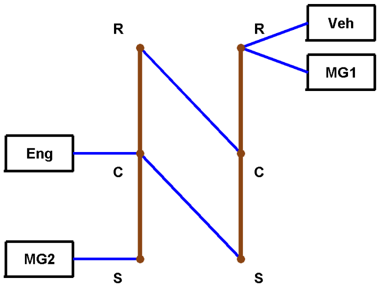
Mode 10



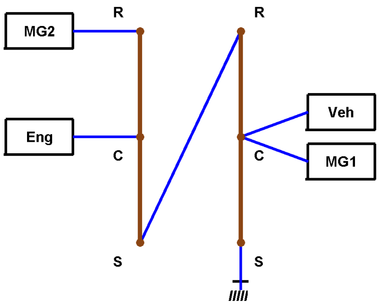
Mode 11



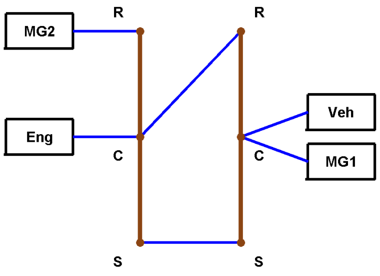
Mode 12



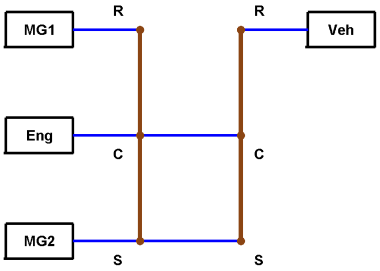
Mode 13



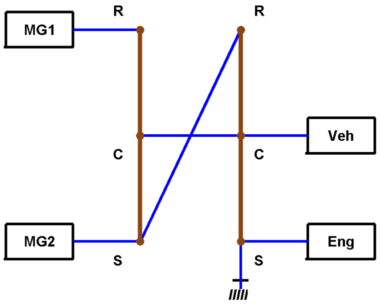
Mode 14



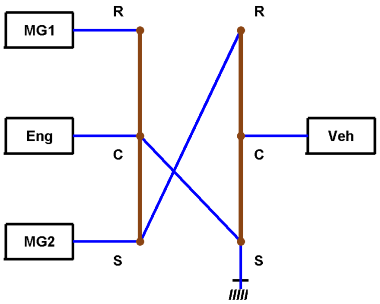
Mode 15



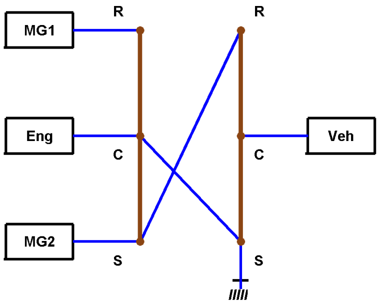
Mode 16



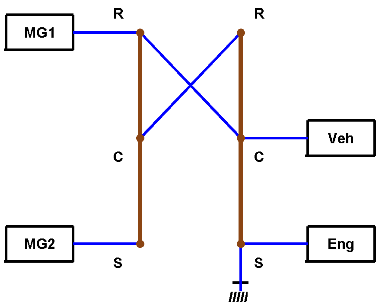
Mode 17



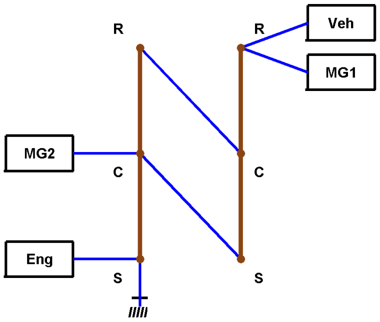
Mode 18



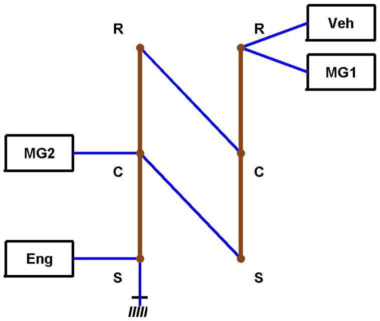
Mode 19



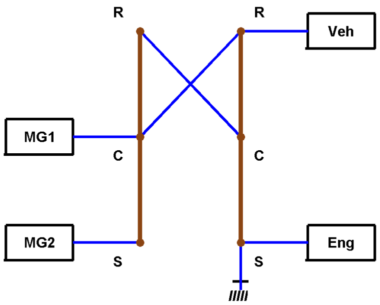
Mode 20



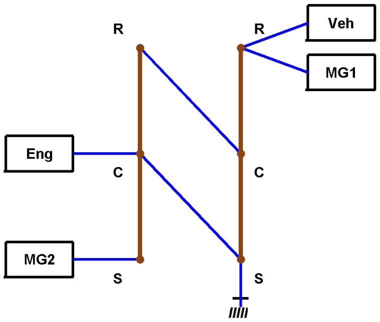
Mode 21



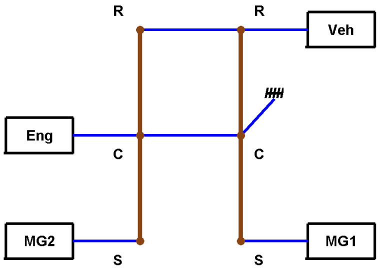
Mode 22



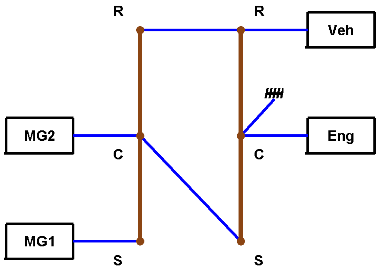
Mode 23



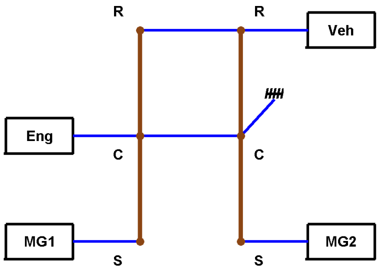
Mode 24



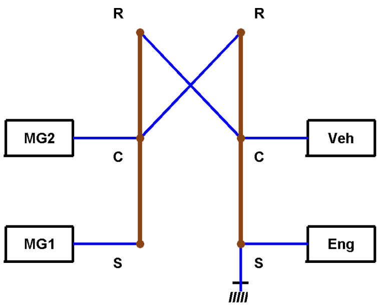
Mode 25



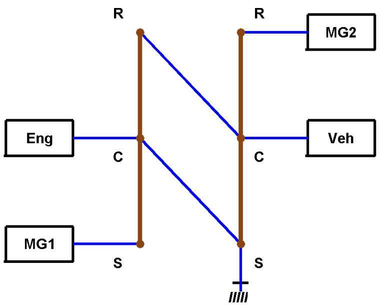
Mode 26



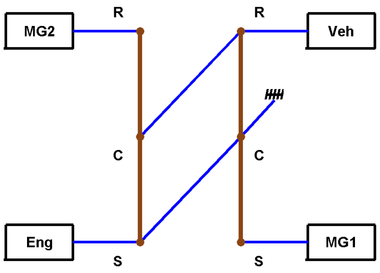
Mode 27



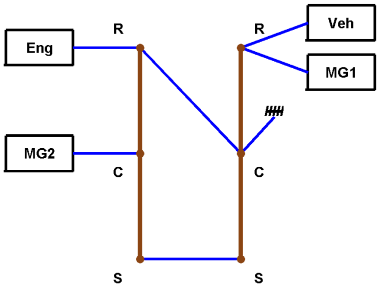
Mode 28



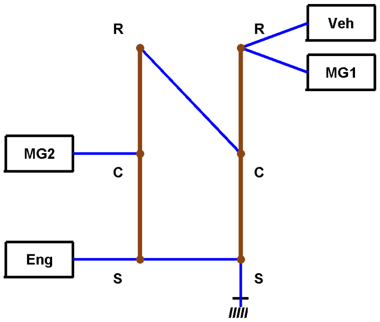
Mode 29



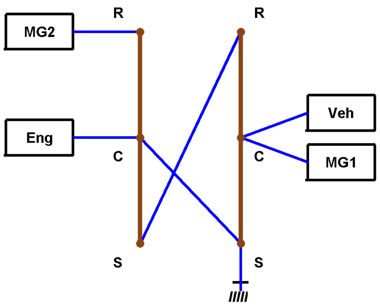
Mode 30



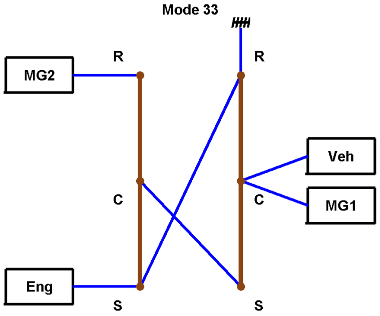
Mode 31



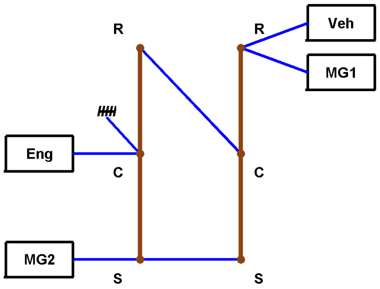
Mode 32



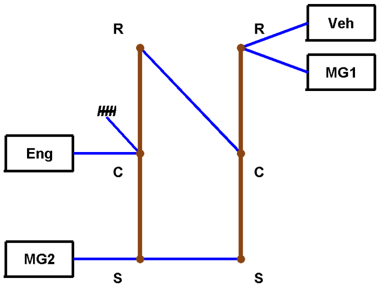
Mode 33



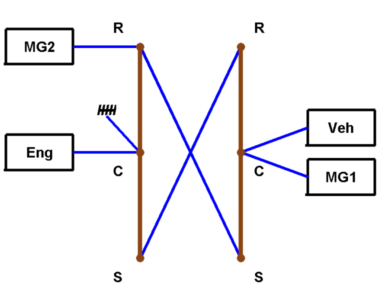
Mode 34



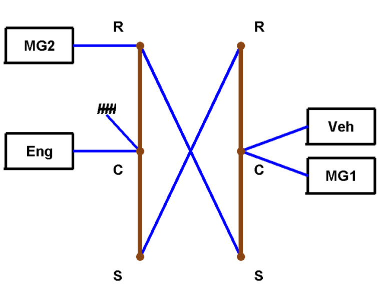
Mode 35



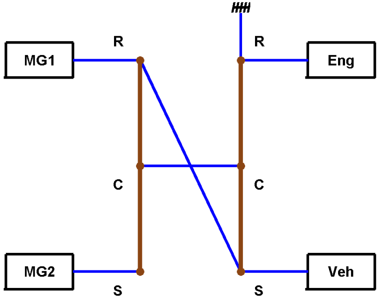
Mode 36



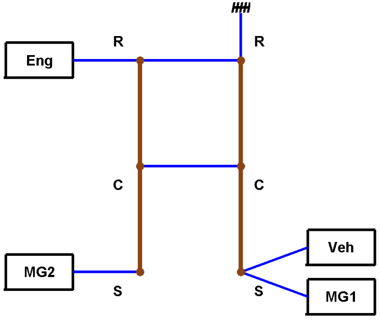
Mode 37



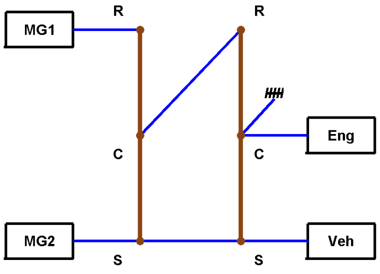
Mode 38



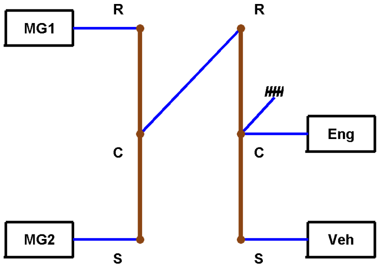
Mode 39



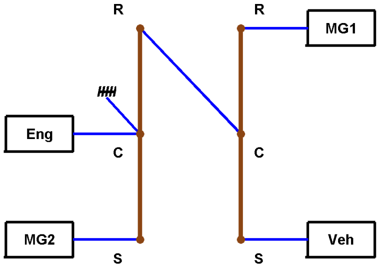
Mode 40



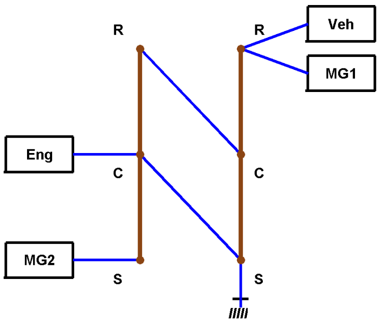
Mode 41



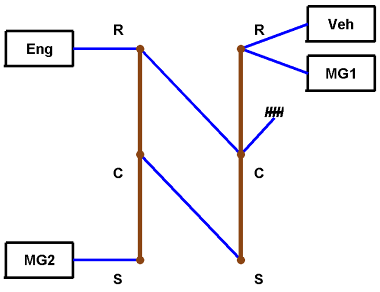
Mode 42



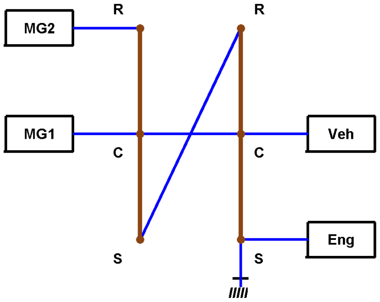
Mode 43



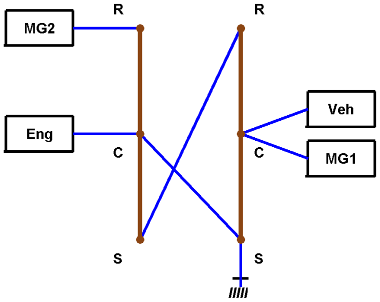
Mode 44



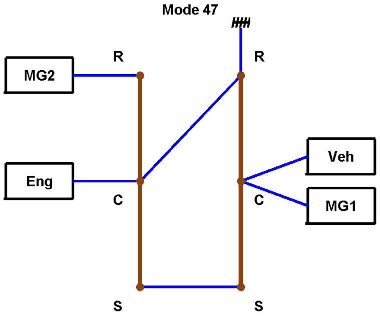
Mode 45



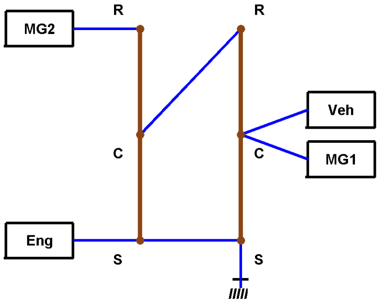
Mode 46



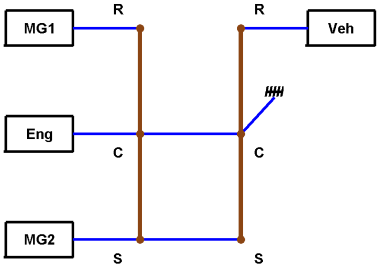
Mode 47



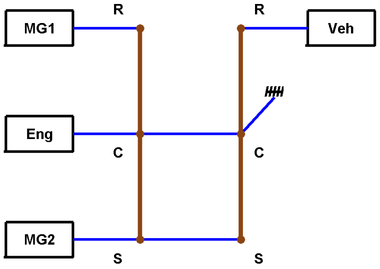
Mode 48

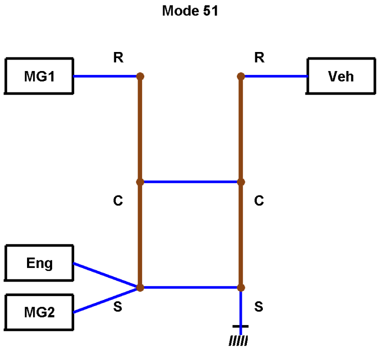


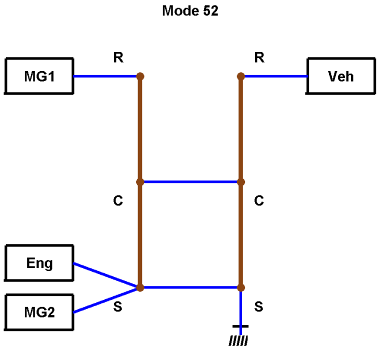
Mode 49



Mode 50




Mode 51
MG1
R
R
Veh
C
C
Eng
MG2
S
S


Mode 52
MG1
R
R
Veh
C
C
Eng
MG2
S
S

BIBLIOGRAPHY

BIBLIOGRAPHY

Ahn, K., S. Cho, and S. Cha (2008), Optimal operation of the power-split hybrid electric vehicle powertrain, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *222*(5), 789–800.

Ai, X., and S. Anderson (2005), An electro-mechanical infinitely variable transmisson for hybrid electric vehicles, *SAE Technical Paper 2005-01-0281*.

Allison, J., D. Walsh, M. Kokkolaras, P. Y. Papalambros, and M. Cartmell (2006), Analytical target cascading in aircraft design, in *44th AIAA aerospace sciences meeting and exhibit*, pp. 9–12.

Antony, G. G., and A. Pantelides (2006), Precision planetary servo gearheads, *Tech. rep.*, American Gear Manufacturers Association.

Aoki, K., S. Kuroda, S. Kajiwara, H. Sato, and Y. Yamamoto (2000), Development of integrated motor assist hybrid system: Development of the insight, a personal hybrid coupe, *Tech. rep.*, SAE Technical Paper.

Bellman, R. (1954), The theory of dynamic programming, *Tech. rep.*, DTIC Document.

Bendsøe, M. P., and N. Kikuchi (1988), Generating optimal topologies in structural design using a homogenization method, *Computer methods in applied mechanics and engineering*, *71*(2), 197–224.

Benford, H. L., and M. B. Leising (1981), The lever analogy: A new tool in transmission analysis, *Tech. rep.*, SAE Technical Paper.

Blouin, V., G. Fadel, I. Haque, and J. Wagner (2004), Continuously variable transmission design for optimum vehicle performance by analytical target cascading, *International Journal of Heavy Vehicle Systems*, *11*(3), 327–348.

Borrvall, T., and J. Petersson (2003), Topology optimization of fluids in stokes flow, *International journal for numerical methods in fluids*, *41*(1), 77–107.

Buchsbaum, F., and F. Freudenstein (1970), Synthesis of kinematic structure of geared kinematic chains and other mechanisms, *Journal of mechanisms*, *5*(3), 357–392.

Byrd, R. H., M. E. Hribar, and J. Nocedal (1999), An interior point algorithm for large-scale nonlinear programming, *SIAM Journal on Optimization*, *9*(4), 877–900.

Byrd, R. H., J. C. Gilbert, and J. Nocedal (2000), A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming*, *89*(1), 149–185.

Cagan, J., M. I. Campbell, S. Finger, and T. Tomiyama (2005), A framework for computational design synthesis: model and applications, *Journal of Computing and Information Science in Engineering*, *5*(3), 171–181.

Chatterjee, G., and L.-W. Tsai (1996), Computer-aided sketching of epicyclic-type automatic transmission gear trains, *Journal of Mechanical Design*, *118*(3), 405–411.

Cheong, K. L., P. Y. Li, and T. R. Chase (2011), Optimal design of power-split transmissions for hydraulic hybrid passenger vehicles, in *American Control Conference (ACC), 2011*, pp. 3295–3300, IEEE.

Choudhary, R., A. Malkawi, and P. Papalambros (2005), Analytic target cascading in simulation-based building design, *Automation in construction*, *14*(4), 551–568.

Conlon, B. M., P. J. Savagian, A. G. Holmes, M. O. Harpster, et al. (2011), Output split electrically-variable transmission with electric propulsion using one or two motors, uS Patent 7,867,124.

Delprat, S., T. Guerra, and J. Rimaux (2002), Control strategies for hybrid vehicles: optimal control, in *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, vol. 3, pp. 1681–1685, IEEE.

Delprat, S., J. Lauber, T. M. Guerra, and J. Rimaux (2004), Control of a parallel hybrid powertrain: optimal control, *Vehicular Technology, IEEE Transactions on*, *53*(3), 872–881.

Dijkstra, E. W. (1959), A note on two problems in connexion with graphs, *Numerische mathematik*, *1*(1), 269–271.

Fan, Z., J. Hu, K. Seo, E. Goodman, R. Rosenberg, and B. Zhang (2001), Bond graph representation and gp for automated analog filter design, in *Proc. of Genetic and Evolutionary Computation Conference Late-Breaking Papers*, pp. 81–86, ISGEC Press, San Francisco.

Fathy, H. K., J. A. Reyer, P. Y. Papalambros, and A. Ulsov (2001), On the coupling between the plant and controller optimization problems, in *American Control Conference, 2001. Proceedings of the 2001*, vol. 3, pp. 1864–1869, IEEE.

Fortin, S. (1996), The graph isomorphism problem, *Tech. rep.*, Technical Report 96-20, University of Alberta, Edomonton, Alberta, Canada.

Freudenstein, F., and A. Yang (1972), Kinematics and statics of a coupled epicyclic spur-gear train, *Mechanism and Machine Theory*, *7*(2), 263–275.

Freyermuth, V., E. Fallas, and A. Rousseau (2008), Comparison of powertrain configuration for plug-in hevs from a fuel economy perspective, *Tech. rep.*, SAE Technical Paper.

Harary, F., C. King, A. Mowshowitz, and R. C. Read (1971), Cospectral graphs and digraphs, *Bulletin of the London Mathematical Society*, *3*(3), 321–328.

Holmes, A., and M. Schmidt (2002), Hybrid electric powertrain including a two-mode electrically variable transmission, uS Patent 6,478,705.

Holmes, A., D. Klemen, and M. Schmidt (2003), Electrically variable transmission with selective input split, compound split, neutral and reverse modes, uS Patent 6,527,658.

Hsieh, H.-I., and L.-W. Tsai (1996a), Kinematic analysis of epicyclic-type transmission mechanisms using the concept of fundamental geared entities, *Journal of Mechanical Design*, *118*(2), 294–299.

Hsieh, H.-I., and L.-W. Tsai (1996b), A methodology for enumeration of clutching sequences associated with epicyclic-type automatic transmission mechanisms, *Tech. rep.*, SAE Technical Paper.

Huang, C., and B. Roth (1993), Dimensional synthesis of closed-loop linkages to match force and position specifications, *Journal of Mechanical Design*, *115*(2), 194–198.

Jensen, E. D. (1992), *Topological structural design using genetic algorithms*, UMI.

Kahraman, A., H. Ligata, K. Kienzle, and D. Zini (2004), A kinematics and power flow analysis methodology for automatic transmission planetary gear trains, *Journal of Mechanical Design*, *126*(6), 1071–1081.

Kang, N., M. Kokkolaras, P. Papalambros, J. Park, W. Na, S. Yoo, and D. Featherman (2012), Optimal design of commercial vehicle systems using analytical target cascading, in *12th American Institute of Aeronautics and Astronautics Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, pp. 1–13.

Karnopp, D. C., D. L. Margolis, and R. C. Rosenberg (2012), *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, 5th ed., Wiley, Hoboken, NJ.

Kicinger, R., T. Arciszewski, and K. D. Jong (2005), Evolutionary computation and structural design: A survey of the state-of-the-art, *Computers & Structures*, *83*(23), 1943–1978.

Kim, H. M. (2001), Target cascading in optimal system design, Ph.D. thesis, The University of Michigan.

Kim, H. M., M. Kokkolaras, L. S. Louca, G. J. Delagrammatikas, N. F. Michelena, Z. S. Filipi, P. Y. Papalambros, J. L. Stein, and D. N. Assanis (2002), Target cascading in vehicle redesign: a class vi truck study, *International Journal of Vehicle Design*, *29*(3), 199–225.

Kim, H. M., N. F. Michelena, P. Y. Papalambros, and T. Jiang (2003a), Target cascading in optimal system design, *Journal of mechanical design*, *125*(3), 474–480.

Kim, H. M., D. G. Rideout, P. Y. Papalambros, and J. L. Stein (2003b), Analytical target cascading in automotive vehicle design, *Journal of Mechanical Design*, *125*(3), 481–489.

Kim, N., S. Cha, and H. Peng (2011), Optimal control of hybrid electric vehicles based on pontryagin's minimum principle, *Control Systems Technology, IEEE Transactions on*, *19*(5), 1279–1287.

Kiziltas, G., D. Psychoudakis, J. L. Volakis, and N. Kikuchi (2003), Topology design optimization of dielectric substrates for bandwidth improvement of a patch antenna, *Antennas and Propagation, IEEE Transactions on*, *51*(10), 2732–2743.

Kokkolaras, M., L. Louca, G. Delagrammatikas, N. Michelena, Z. Filipi, P. Papalambros, J. Stein, and D. Assanis (2004), Simulation-based optimal design of heavy trucks by model-based decomposition: An extensive analytical target cascading case study, *International Journal of Heavy Vehicle Systems*, *11*(3), 403–433.

Koza, J. R., F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap (1997), Automated synthesis of analog electrical circuits by means of genetic programming, *Evolutionary Computation, IEEE Transactions on*, *1*(2), 109–128.

Li, Z., M. Kokkolaras, P. Papalambros, and S. J. Hu (2008), Product and process tolerance allocation in multistation compliant assembly using analytical target cascading, *Journal of Mechanical Design*, *130*(9), 091,701.

Lin, C.-C., H. Peng, J. W. Grizzle, and J.-M. Kang (2003), Power management strategy for a parallel hybrid electric truck, *Control Systems Technology, IEEE Transactions on*, *11*(6), 839–849.

Liu, J. (2007), Modeling, configuration and control optimization of power-split hybrid vehicles, Ph.D. thesis, The University of Michigan.

Liu, J., and H. Peng (2006), Control optimization for a power-split hybrid vehicle, in *American Control Conference, 2006*, pp. 6–pp, IEEE.

Liu, J., and H. Peng (2008), Modeling and control of a power-split hybrid vehicle, *Control Systems Technology, IEEE Transactions on*, *16*(6), 1242–1251.

Lohn, J. D., and S. P. Colombano (1998), Automated analog circuit synthesis using a linear representation, in *Evolvable Systems: From Biology to Hardware*, pp. 125–133, Springer.

Martins, J. R., and A. B. Lambe (2013), Multidisciplinary design optimization: a survey of architectures, *AIAA journal*, *51*(9), 2049–2075.

Masrur, A., R. Smith, and A. Nedungadi (2012), Quantitative analysis of a hybrid electric hmmwv for fuel economy improvement, *Tech. rep.*, DTIC Document.

MATLAB (2014), *graphisomorphism*, The MathWorks Inc., Natick, Massachusetts.

McKay, B. D., et al. (1981), *Practical graph isomorphism*, Department of Computer Science, Vanderbilt University.

Michelena, N., H. Park, and P. Y. Papalambros (2003), Convergence properties of analytical target cascading, *AIAA journal*, *41*(5), 897–905.

Nelson, P., K. Amine, and A. Rousseau (2007), Advanced lithium-ion batteries for plug-in hybrid-electric vehicles, *Lithium Batteries: Research, Technology and Applications*, p. 203.

Nicolaou, C. A., J. Apostolakis, and C. S. Pattichis (2009), De novo drug design using multiobjective evolutionary graphs, *Journal of Chemical Information and Modeling*, *49*(2), 295–307.

Paganelli, G., S. Delprat, T.-M. Guerra, J. Rimaux, and J.-J. Santin (2002), Equivalent consumption minimization strategy for parallel hybrid powertrains, in *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 4, pp. 2076–2081, IEEE.

Peters, D., P. Papalambros, and A. Ulsoy (2010), Relationship between coupling and the controllability grammian in co-design problems, in *American Control Conference (ACC), 2010*, pp. 623–628, IEEE.

Pontryagin, L. S. (1987), *Mathematical theory of optimal processes*, CRC Press.

Raghavan, M. (1989), Analytical methods for designing linkages to match force specifications, Ph.D. thesis, Stanford University.

Raghavan, M., N. K. Bucknor, and J. D. Hendrickson (2007), Electrically variable transmission having three interconnected planetary gear sets, two clutches and two brakes, uS Patent 7,179,187.

Read, R. C., and D. G. Corneil (1977), The graph isomorphism disease, *Journal of Graph Theory*, *1*(4), 339–363.

Rosenberg, R. C. (1971), State-space formulation for bond graph models of multiport systems, *Journal of Dynamic Systems, Measurement, and Control*, *93*(1), 35–40.

Rosenberg, R. C., and D. Karnopp (1972), A definition of the bond graph language, *Journal of Dynamic Systems, Measurement, and Control*, *94*(3), 179–182.

Roth, B. (1967), Finite-position theory applied to mechanism synthesis, *Journal of Applied Mechanics*, *34*(3), 599–605.

Sandgren, E., E. Jensen, and J. Welton (1990), Topological design of structural components using genetic optimization methods.

Sasaki, S. (1998), Toyota's newly developed hybrid powertrain, in *Power Semiconductor Devices and ICs, 1998. ISPSD 98. Proceedings of the 10th International Symposium on*, pp. 17–22, IEEE.

Schmidt, M. (1996a), Two-mode, split power, electro-mechanical transmission, uS Patent 5,577,973.

Schmidt, M. (1996b), Two-mode, input-split, parallel, hybrid transmission, uS Patent 5,558,588.

Schmidt, M. (1999a), Electro-mechanical powertrain, uS Patent 5,935,035.

Schmidt, M. (1999b), Two-mode, compound-split electro-mechanical vehicular transmission, uS Patent 5,931,757.

Schneider, G., and U. Fechner (2005), Computer-based de novo design of drug-like molecules, *Nature Reviews Drug Discovery*, *4*(8), 649–663.

Schneider, G., M.-L. Lee, M. Stahl, and P. Schneider (2000), De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks, *Journal of Computer-Aided Molecular Design*, *14*(5), 487–494.

Seo, K., Z. Fan, J. Hu, E. D. Goodman, and R. C. Rosenberg (2003), Toward a unified and automated design methodology for multi-domain dynamic systems using bond graphs and genetic programming, *Mechatronics*, *13*(8), 851–885.

Serrao, L., S. Onori, and G. Rizzoni (2009), Ecms as a realization of pontryagin's minimum principle for hev control, in *American Control Conference*, pp. 3964–3969, IEEE.

Shea, K., and A. C. Starling (2003), From discrete structures to mechanical systems: a framework for creating performance-based parametric synthesis tools, in *Proceedings of the AAAI 2003 Symposium on Computational Synthesis: From Basic Building Blocks to High Level Functionality*, pp. 210–217.

Snavely, G., L. Pomrehn, and P. Papalambros (1990), Toward a vocabulary for classifying research in mechanical design automation, in *First International Workshop on Formal Methods in Engineering Design, Manufacturing and Assembly*, Citeseer.

Tosserams, S., L. Etman, P. Papalambros, and J. Rooda (2005), Augmented lagrangian relaxation for analytical target cascading, in *In 6th World Congress on Structural and Multidisciplinary Optimization, May 30–June 3*, Citeseer.

Tosserams, S., L. Etman, P. Papalambros, and J. Rooda (2006), An augmented lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers, *Structural and Multidisciplinary Optimization*, *31*(3), 176–189.

Walters, W. P., M. T. Stahl, and M. A. Murcko (1998), Virtual screening–an overview, *Drug Discovery Today*, *3*(4), 160–178.

Weber, L. (2002), Multi-component reactions and evolutionary chemistry, *Drug Discovery Today*, *7*(2), 143–147.

Whitefoot, J. W., K. Ahn, and P. Y. Papalambros (2010), The case for urban vehicles: powertrain optimization of a power-split hybrid for fuel economy on multiple drive cycles, in *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 197–204, American Society of Mechanical Engineers.

Zebulum, R. S., M. A. Pacheco, and M. Vellasco (1998), Comparison of different evolutionary methodologies applied to electronic filter design, in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pp. 434–439, IEEE.

Zhang, X., C.-T. Li, D. Kum, and H. Peng (2012), Prius+ and volt- : Configuration analysis of power-split hybrid vehicles with a single planetary gear, *IEEE Transactions on Vehicular Technology*, *61*(8), 3544–3552.