Systematic methodology for generation and design of hybrid vehicle powertrains

Bilal Kabalan

N°d'ordre NNT : xxx

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole Doctorale N° 160
Électronique, Électrotechnique, Automatique (EEA)

Spécialité de doctorat : Génie électrique
Discipline : Sciences pour l'ingénieur

Soutenue publiquement le 10/03/2020, par :

Bilal KABALAN

Systematic methodology for generation and design of hybrid vehicle powertrains

---

Méthodologie de génération systématique et de conception des chaines de traction de véhicules hybrides

Devant le jury composé de :

M. Hofman Theo, Professeur Associé, Eindhoven University of Technology (TU/e)	Rapporteur
Mme Chrenko Daniela, Maître de Conférences, Université de Technologie de Belfort-Montbéliard	Rapporteuse
M. Sari Ali, Professeur des Universités, Université Claude Bernard Lyon1	Examinateur
Mme Semail Betty, Professeure des Universités, Université de Lille	Examinatrice
M. Trigui Rochdi, Directeur de Recherche, IFSTTAR	Directeur de thèse
M. Vinot Emmanuel, Chargé de recherche, IFSTTAR	Examinateur
M. Dumand Clément, Responsable Industriel, Groupe PSA	Invité

Université Claude Bernard – LYON 1

Président de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-Président du Conseil d'Administration	M. Didier REVEL
Vice-Président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALLIER
Vice-Président de la Commission de Recherche	M. Jean-François MORNEX
Directeur Général des Services	M. Damien VERHAEGHE

COMPOSANTES SANTE

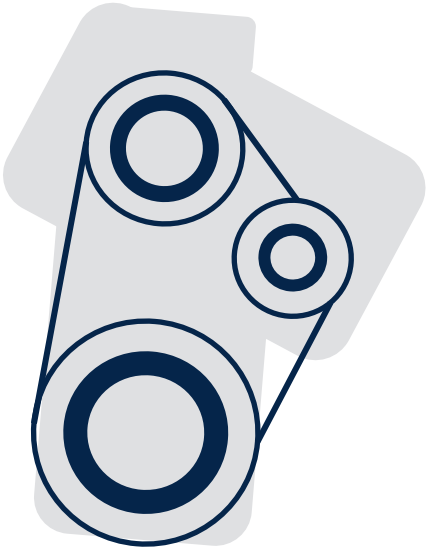
Faculté de Médecine Lyon-Est – Claude Bernard	Doyen : M. Gilles RODE
Faculté de Médecine et Maïeutique Lyon Sud Charles. Mérieux	Doyenne : Mme Carole BURILLON
UFR d'Odontologie	Doyenne : Mme Dominique SEUX
Institut des Sciences Pharmaceutiques et Biologiques	Directrice : Mme Christine VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. Xavier PERROT
Département de Formation et Centre de Recherche en Biologie Humaine	Directrice : Mme Anne-Marie SCHOTT

COMPOSANTES & DEPARTEMENTS DE SCIENCES & TECHNOLOGIE

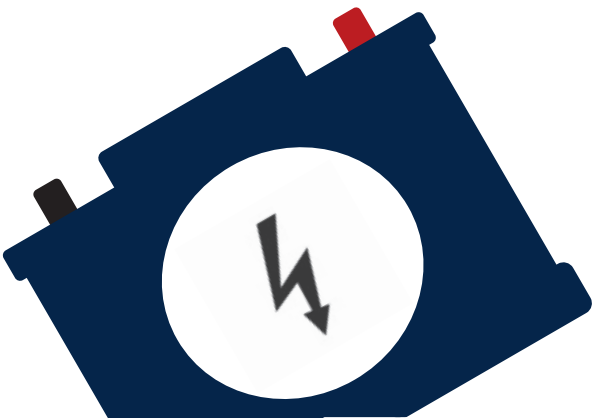
UFR Biosciences	Directrice : Mme Kathrin GIESELER
Département Génie Electrique et des Procédés (GEP)	Directrice : Mme Rosaria FERRIGNO
Département Informatique	Directeur : M. Behzad SHARIAT
Département Mécanique	Directeur M. Marc BUFFAT
UFR - Faculté des Sciences	Administrateur provisoire : M. Bruno ANDRIOLETTI
UFR (STAPS)	Directeur : M. Yannick VANPOULLE
Observatoire de Lyon	Directrice : Mme Isabelle DANIEL
Ecole Polytechnique Universitaire Lyon 1	Directeur : Emmanuel PERRIN
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur : Gérard PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. Christophe VITON
Institut de Science Financière et d'Assurances	Directeur : M. Nicolas LEBOISNE
ESPE	Administrateur Provisoire : M. Pierre CHAREYRON

PSA GROUPE

IFSTTAR



SYSTEMATIC METHODOLOGY
FOR GENERATION AND DESIGN OF
HYBRID VEHICLE POWERTRAINS



10 March 2020

THESIS PREPARATION

This thesis was prepared in the framework of a 3 years CIFRE contract between Groupe PSA and IFSTTAR.

During this period, Bilal Kabalan was employed as an engineer and PhD candidate by Groupe PSA and was hosted 70% of the time in IFSTTAR.

From IFSTTAR, Rochdi Trigui was directing the thesis and Emmanuel Vinot was closely supervising the thesis and offering the necessary support. From Groupe PSA, Yuan Cheng was the supervisor of the thesis and Clément Dumand had supervision shares while being the manager of Yuan and Bilal's team.

This work could not have been done without the contributions of all these people. Special thanks for everyone.

Thanks also to the committee members who accepted to examine this work and are contributing in improving it: Theo Hofman, Daniela Chrenko, Ali Sari and Betty Semail.

Advanced Research Department
Centre Technique de Vélizy A
Route de Gisy, 78140 Vélizy-Villacoublay, France

IFSTTAR / AME / ECO7
25 Avenue François Mitterrand, 69500 Bron, France

ABSTRACT

Systematic methodology for generation and design of hybrid vehicle powertrains

To meet the vehicle fleet-wide average CO2 targets, the stringent pollutant emissions standards, and the clients' new demands, the automakers realized the inevitable need to offer more hybrid and electric powertrains. Designing a hybrid powertrain remains however a complex task. It is an intricate system involving numerous variables that are spread over different levels: architecture, component technologies, sizing, and control. The industry lacks frameworks or tools that help in exploring the entire design space and in finding the global optimal solution on all these levels. This thesis proposes a systematic methodology that tries to answer a part of this need. Starting from a set of chosen components, the methodology automatically generates all the possible graphs of architectures using constraint-programming techniques. A tailored representation is developed to picture these graphs. The gearbox elements (clutches, synchronizer units) are represented with a level of details appropriate to generate the new-trend dedicated hybrid gearboxes, without making the problem too complex. The graphs are then transformed into other types of representation: 0ABC Table (describing the mechanical connections between the components), Modes Table (describing the available modes in the architectures) and Modes Table + (describing for each available mode the global efficiency and ratio of the power flow between all the components). Based on these representations, the architectures are filtered and the most promising ones are selected. They are automatically assessed and optimized using a general hybrid model specifically developed to calculate the performance and fuel consumption of all the generated architectures. This model is inserted inside a bi-level optimization process: Genetic Algorithm GA is used on the sizing and components level, while Dynamic Programming DP is used on the control level. A case study is performed and the capability of the methodology is proven. It succeeded in automatically generating all the graphs of possible architectures, and filtering dismissed architectures that were then proven not efficient. It also selected the most promising architectures for optimization. The results show that the proposed methodology succeeded in finding an architecture better than the ones proposed without the methodology (consumption about 5% lower).

Keywords: Powertrain design, hybrid electric vehicles, optimization, sizing, genetic algorithms, dynamic programming, constraint programming, architecture generation.

RESUMÉ

Méthodologie de génération systématique et de conception des chaines de traction de véhicules hybrides

Pour répondre aux objectifs de consommation des flottes de véhicules, au normes d'émissions de polluants et aux nouvelles demandes de l'usager, les constructeurs automobiles doivent développer des motorisations hybrides et électriques. Réaliser une chaine de traction hybride reste cependant une tâche difficile. Ces systèmes sont complexes et possèdent de nombreuses variables réparties sur différents niveaux : architecture, technologie des composants, dimensionnement et contrôle/commande. L'industrie manque encore d'environnements et d'outils pouvant aider à l'exploration de l'ensemble de l'espace de dimensionnement et à trouver la meilleure solution parmi tous ces niveaux. Cette thèse propose une méthodologie systématique pour répondre au moins partiellement à ce besoin. Partant d'un ensemble de composants, cette méthodologie permet de générer automatiquement tous les graphes d'architectures possibles en utilisant la technique de programmation par contraintes. Une représentation dédiée est développée pour visualiser ces graphes. Les éléments de boites de vitesse (embrayages, synchroniseurs) sont représentés avec un niveau de détails approprié pour générer de nouvelles transmission mécaniques sans trop complexifier le problème. Les graphes obtenus sont ensuite transformés en d'autres types de représentation : 0ABC Table (décrivant les connections mécaniques entre les composants), Modes Table (décrivant les modes de fonctionnement disponibles dans les architectures) et Modes Table + (décrivant pour chaque mode le rendement et le rapport de réduction global des chemins de transfert de l'énergie entre tous les composants). Sur la base de cette représentation, les nombreuses architectures générées sont filtrées et seules les plus prometteuses sont sélectionnées. Elles sont ensuite automatiquement évaluées et optimisées avec un modèle général spécifiquement développé pour calculer les performances et la consommation de toute les architectures générées. Ce modèle est inséré dans un processus d'optimisation à deux niveaux ; un algorithme génétique GA est utilisé pour le dimensionnement des composants et la programmation dynamique est utilisée au niveau contrôle (gestion de l'énergie) du système. Un cas d'étude est ensuite réalisé pour montrer le potentiel de cette méthodologie. Nous générons ainsi automatiquement toutes les architectures qui incluent un ensemble de composants défini à l'avance, et le filtrage automatique élimine les architectures présupposées non efficaces et sélectionnent les plus prometteuses pour l'optimisation. Les résultats montrent que la méthodologie proposée permet d'aboutir à une architecture meilleure (consommation diminuée de 5%) que celles imaginées de prime abord (en dehors de toute méthodologie).

Mots clé: dimensionnement de chaine de traction, véhicules électriques hybrides, optimisation, algorithmes génétiques, programmation dynamique, programmation par contraintes, génération automatique d'architectures.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Abstract

Conclusion

Abstract - In the first section of this introductory chapter, a global view is made on the transport sector and its today's changes. Details are given on the energy and environment preoccupations that are currently faced and how they are affecting the emission standards and legislations, the automobile manufacturers' strategies, and the cars market shares per type of powertrain. In the chapter's second section, the hybrid electric powertrains are presented and compared to the conventional and electric powertrains. The design problem of the hybrid electric powertrains is then introduced. It will be the genesis of the research done and the starting point for the other chapters.

1.1 CONTEXT

1.1.1 The transport sector in motion

The transport sector is evolving to comply with the climate and air quality targets and to meet the needs of today's society. The automotive industry is facing never seen before changes [1]. Car manufacturers are announcing in their strategic plans [2][3][4] increasingly actions on 3 main axis:

(A) Electrification and hybridization
(B) ADAS* and autonomous-drive (*Advanced driver-assistance systems)
(C) Car connectivity and new mobility services

The reason behind those actions is to solve the energy and environment issues that the transportation is causing, to answer the society new demands, and to follow the revolution in lifestyles (hi-tech dependency, connected devices,…).

This work is within the context of the axis (A) Electrification and hybridization, the axis that is tightly linked to the energy and environment topics. Axes (B) and (C) are out of scope of this work.

1.1.2 The energy and environment preoccupations

a) The actual situation

Global CO_2 emissions:

The CO_2 emissions on the worldwide scale keeps increasing [5]. The CO_2 emissions in the world, in China, India, USA and the EU are shown in Figure 1. This is the data of the IEA International Energy Agency. From 2000 to 2017, the USA and EU succeeded in decreasing their CO_2 emissions from 100% to 83% and 85% respectively. On the other hand, China and India emissions drastically increased from 100% to 296% and 244% respectively. On the global scale, the CO_2 emissions in 2017 are 141% of the global emissions in 2000. There is a need to decrease this number for a sustainable energy future and in order to limit the global temperature rise. This was the core of the 2015 Paris Agreement on climate change COP21 [6].

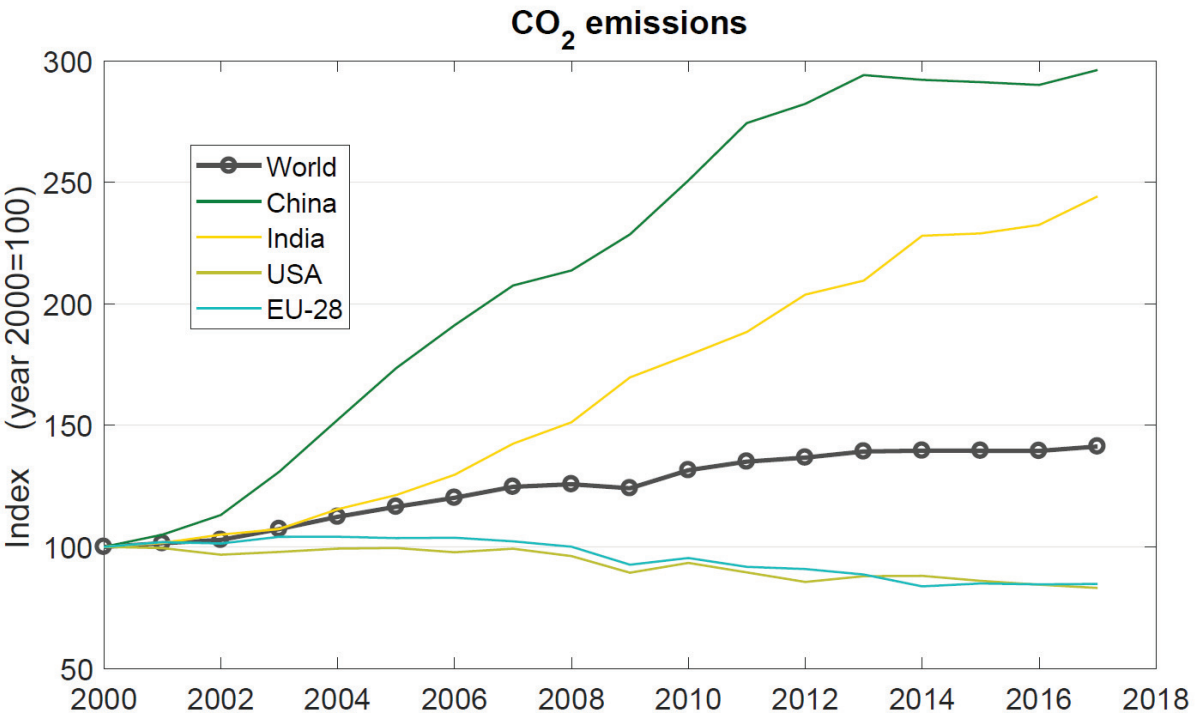


Figure 1: CO_2 emissions, data from IEA International Energy Agency 2018 [5]

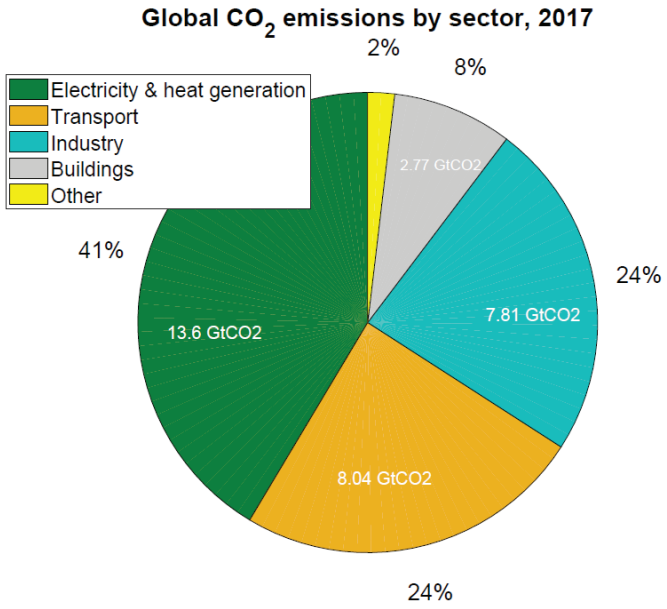


Figure 2: Sectoral disaggregation of 2017 global CO_2 emissions, data from IEA International Energy Agency 2018 [5]

In Figure 2, a sectorial disaggregation is done for the 2017 global CO_2 emissions. It can be seen that the transport sector is responsible of 24% of these CO_2 emissions. The major part of the transport CO_2 emissions are caused by passenger cars and light commercial vehicles which would be responsible of 72.9% of the total transport emissions in EU in 2017, according to [7].

Improving the CO_2 emissions of passenger cars and light commercial vehicles is inevitably a lever to decrease the transport CO_2 emissions and hence the global CO_2 emissions.

It should be mentioned here that on the world scale the electricity and heat generation are responsible of 41% of the global CO_2 emissions (Figure 2). In countries relying on renewable energies and nuclear power, the production of electricity would be responsible of less CO_2 emissions.

Local pollutant emissions:

Passenger cars and light commercial vehicles have also a role in improving the local air quality of cities by decreasing their pollutants emission. According to [8], the transport sector would be responsible of 63% of the NOx and 14 to 17% of PM (particulate matter) emitted in the cities, case of France in 2017 (Figure 3).

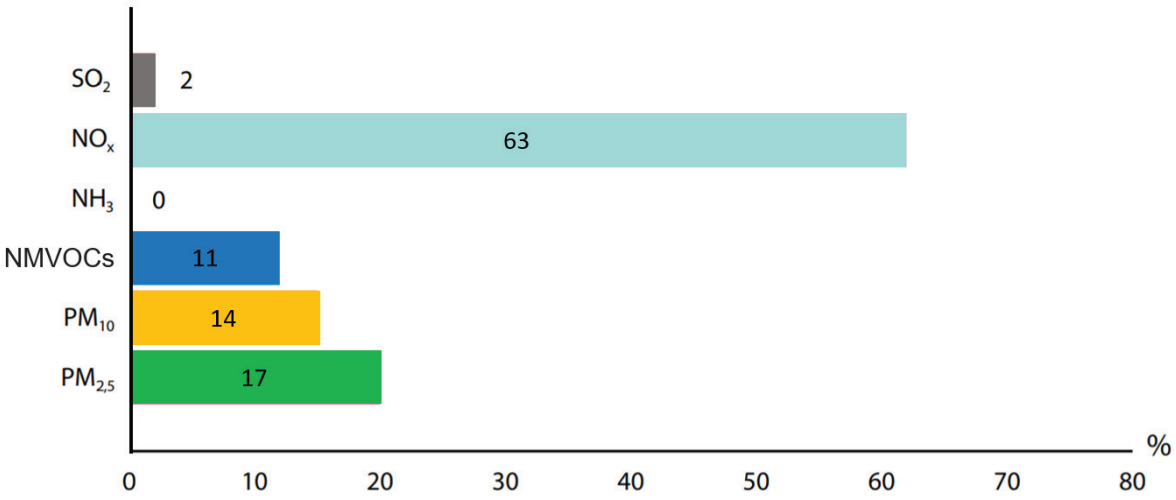


Figure 3: Percentage of pollutants emitted by the transport sector, France 2017 [8]

Electric driving in cities can reduce to zero these pollutants emission values excepting for the particulates emitted by tires and brakes.

b) The evolution in the emission standards and the legislations

Global CO_2 emissions:

In an effort to decrease the global CO_2 emissions and energy usage, legislations and governments set in place the Corporate Average Fuel Economy (CAFE) targets to decrease the average CO_2 emissions of the automaker's car fleets. This target is in g of CO_2 /km. The targets in USA, Europe and China are shown in Figure 4. It is clear how these targets are getting lower and lower over the years. In a recent decision, the EU commission made an objective of 37.5% reduction in 2030 compared to 2021 for passenger cars [9].

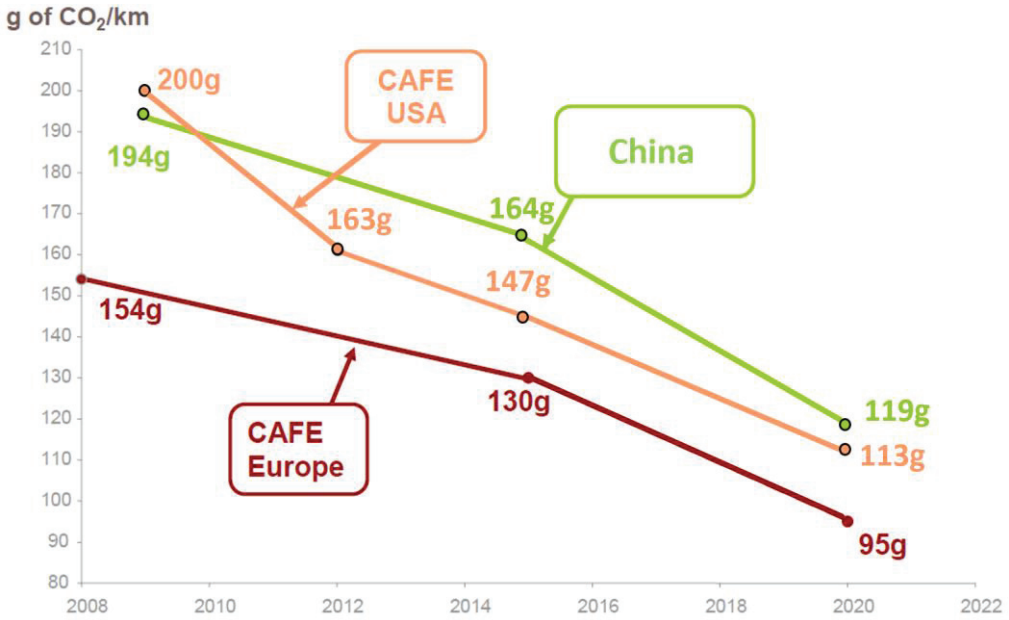


Figure 4: CAFE standards for CO2 emissions [10]

When the automaker fails to meet these objectives, a penalty is paid per vehicle for every gram of CO_2 /km above the target (95 € per vehicle and per gram in Europe). With a decrease in the sales of low CO_2 diesel vehicles bad marketed by the VW diesel Gate [11], reaching such low CO_2 targets cannot be done solely with conventional gasoline powertrains. Alternative powertrains are needed, electrification included.

Local pollutant emissions:

In parallel, on the local pollutant emission side, strict regulations are applied for vehicles homologation on the maximum tailpipe emissions [8]. An evolution of the normalized values of the pollutants is shown in Figure 5 to illustrate how stringent these regulations are becoming.

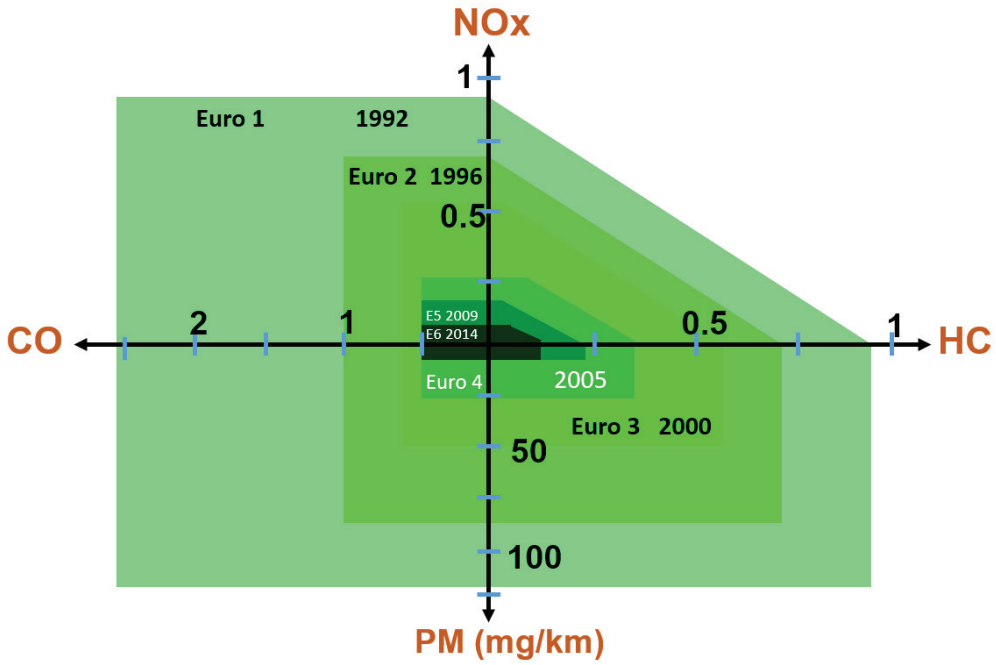


Figure 5: Evolution of the pollutants limits for diesel vehicles in g/km, PM in mg/km

City policies:

In addition, city governors are putting in place policies to enhance the air quality. Such policies include applying LEZ Low Emission Zones where some vehicles are banned to circulate or even ZEZ Zero Emission Zones like the one studied in [12] for Oxford city. The LEZ that are already in place in Europe are show in Figure 6 .

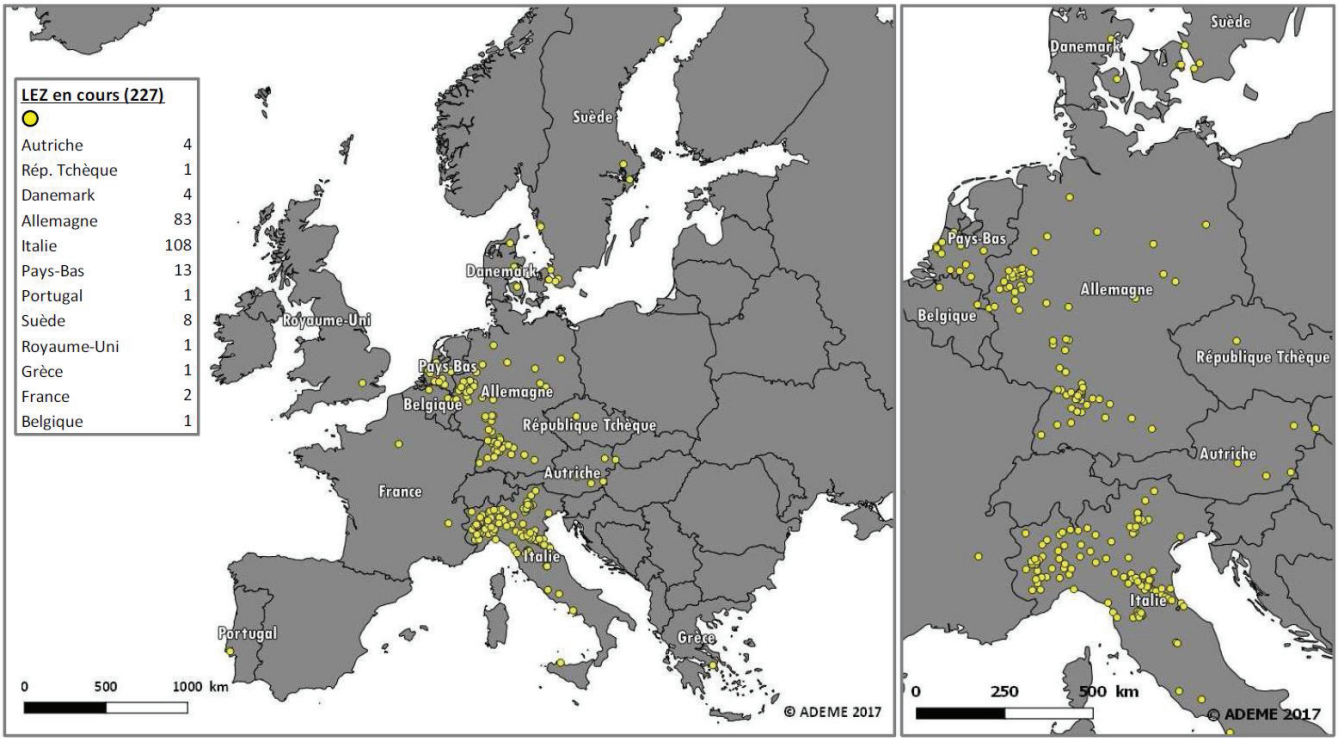


Figure 6: LEZ, ADEME 2017

In France, the CRIT'AIR Air Quality Certificate was put in place (Figure 7). It is a round sticker applied on the car glass and which corresponds to the vehicle class (1 until 5). It categorizes the vehicles according to their pollutants emissions. Restrictions can be applied on the categories allowed to access the LEZ. For example, category 5 has already been banned since May 2019 in the Grenoble-Alpes metropole LEZ. In this LEZ, only categories 1 and electric vehicles will be allowed to circulate in 2025 (Figure 7).

Figure 7: Grenoble-Alpes metropole action plan [13]

Another type of measures was to ban old diesel vehicles from circulating in cities (case of Paris [14]). Some governments also announced plans of banning the sales of conventional pure diesel / gasoline vehicles in the future, hybrids not included (UK and India for example). Other governments were stricter and announced that all new cars should have zero-emissions in the future (France, Norway and Netherlands for example). More details are given in Table 1 based on [15].

Table 1: Example of countries bans [15]

Country	Ban Starts	Scope
France	2040	End sales of vehicles that emit greenhouse gases [16]
Norway	2025	All new light vehicles, new city buses, and new light commercial vans should be zero-emissions vehicles [17]
Netherlands	2030	All new cars should be zero-emissions vehicles [18]
India	2030	End sales of petrol and diesel cars [19]
United Kingdom	2040	End conventional car and van sales (gasoline and diesel, the ban does not include hybrids) [20]

a) The evolution in the strategies and the technology choices of the automobile manufacturers

To meet the fleet average CO_2 targets, the stringent pollutant emissions standards, and the clients' new demands, automakers including Groupe PSA realized the inevitable need to develop alternative powertrains and to increase their offer to the clients. Battery Electric Vehicles (BEVs), Hybrid Electric Vehicles (HEVs), Plug-in (PHEVs), Fuel Cell Electric Vehicles (FCEV) are the main considered alternative powertrains.

A move in the course of vehicle hybridization and electrification was widely seen in the announced strategic plans of car manufacturers. Examples are shown in Table 2.

Table 2: Manufacturers electrification and hybridization plans

Manufacturer	Date	Plan
Groupe PSA	2016	7 BEV models to be lunched from 2019 to 2021 7 PHEV models to be lunched from 2019 to 2021 [2] [21]
	2018	The group will be 100% electrified in 2025 [22], meaning that all cars will at least be equipped with light hybridization
	2018	48V Mild HEV models to be lunched starting 2022 Production of 600 000 e-DCT / year (electrified dual clutch transmission with Punch Powertrains) for these Mild HEV models [23]
Renault-Nissan-Mitsubishi alliance	2017	12 new BEV models to be lunched by 2022 BEV will cover all main segments by 2022 in Japan, USA, China and Europe markets [3]
	2019	HEV and PHEV models for Clio, Captur and Megane (E-TECH) to be launched in 2020 [24]
VW Group	2017	Roadmap E: Entire model portfolio electrified by 2030, meaning that at least 1 electrified version of each of the 300 or so Group models [4]

b) The evolution in hybrids and electrics market shares

All the previous explained contents resulted in a market share increase for the hybrids and electrics powertrains. The evolutions in the millions of BEV and PHEV deployed in China, Europe, USA and the world from 2013 to 2018 are shown in Figure 8, based on the IEA IEA-International Energy Agency data [25]. From 2013 to 2018, the number of BEV and PHEV was roughly multiplied by 5.

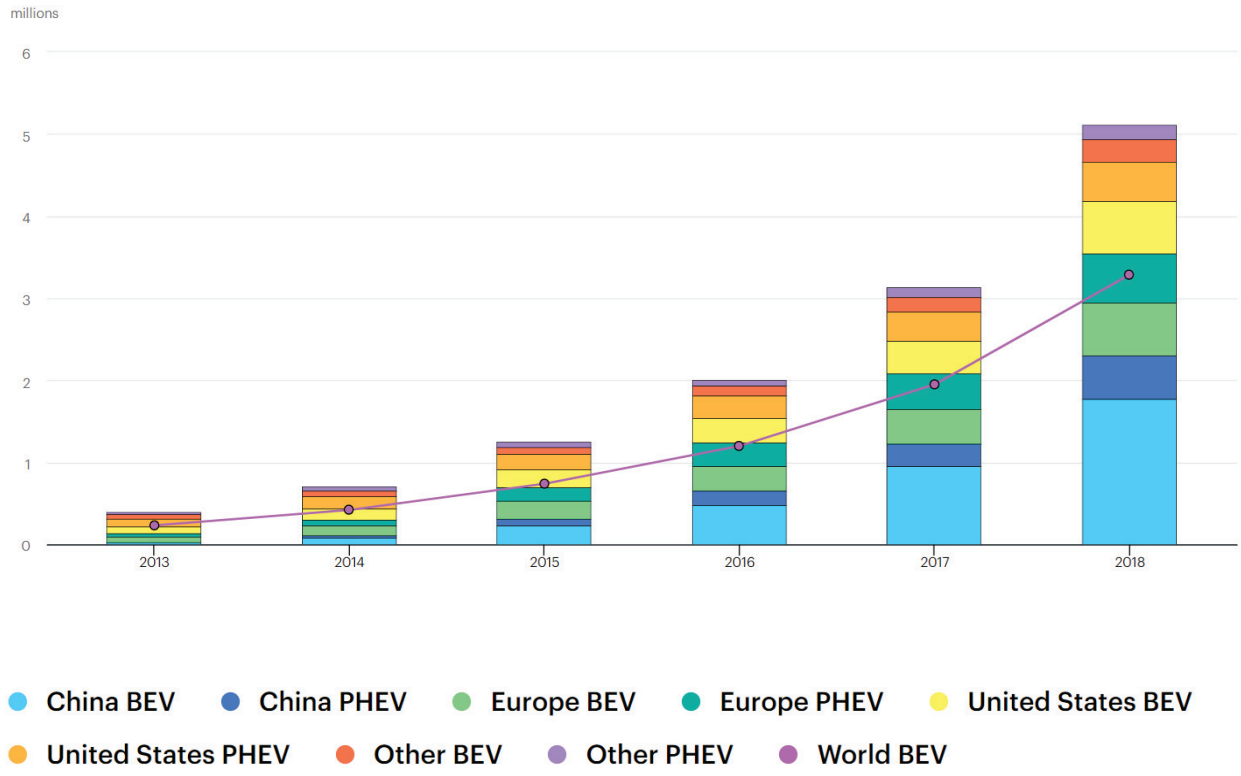
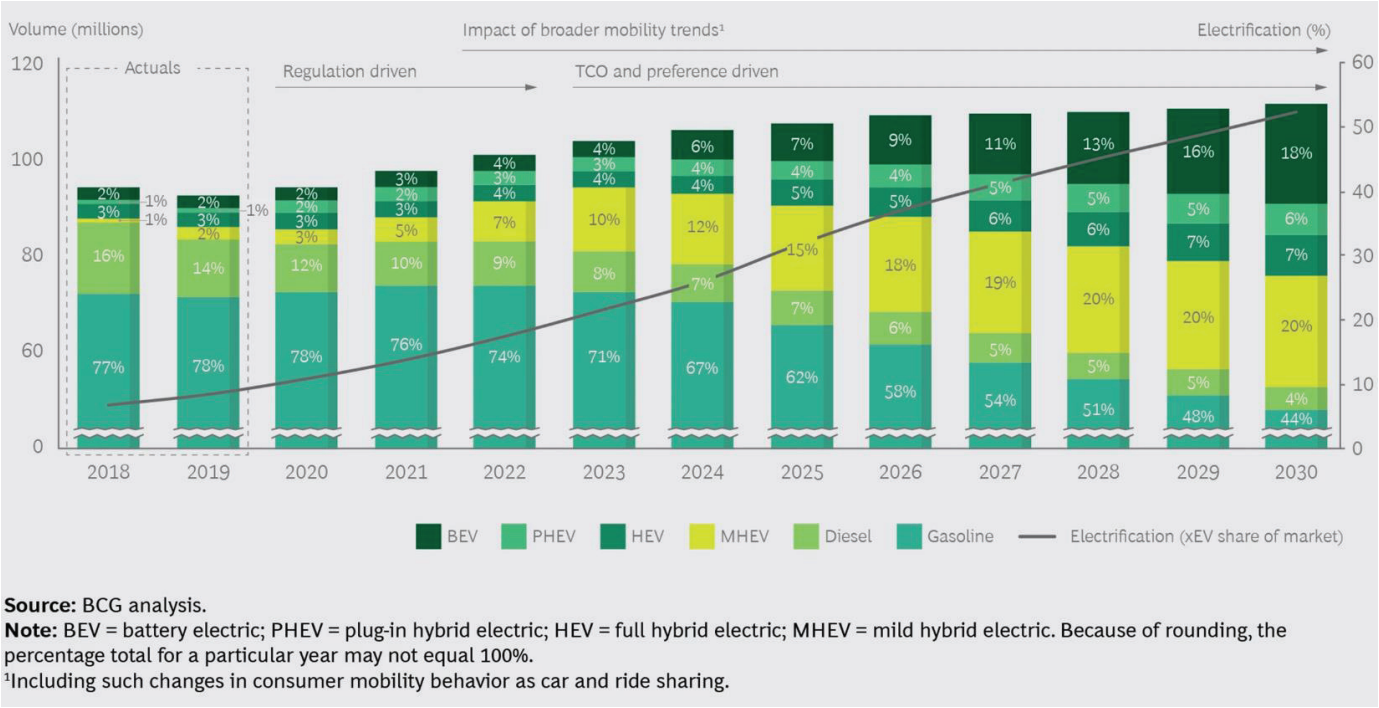


Figure 8: BEV and PHEV deployment [25]

Different industry consulting groups and market analysts already projected couple of years ago this important penetration of the electrified powertrain into the market. Now that their expectations were actually realized and faster than expected, their projections are now more optimistic. For instance, the Boston Consulting Group (BCG) recently updated its projections in January 2020 to confirm that the sales of electrified cars (xEVs) are growing faster than expected. In January 2018, they have published forecasts showing that sales of xEVs would be 1/4 of the market by 2025 and approaching 50% in 2030. In their January 2020 updated report, these forecasts are updated to 1/3 of the market in 2025 and 51% in 2030. These projections are shown in Figure 9. The major growth is expected for Mild HEV or 48V hybrids that represent today 2% of the market but will be 15% of the market in 2025, taking the part of pure conventional diesel and gasoline. BEVs that are today 2% of the market are expected to be 7% in 2025. PHEVs that are today 1% of the market are expected to be 4% in 2025. Full HEV are expected to grow from 3% to 5%.



Source: BCG analysis.
Note: BEV = battery electric; PHEV = plug-in hybrid electric; HEV = full hybrid electric; MHEV = mild hybrid electric. Because of rounding, the percentage total for a particular year may not equal 100%.
[1]Including such changes in consumer mobility behavior as car and ride sharing.

Figure 9: BCG forecasts in January 2020,

Volume(millions) is the yearly production volume in the world; Electrification = xEV = BEV + PHEV + HEV + MHEV ; TCO = total cost of ownership including purchase price (battery price included), maintenance cost and fuel/electricity cost

1.2 HYBRIDIZATION AND ELECTRIFICATION

In this section, the hybrid electric powertrains are explained. Their design problem is then introduced. The stated information is based on industry knowledge, technical courses handouts (LAU[26] and IFP School[27]) and literature reviews [28] [29].

1.2.1 Vehicle powertrains: Conventional vs Hybrid vs Electric

The powertrain of a vehicle is the group of components generating power and delivering it to the wheels in order to propel the vehicle. A comparison between conventional, battery electric and hybrid electric powertrains is done is Figure 10.

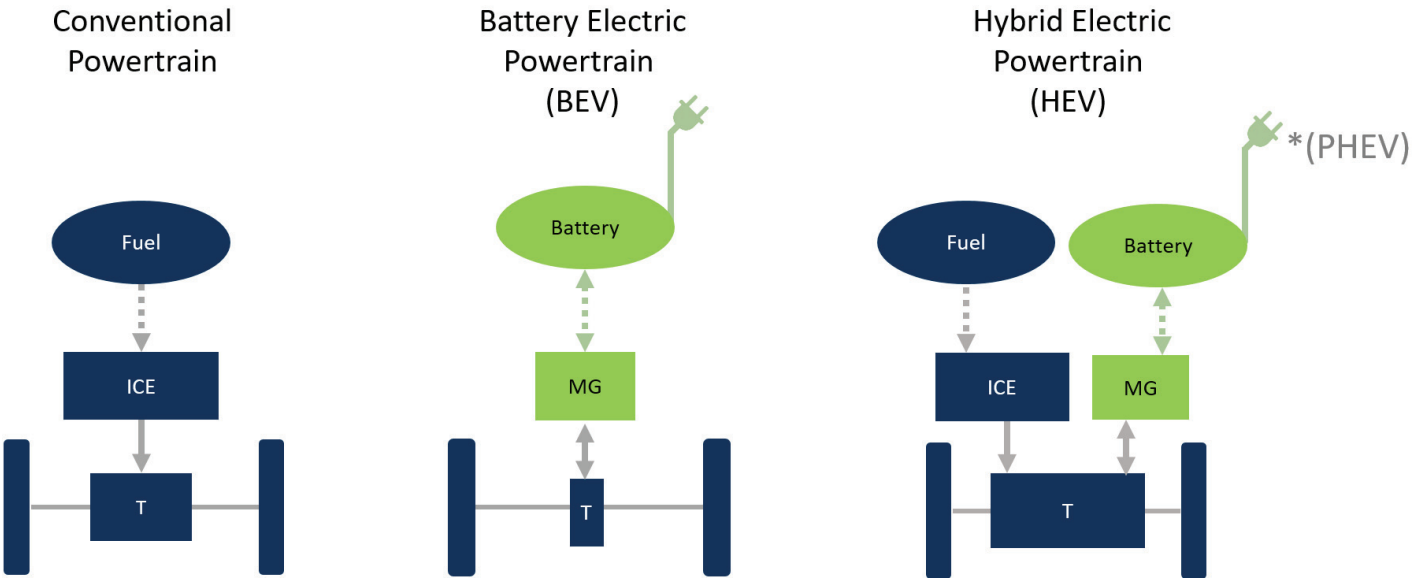


Figure 10: Comparison between conventional, electric and hybrid powertrains

In a conventional powertrain, the energy is stored in a tank that supplies the fuel (Gasoline, Diesel, others...) to the powertrain. The Internal Combustion Engine (ICE) transforms the fuel energy into mechanical power. This generated power is delivered to the wheels through a Transmission (T). It includes a clutch or a torque converter, a gearbox of different gear ratios, a final drive, and a differential. In braking, the energy is dissipated as heat in the braking discs, by friction.

In Battery Electric Vehicle (BEV) powertrains, the energy is stored in a battery (reversible energy storage system) that can be plugged for recharging and that supplies/receives the electric power to/from a Motor Generator (MG), i.e. an electric machine. In traction, the MG transforms the electrical power into mechanical power. This generated power is transferred to the wheels through a simplified Transmission (T). It includes generally a gear and a differential. In regenerative braking, the MG transforms the mechanical power into electrical power to recharge the battery. The braking energy is not completely lost.

A Hybrid Electric Vehicle (HEV) powertrain combines the conventional engine-based traction system with a battery electric traction system. If the battery can be recharged by plugging the vehicle into a charge point, the powertrain is referred to as Plug-in HEV or PHEV. In the following, (P)HEV will be used to refer to HEV and PHEV. The battery electric system can be sized to be minimally present or significantly present, compared to the engine-based system. This

leads to different hybridization ratios or different levels of hybridization for (P)HEVs. This will be explained in section 1.2.2. In addition, the two systems can be connected through different means, resulting in various (P)HEV architectures. This will be shown in section 1.2.3.

Different challenges come with the electrified powertrains. BEVs have a relatively simple powertrain, but their main challenge is on the battery side (autonomy, cost, charging time, charging infrastructure...). (P)HEVs are challenged by the complexity of the powertrain, its cost and its complicated control. The design of a (P)HEV is a complex problem involving numerous variables, constraints and objectives. This will be explained in section 1.2.4.

1.2.2. Different levels of hybridization

Depending on the sizing of the electric traction system compared to the engine-based system, (P)HEVs can have different levels of hybridization. This is presented in Figure 11.

To the left is placed the conventional powertrain, to the right the electric powertrain with the example of the 2019 Peugeot e-208 and the 2012 Renault Zoe. The different levels of hybridization are placed in order from left to right: micro hybrid, mild hybrid, full hybrid and Plug-in & Range extender.

In micro hybrids, the MG power is couples of kW (x KW) and is capable of doing automatic Stop & Start to stop the engine when unneeded. In some micro hybrids, the MG can also recuperate some energy by doing regenerative braking and recharging the battery. The 2011 Peugeot 308-eHDI is an example.

If the MG can provide torque to the wheels, assisting the ICE in vehicle propulsion, this is called a boost mode option and the powertrain is named mild hybrid. It is a term invented by Honda. An example of mild hybrids is the 2009 Honda Insight.

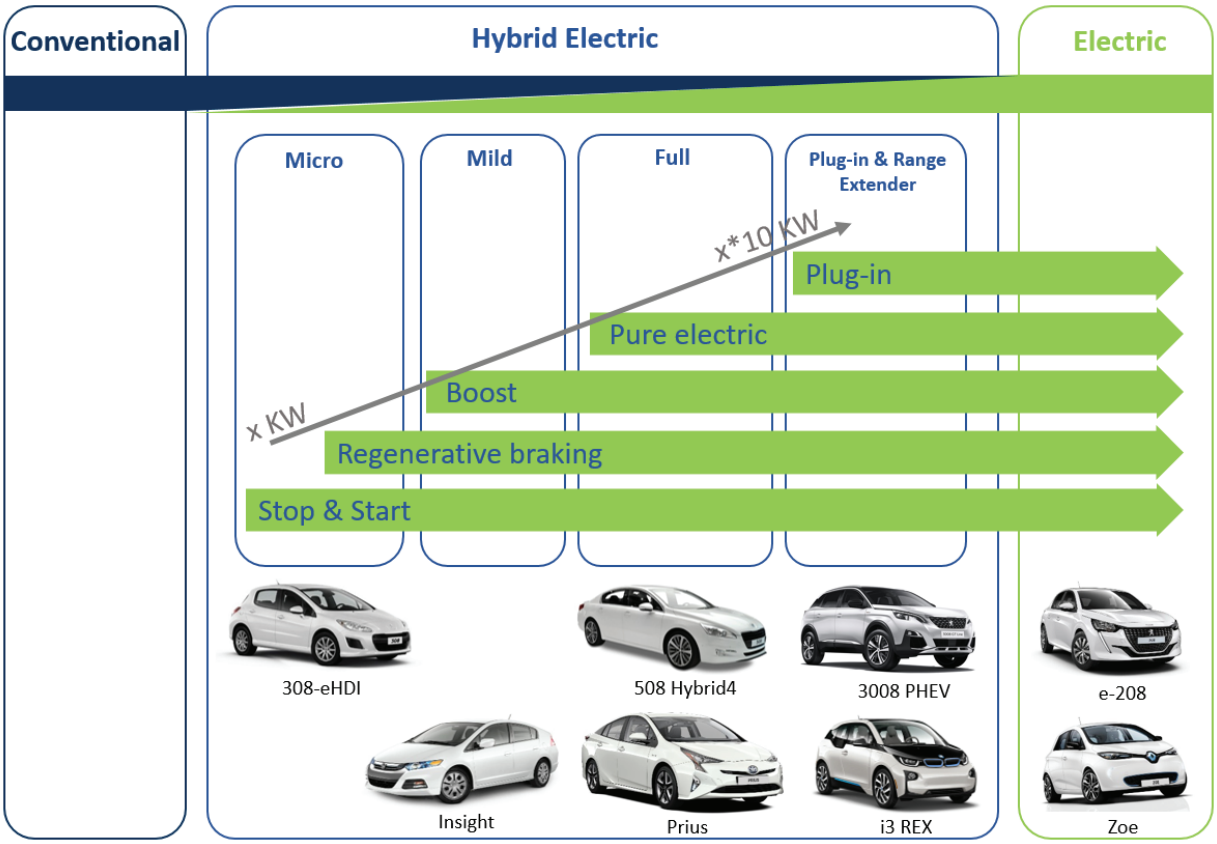


Figure 11: Different levels of hybridization

If the MG can solely provide all the power needed by the wheels and can therefore drive the vehicle in pure electric mode, the powertrain is referred to as full hybrid. The shown examples of the full hybrids are the 2013 Peugeot 508 Hybrid4 and the 2016 Toyota Prius. The MG power is couples of 10kW (x*10 KW).

Finally if the powertrain battery can be plugged in for recharging, this is the category of Plug-in and range extenders. The MG power can be in the order of magnitude of 100kW (for a vehicle of 1 to 1.5 tones). The 2019 Peugeot 3008 PHEV and the 2013 BMW i3 REX are examples.

When compared to a conventional diesel/gasoline vehicle, hybrid electric vehicles offer a reduction in CO2 emission. According to IFSTTAR figures in urban driving, this reduction can have an order of magnitude of 5% to 10% for micro hybrids. It can reach 20% to 40% for full hybrids and even more for plug-in depending on their electric autonomy, until reaching 0g CO2 emitted during use, in the case of full electrics.

1.2.3. Different architectures

(P)HEV powertrains combine a battery electric traction system with a conventional engine-based traction system. The two systems can be connected through different means, resulting in various (P)HEV powertrain architectures. Series, parallel and series-parallel (power-split and non-power-split) are the main categories of existing hybrid architectures [29]. Simplifying diagrams showing the difference between these architectures are shown in Figure 12.

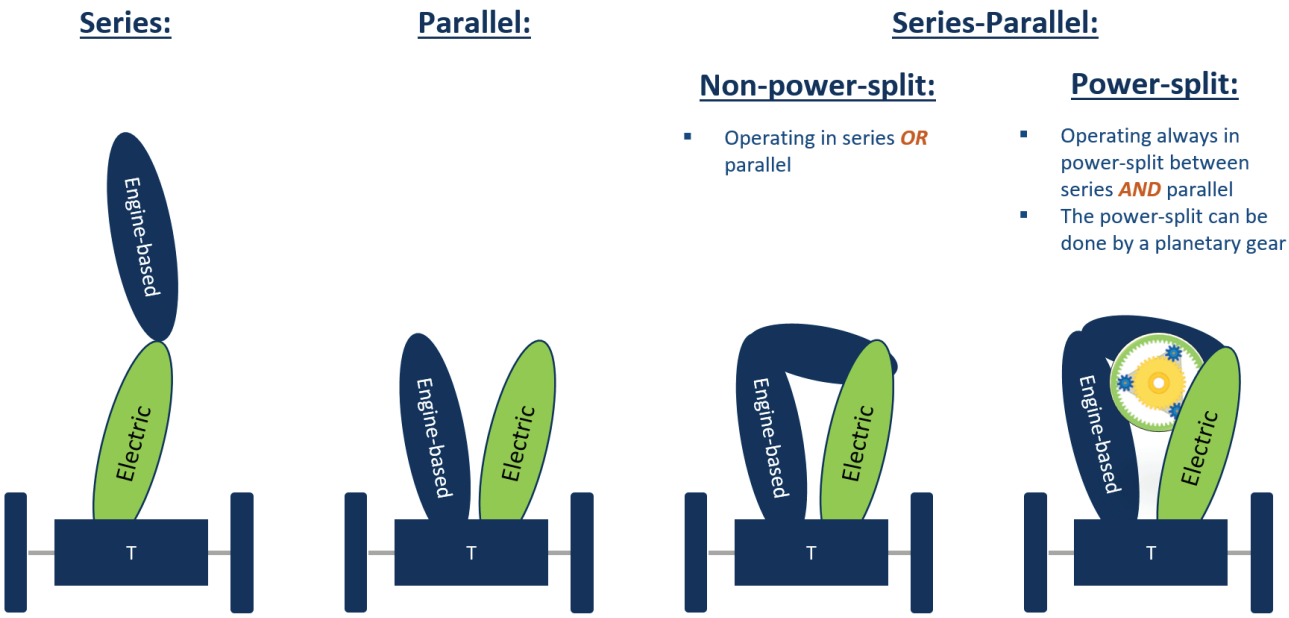


Figure 12: The different existing hybrid architectures

- **Series:**

In the series architecture, the engine-based system is not mechanically connected to the wheels and does not provide any mechanical power to the wheels. It backs the electric system which is alone in connection with the wheels.

This architecture is shown in more details in Figure 13. The ICE is mechanically connected to an electric machine MG2 acting as a generator. Together they constitute a unit generating

electrical power to the system and is referred to as Auxiliary Power Unit (APU). A different electric machine MG1 is mechanically connected to the wheels through a simple transmission. This electric machine is solely propelling the vehicle in traction and recuperating the energy in braking, same as in an pure electric powertrain.

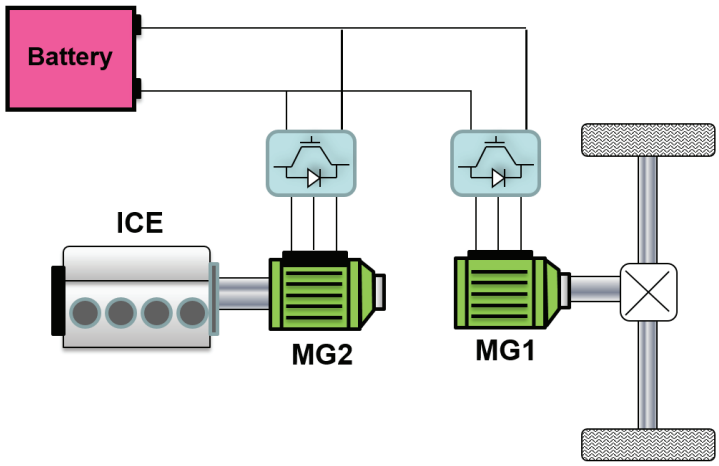


Figure 13: Series architecture

The decoupling between the engine and the wheels allows the engine to operate on its optimal operating line in function of the electrical power asked from the APU. This decoupling also allows the APU location in the vehicle to be flexible.

The disadvantages of this architecture is the relatively low efficiency of the energy path from the engine to the wheels, caused by the cascade arrangement. This low path efficiency is countering the good efficiency offered to the engine. A second disadvantage is that all the traction components need to be sized for the maximum power demand of the wheels because the power to the wheels is not summed in the transmission. All the power going to the wheels is obliged to take one mechanical path instead of two (as in other architectures).

- **Parallel:**

In the parallel architecure, the engine-based and the electric system are both mechanically connected to the wheels.

One electric machine MG is required. The MG and ICE can alone or together propel the vehicle. In function of MG location in the powertrain, different parallel architectures exist (Figure 14) and are identified in the industry as:

- P0 or P1f: MG as a belt-driven position
- P1 or P1r: MG cranckshaft mounted
- P2: MG upstream of the gearbox
- P3: MG downstream of the gearbox
- P4: MG on the other axle

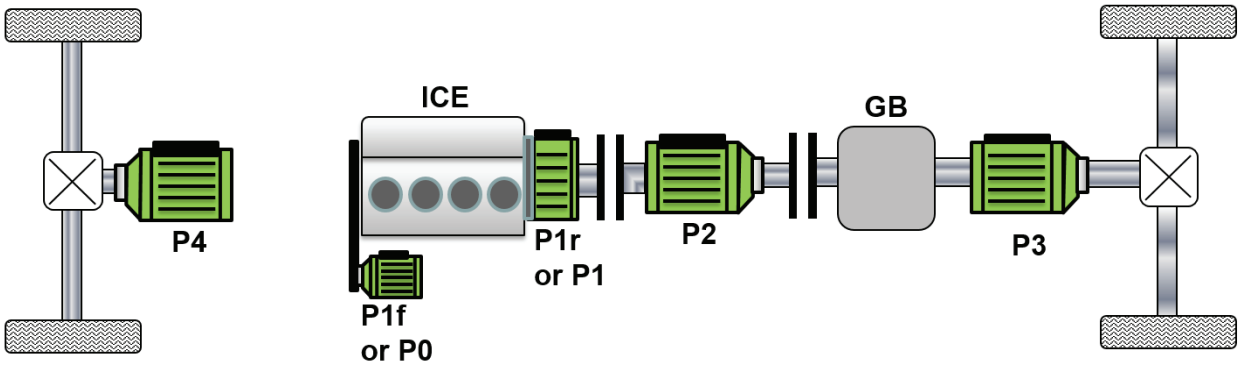


Figure 14: Mechanical connection in the different parallel architectures

The ICE has a torque degree of freedom (DoF) but does not have a continuous speed DoF because its speed is directly imposed by the wheels and the gearbox, once the gear is chosen. The gear choice is a DoF with discrete values and it affects the engine speed. Exepting the use of a CVT (continuous variable transmission), operating the ICE in its high efficiency zone is more difficult than in the case of series. On the other hand, the power paths here have better efficiencies than in the series case.

- **Series-Parallal, power-split:**

This architecture operates always in a power split between the series and the parallel paths. This power split is achieved by a planetary gearset (PG).

The Toyota Prius was the first adopter of this architecture in the Toyota Hybrid System (THS) [30], (Figure 15). Chevrolet Volt and Opel Ampera also adopted a power-split architecture [31].

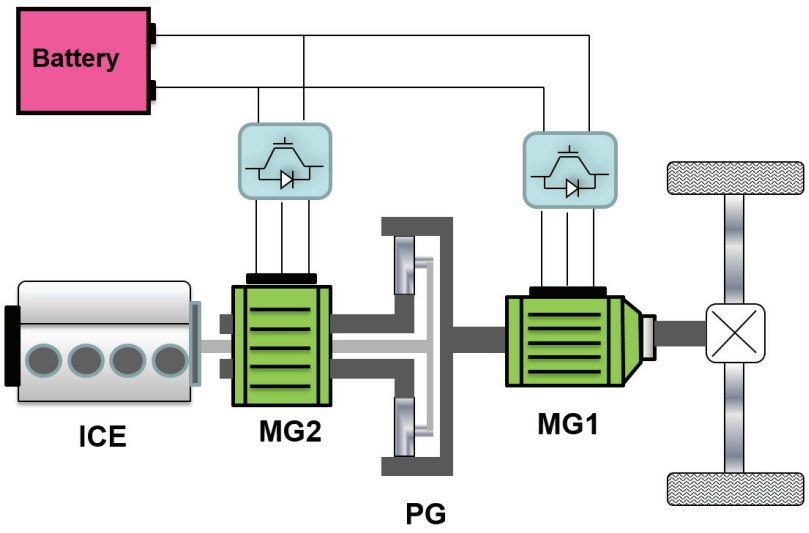


Figure 15: THS power-split architecture

In power-split architectures, the PG decouples the ICE speed from the wheels speed, allowing a speed DoF, in addition to the torque DoF that is present in any parallel architecture. This helps in moving the operating points of the ICE on its high efficiency area. The cost of this double freedom is that the power going to the wheels is always split between a parallel path and a series path, the latter having a lower efficiency due to the added energy conversion stages. Still, power-split SPHEV remain ones of the most efficient mass produced HEV [32], [33] and the THS is widely used as a reference powertrain.

- **Series-Parallal, non-power-split:**

This architecture has the components allowing it to operate in series or parallel modes (Figure 16). In contrast to power-split, the powertrain does not operate in power-split between series and parallel modes. The shifting between the series and hybrid modes is done throught changes in the states of the clutches.

Its advantage is the possibility to operate in pure parallel mode, avoiding the losses of the series path. In addition, it has a simpler design and control than power-split. Its disadvantage is that the speed DoF is only available in the series mode. It is then more difficult to operate the system in its best efficiency areas compared to power-split. In fact, the vehicle speed and torque demands might constraint the powertrain to operate sometimes in series mode even when it has relatively low efficiency

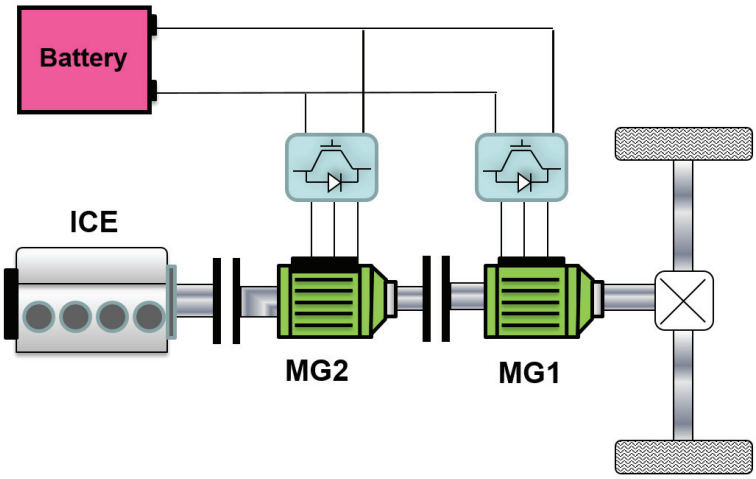


Figure 16: A simple series-parallel architecture

These are the four main categories of existing hybrid architectures. Multiple variants can be created in each category by the addition or elimination of clutches, gears or gearboxes, and by the change of location of the components.

1.2.4. Introduction to the design problem of (P)HEV

The design problem of (P)HEV that engineers need to solve is illustrated in Figure 17.

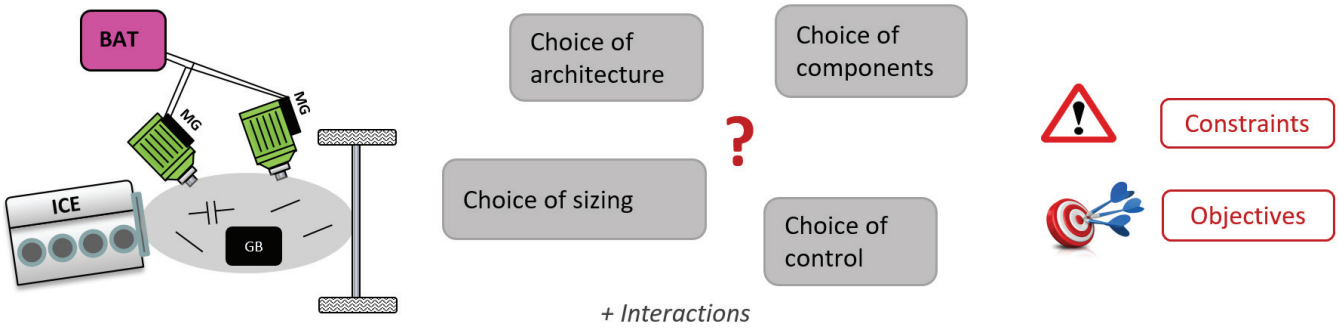


Figure 17: The design problem of (P)HEV that engineers need to solve

Considering the (P)HEV powertrain as a combination of powertrain components (ICE, MGs, BAT, wheels) and connecting elements (shafts, clutches, gears, gearbox,…) as shown in the left of Figure 17, the design engineer needs to choose the architecture or the topology linking these components. In addition, a choice of the components technologies is also required (engine family,

electric machines type, battery technology…). Moreover, different sizing can be made (battery size, components power, gear ratios). The control of the components is also to be decided.

The engineer faces all these variables to decide while ensuring that the powertrain always respects some constraints and while trying to optimize some powertrain objectives. A compromise is to be done between several conflicting considerations: dynamic performance, fuel consumption, cost and compactness of the powertrain, drivability, etc.

This is a complex problem in the industry. It will be explained in more details in the coming chapter. This PhD work proposes a part of the answer to this problem.

CHAPTER 2: THE DESIGN PROBLEM OF (P)HEV

Abstract - This second chapter states the problem, overviews the state of the art, and introduces the authors' proposed methodology to solve it. First, the (P)HEV design problem is explained from an industrial perspective. Then it is drawn in the optimization context where it spreads on multiple levels related to design or control. Afterwards, the state of the art is overviewed: the methods used on the control level, the methods used on the design level, the coordination approaches between the two levels, and the works that started in the architecture exploration. Finally, the methodology that is proposed to solve the entire problem is introduced. The detailed explanation of the different parts of this methodology will be explained in chapter 3 and 4.

2.1. (P)HEV Design Problem

2.1.1 The industrial need

In any vehicle design project, the development of the powertrain follows the V cycle design process that is widely practiced in the automotive industry [34]. This V cycle is shown in Figure 18 where the scope of intervention of the PhD is also positioned.

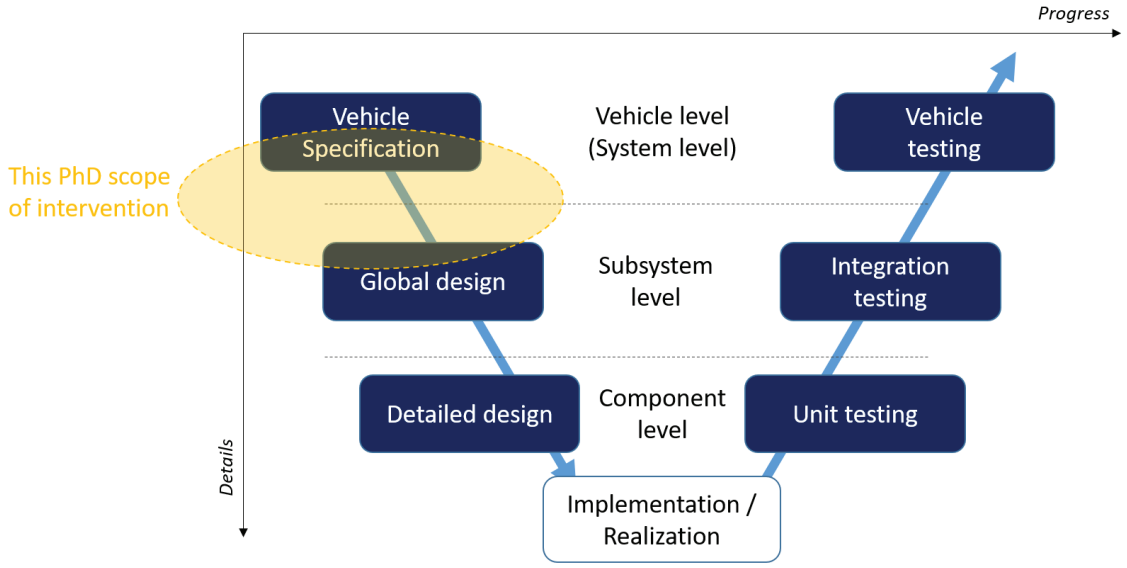


Figure 18: The V cycle for vehicle design projects

The design process starts from a definition of the vehicle specifications. These can be the vehicle segment (B, C, SUV...), the performance requirements, the fuel consumption and emission targets, and others. The design engineers need to choose the powertrain variables that meet all the specifications. In this pre-design phase and on this high vehicle level, numerical simulations are used to assess and optimize the proposed designs. For the performance, fuel consumption and emissions calculation, quasi static energetic models are normally used.

In conventional gasoline or diesel powertrains, the powertrain variables are the engine family and size, the type of gearbox (manual, automatic, automated manual, CVT,..), the number of gears and the gear ratios. In development projects, the choice of engine family and size is restricted to the car manufacturer's available engines; the gearbox is constraint by the traditional layout (Front, Rear or All Wheel Drive: FWD, RWD, AWD), technologies used and supplier constraints. The wider freedom is in the choice of the gear numbers and gear ratios. The first gear is calculated from the takeoff acceleration requirement. The last gear is calculated from the

maximum speed requirement, and equal to the ratio that coincide the maximum vehicle speed on the engine speed where its maximum power is reached. This process is well known in the industry and is not very intricate.

In Battery Electric Vehicles, the gearbox is generally replaced by a single gear ratio and the battery technology choice and size are added. The number of design variables is reduced and the design problem remains not complicated.

The complexity of the design problem arise in the hybrid systems in general, (P)HEV in the case of this work. Since at least two traction systems exist, new dimensions are added to the problem due to the added flexibility or degree of freedom in the system.

The first added dimension is the way the connection between the two traction systems is done; this is called the '**architecture**'. In fact, (P)HEV powertrains combine a battery electric traction system with a conventional engine-based traction system. The two systems can be connected through different means, resulting in various (P)HEV powertrain architectures. As explained in the previous chapter, series, parallel and series-parallel (power-split and non-power-split) are the main categories of existing hybrid architectures [29].

Once the architecture is chosen, different '**components technologies'** can be selected, and different '**sizing'** can be made (battery size, components power, gear ratios). The powertrain operation and fuel consumption on a selected driving cycle will depend on the architecture chosen, on the components chosen, on their sizing, and lastly on the energy management or **'control'** during the vehicle operation. A strong interaction exist between these different variables.

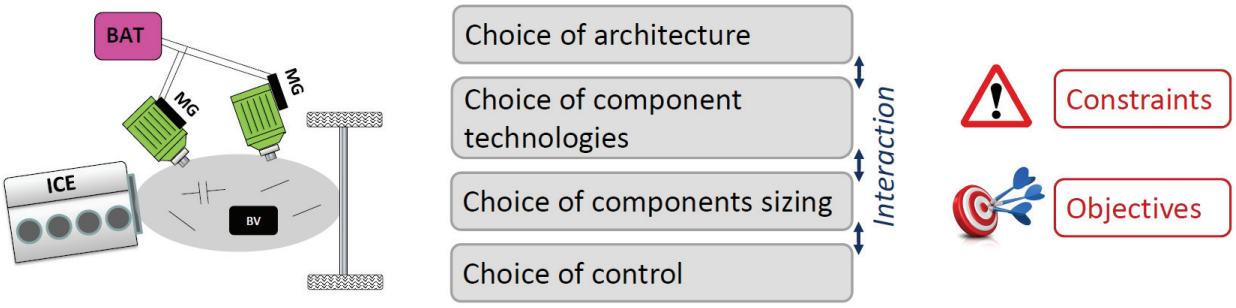


Figure 19: (P)HEV design problem

When deciding these variables, design engineers should hence look to the system globally. They need to respect the constraints coming from the component level (maximum speed and

torque of components for example) and from the system level (maximum vehicle speed, minimum acceleration for example). The design process also has objectives to optimize. These can be the fuel consumption or emissions on a driving cycle, the powertrain cost or compactness.

Handling all these variables, constraints and objectives becomes a complex task for the design engineers in the case of (P)HEV. Therefore, there is a need to create a methodology that can be used in the pre-design phase and that can help design engineers in making those decisions in function of the manufacturer (here Groupe PSA) objectives / constraints and the available components. This was the industrial need that initiated this PhD,

2.1.2 The system design problem in its optimization context and its spread on multiple levels

In this subsection, the problem is formulated in its optimization context. The introduced industrial problem in 2.1.1 can be stated as a multi-objective optimization problem that is spread over multiple levels [35] [36] and involving a system, variables, constraints and objectives.

- The considered **System**: the (P)HEV vehicle.

- The involved **Variables (x)**: can be divided into 3 levels (Figure 20): (1) Architecture level, (2) Components technology and sizing level, (3) Control level.

$$x = [\, x_{Architecture}\,,\, x_{Components}\,,\, x_{Sizing}\,,\, x_{Control}\,(t)\,]$$

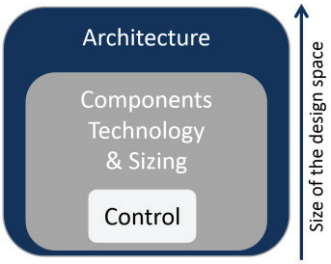


Figure 20: (P)HEV powertrain design space

  - $x_{Architecture}$ can be: the type of architecture (series, parallel, power-split…), presence or not of a gearbox, location of the gears, location of the EMs, presence or not of clutches, number of gears, ….
  - $x_{Components}$ can be: the type of engine, EMs, battery cells, …

- x_{Sizing} can be: number of battery cells, maximum power of the ICE, maximum power of EMs, gear ratios in the gearbox, ratios of the components gears, final drive ratio…
- $x_{Control}$ (t): these are not design variables. They are control variables that should be decided in function of time when the powertrain is operated. They can be the choice of operating mode (pure electric, parallel hybrid, series hybrid…), choice of gear ratio, choice of power split between the components…

- The **Constraints**: these are the limits coming from the components level or the system level and that need to be respected in any design candidate.
  - Components: minimum and maximum speed, minimum and maximum torque, minimum and maximum voltage and current for the battery, SOC limits…
  - System: performance requirement
    - Maximum speed Vmax > *value*
    - Acceleration time 0-100km/h or 80-120km/h < *value*
    - Minimum electric range > *value*
    - ...

- The **Objectives (J)**: different objectives have been considered in the literature, they can be divided into 2 groups
  - Design related objectives: can be calculated when the design candidate is chosen without the need to operate the vehicle. These can be the total cost of the powertrain (summation of components costs), total volume or weight,..

$$J = f\left(x_{Architecture}, x_{Components}, x_{Sizing}\right)$$

  - Design and control related objectives: can not be calculated without simulating the vehicle operation on a chosen driving cycle or on some chosen operating points. For instance: the integral of the fuel consumption or pollutants emission or battery aging at the end of the driving cycle.

$$J = f\left(x_{Architecture}, x_{Components}, x_{Sizing}, x_{Control}(t)\right)$$

These objectives can be in contradiction. Therefore, most of the time more than one objective are considered. This means that a multi-objective optimization algorithm is required. Another

possibility is to reformulate the objective function as a weighted sum of different objectives. A special care would need to be done on the choice of the weights multiplying the objectives.

This was a simplified but global description of the optimization problem behind the (P)HEV design problem. As it can be seen, the design space of this problem is large and spread on different levels. A strong interaction exist also between these levels. A global optimization is required to reach the optimal solution. Fixing a level and optimizing the others implies a sub-optimal solution. In addition, the resolution of the control problem can be complex and time consuming. This being said, simulating and assessing all the combinations of design and control (brute force or exhaustive search) is not time feasible. For that reason, several optimization methodologies have been developed and will be screened hereafter.

2.2. State of the Art

Various works are done in the literature in the context of the (P)HEV optimization and several methodologies have been developed to solve the previously explained problem. A clear and comprehensive analysis of these methodologies was done in [35], which is a main reference for this chapter.

Traditionally, the works done in the literature tackle the problem by doing an optimization on level 2 (Components technology & sizing) and 3 (Control) with less effort on including the level of architecture in the optimization: few benchmark architectures are arbitrary selected, optimized on the two levels (2) and (3), and then compared (Figure 21). Examples of these works will be shown in section 2.2.3. The different optimization methods used on each level and the coordination approaches between them are explained in detail in the following.

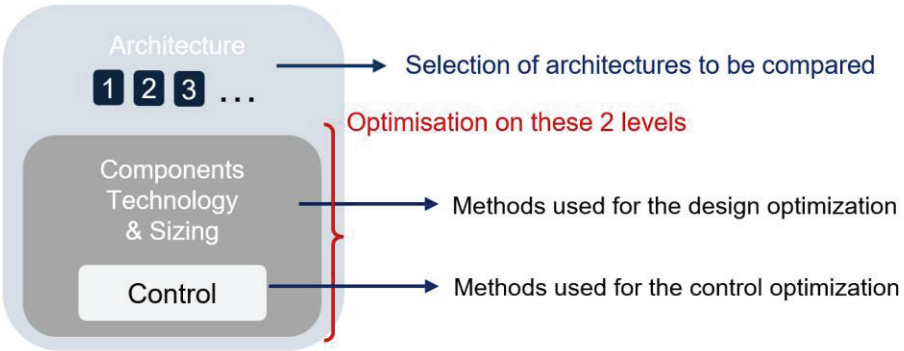


Figure 21: Traditionally

2.2.1. Methods used for the control optimization

As explained before, the (P)HEV powertrain has control variables that need to be decided instantaneously when the powertrain is being operated. Here are considered only the high level control variables managing the powertrain operating states and the energy sharing between the components. Low level component control (like injection control for the engine) is not considered.

In conventional pure thermal vehicles, if the gear ratio is already selected, a given speed and power demand on the wheels imposes directly the speed and torque of the engine. The only control variable is therefore the gear selection in the case of an automatic gearbox. However, in the case of a (P)HEV, a speed and power demand on the wheels can be met by the powertrain through different means. In fact, (P)HEVs have more than one energy converter onboard and can be operated on different modes or ratios. More degrees of freedom are therefore present: choice of operating mode, gear selection when a gearbox is involved, and the power sharing between the energy converters. These are the control variables that should be decided instantaneously in order to optimize the objective function (fuel consumption for example) while respecting the components constraints (maximum power for example) and other operation constraints like the final SOC of the battery. This is called the Energy Management Strategy (EMS) in (P)HEVs. The full energy saving potential of a (P)HEV can only be reached when an optimal EMS is used.

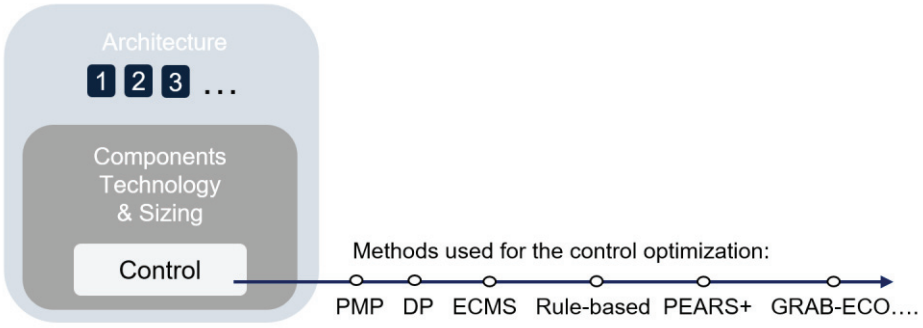


Figure 22: Methods used for the control optimization

Different EMS have been proposed for (P)HEV control (Figure 22). They differ in their optimality, computation time and ability to be implemented or not in real time vehicle operation. In [37], a good screening is made on these different EMS. Traditionally known EMS can be

categorized into two main families as shown in Figure 23: rule based methods and optimization based methods.

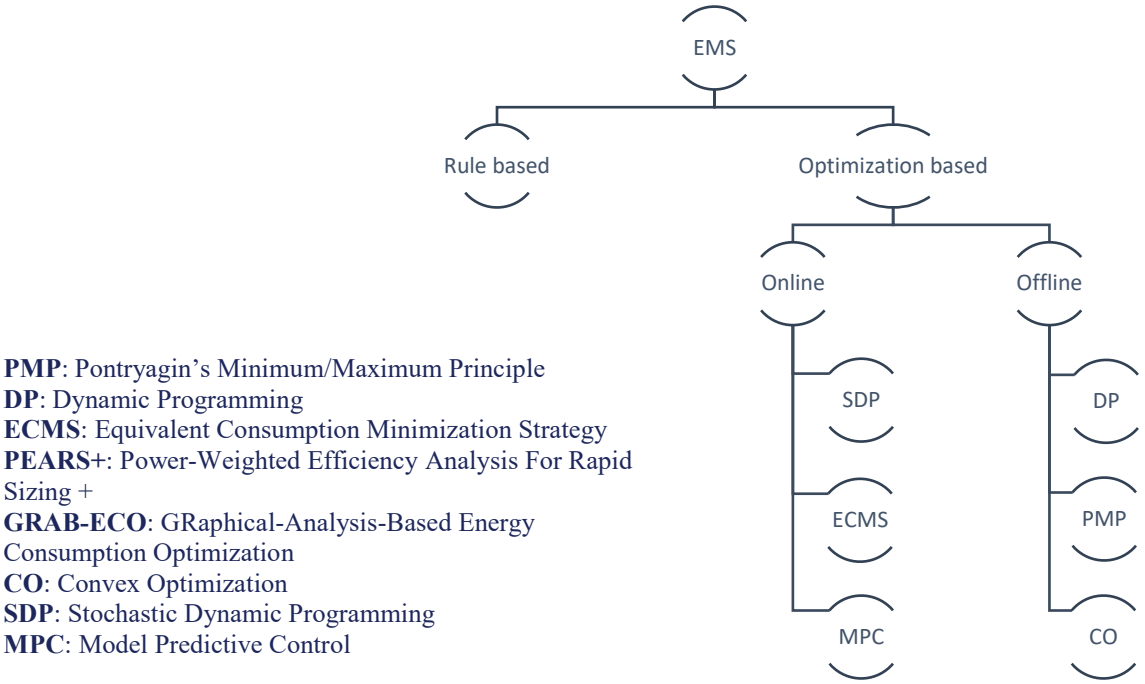


Figure 23: The main existing EMS

The first family of EMS and first to be used in vehicle applications are:

- **Rule based:**

  Real time implementable, these are strategies that do not involve instantaneous optimizations or global optimizations over the vehicle mission (driving cycle). Deciding the instantaneous values of the control variables is based on a set of rules that are derived from intuition, heuristics or the results of some beforehand made offline optimizations [37]. These rules can be of the following style:

  If wheel demand power (t) > threshold value f(SOC),

  Engine_state (t) = ON and P_engine (t) = value;

  Else,

  Engine_state(t) = OFF;

Rule based EMS are not optimal and require calibration efforts for each application case, which make them not suitable for design optimization and architectures comparison.

The second family of EMS are based on optimization techniques. These can be online EMS that can be implemented in the vehicle electronics for real-time operation (instantaneous optimizations), or offline EMS that are used on engineers PCs but not in real-time vehicle operation (global optimizations).

A global optimization of the control variables on the entire vehicle mission can only be done on a prescheduled trip or known driving cycle. Consequently, the EMS that guarantee a global optimal control cannot be online or real-time implementable. These are offline solutions. This is not an issue in our case because we are dealing with an offline design problem and we are not developing EMS to be real time implemented. The most used offline global optimal EMS are:

- **DP (Dynamic Programming):**

  DP is a widely used offline optimization algorithm in (P)HEV simulation and is based on the Bellman's Principle of Optimality [38]. DP ensures optimality (neglecting the discretization error) and serves as a reference in EMS benchmarking and in EMS improvements. Numerous works in (P)HEV optimization using DP as EMS can be found in the literature [39][40][32][35][36].

  Its principle is that a complex control problem can be solved by breaking it into multiple sub problems. This is done through numerical computation. This implies a computation time burden when the size of the problem is big. Understanding the physical meaning behind the control decisions is also harder than the analytical approaches (see PMP).

- **PMP (Pontryagin's Minimum/Maximum Principle):**

  PMP was first described in [41]. For (P)HEVs, the optimal control can be reached by instantaneously minimizing the Hamiltonian function, which is a sum of the fuel consumption and the electric consumption multiplied by an equivalence factor. This equivalence factor can be seen as the price of the electricity with respect to the fuel.

  For a given value of the equivalence factor, the minimum of the Hamiltonian is calculated by a numeric computation or an analytic solution. The final state of the system (final SOC) will depend on this given value. Iterative methods or root finding methods are

applied to find the value of the equivalence factor that will ensure the respect of the final state of the system (final SOC). Multiple works in (P)HEV optimization using PMP as EMS can be found in the literature [42][43][44]. In some works, a vectorization is done on the PMP calculation procedure and reduces the computation time [44].

DP and PMP has globally same results, even if some minor differences might appear sometimes due to the numerical discretization.

- **CO (Convex Optimization):**
  The optimal control can also be solved by convex optimization if the objective function and constraint functions can be adapted to convex ones. This is challenging in the case of (P)HEV. Some nested control techniques combining CO and PMP or DP has been proposed to solve CO alone problems [44][45].

Instead of using rule-based EMS for online application, some work proposed EMS based on optimizations but which do not require the pre knowledge of the entire vehicle mission. They reduce the global problem into an instantaneous formulation of a cost function. The latter is instantaneously optimized based on instantaneous information and on prediction information in some cases. The most known are **SDP** (Stochastic Dynamic Programming), **ECMS** (Equivalent Consumption Minimization Strategy)[46] and **MPC** (Model Predictive Control).

When the fuel consumption of one powertrain is being assessed, the simulation time difference between these EMS will vary from fractions of second to couple of minutes. This is negligible compared to one working hour. However, when a design space is being screened and thousands of design candidates are to be assessed, this difference can reach the order of couple of working hours.

This is why new near-optimal methods have been developed recently. Their aim is to reduce the computation time of the EMS without deteriorating the optimality, the latter can only be guaranteed by DP or PMP. These newly proposed EMS are offline methods used for the moment only in the design problem, but can be real-time implementable. Below are some examples. They do not appear in Figure 23 because they do not belong to the traditionally known EMS.

- **PEARS**:

  PEARS or Power-Weighted Efficiency Analysis For Rapid Sizing [47] is a near optimal EMS developed to replace DP in design optimization problems. PEARS starts by extracting the speed and acceleration data of the driving cycle (functions of time) and put them in a 2D table. This removal of the time scale will avoid calculating the system on a (speed, acceleration) point that is repeated many times. Then for each point, the different modes are considered and the best control is decided in a way to instantaneously optimize the system efficiency, once in electric and once in hybrid. The battery available energy is known and the required energy is calculated first by considering all the points to be electric. Then in function of the difference between these energies, points are shifted from electric to hybrid until the electric available energy meets the electric required energy.

- **PEARS+**:

  PEARS+ is a modified version of PEARS where the DP is applied to determine the mode shifts while the other control variables are determined as in PEARS [48][49].

- **GRAB-ECO**:

  GRAB-ECO or GRaphical-Analysis-Based Energy Consumption Optimization [44], is a proposed EMS that approximates the minimal energy consumption by maximizing the average operating efficiency of the energy source that has the worst efficiency.

2.2.2. Methods used for the design optimization

For the design optimization, the design space has a number of dimensions equal to the number of design variables (components technology and sizing). This number can be in the order of magnitude of 10. For each point of this design space, the evaluation of the point requires the calculation of the objective functions. The role of the design optimization method is to find the optimal design among all these points.

The least intelligent way to do this is by evaluating all the points. This method is called Exhaustive Search **ES** or brute-force search. When the number of design variables is important and the evaluation time of objective functions is not negligible, such method is not viable due to its computation time burden. Instead, more smart methods have been developed to find the optimal point without evaluating all the points (Figure 24).

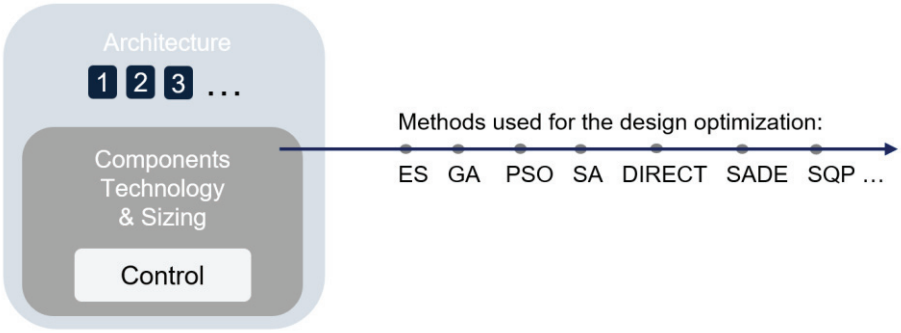


Figure 24: Methods used for the design optimization

They can be split into two main categories (Figure 25): gradient based methods and gradient free methods. In the first category one can cite **SQP** (Sequential Quadratic Programming) [50] [51] and **CO** (Convex Optimization) [52]. In the second category, the most used algorithms are **GA** (Genetic Algorithm) [32][53][54], **PSO** (Particle Swarm Optimization)[55][56][57][58], **SA** (Simulated Annealing) [59], **DIRECT** (Dividing Rectangles) [54] [44], **SADE** (Self Adaptive Differential Evolution) [60].

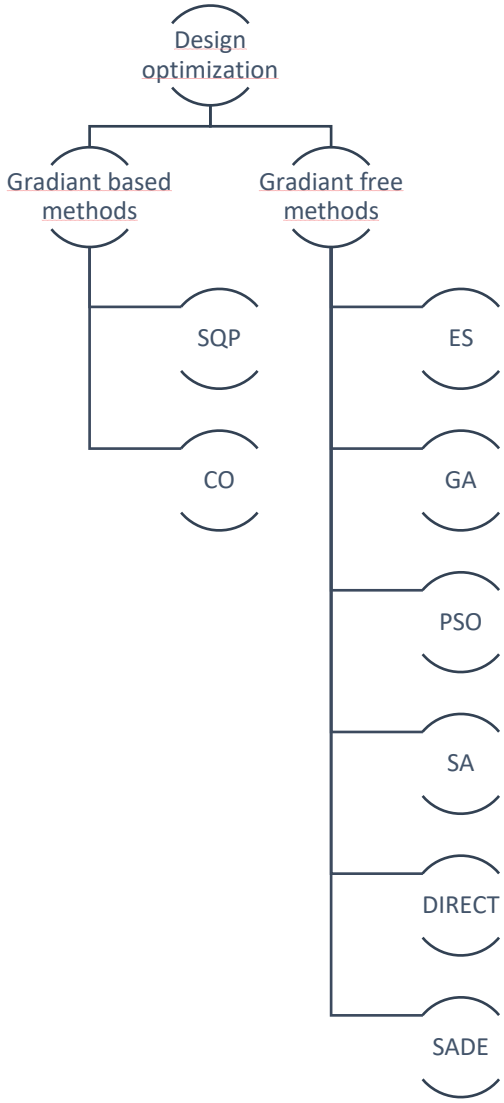


Figure 25: Categorization of the design optimization methods

2.2.3. Coordination approaches between the two levels: sequential, alternating, nested, simultaneous

After having exposed the methods that can be used on each level, now the coordination approaches between the optimization methods of each level are presented in Figure 26. They can be divided into 4 types [61], [35]:

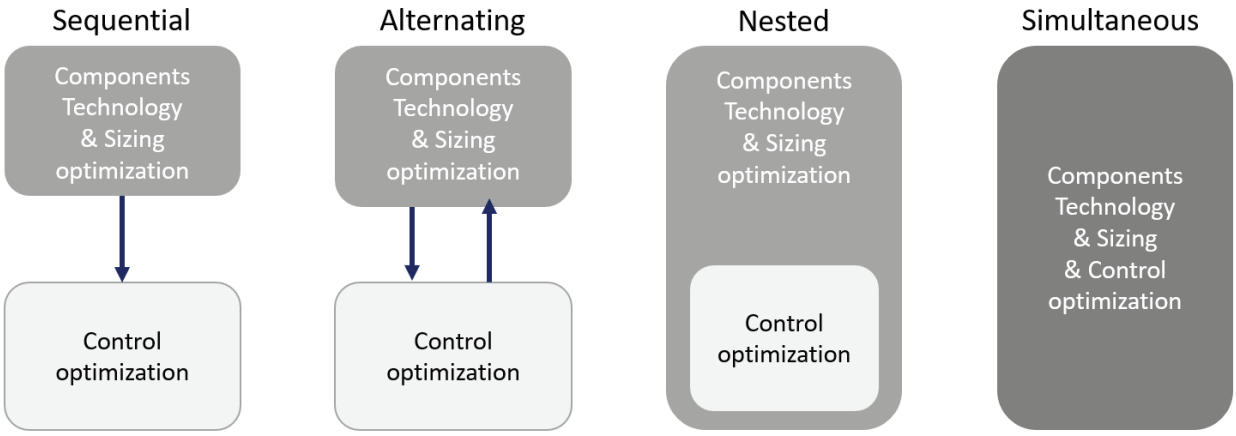


Figure 26: Coordination approaches between the levels

- Sequential: represents the traditional process used in the industry for powertrain design. This method starts by selecting the components technology and sizing. This can be done based on the conversion of the requirements from system level to component level, or on optimizations done with a fixed control. Then in a second step, the control is optimized for this selected design.

- Alternating: the component technology and sizing is optimized for a fixed control. The control is optimized for this selected design. Then the design is re-optimized with this optimized control. The control is re-optimized for this selected design… etc

- Nested: the component technology and sizing is optimized. The evaluation is not done using a fixed control like in the above types. For each design candidate the evaluation is done after a control optimization. One optimization algorithm is handling the design optimization level and for each evaluation, a control optimization algorithm is calculating the optimal control.

- Simultaneous: the variables from all levels are handled at the same time by one optimization algorithm.

As said before, in the traditional works found in the literature, few benchmark architectures are selected, optimized on the two levels (2) and (3), and then compared (Figure 21). Table 3 shows examples of these works.

Table 3: Examples of optimization methodology on level (2) and (3)
*FACE** [44]*: Fully-Analytic energy Consumption Estimation*

	IFSTTAR [32][61]	IFSTTAR [50]	IFPEN [44]	IFPEN [44]	TU Eindhoven [62][63]	U Michigan [64]
Components technology and Sizing	GA	SQP	DIRECT	FACE	GA; DIRECT; PSO; SQP; ES	ES
Control	DP		PMP; GRAB-ECO		DP	PEARS+
Coordination approach	Nested	Simultaneous	Nested	Simultaneous	Nested	Nested

Some works started to propose platforms where many combinations of vehicle application, driving cycles, architectures, components choice, sizing, and control can be assessed and compared. For instance, in [58] and [57] a multi-architecture / multi-application platform is proposed to design and compare powertrain solutions. The models are developed using EMR [65] (energetic macroscopic representation) combined with OOP (object oriented programming), making the models modular and reusable. The control can be done by DP, ECMS, deterministic RB, and fuzzy logic RB. PSO is used for optimization. Three architectures choice are available: series, parallel and power-split. Combinations of applications, architectures, driving cycles and control options are done and compared in [57].

2.2.4. Exploration of the architecture level: enumeration, automatic generation, filtering

In contrast with levels (2) and (3), the architecture level has not been sufficiently explored. This is because listing and modelling manually all the architectures that we can imagine is infeasible. In addition, it is because of the model complexity and computation time burden

associated with the architectures evaluation and optimization, in case of exhaustive search.

Nevertheless, exploration of the architecture level has already started. Few works have proposed architecture modifications by hand [40]. Others started to perform some automatic generation of architectures, in a way to discover a wider part of the architecture level instead of sticking to the few benchmark architectures.

In [49][66][67] a systematic design methodology is proposed. It generates and compares power-split architectures with two planetary gears. The involved automatic modelling is based on system matrix generation and analysis. Automatic mode screening and categorization is also done.

In [68], a methodology that can automatically generate graphs of hybrid architectures is proposed. It starts by selecting components from a predefined library of components. Some cost and functional constraints are also predefined. The constraint satisfaction problem behind this generation problem is clearly explained and solved using SWI-Prolog (SWI: initials of Social Science Informatics in Dutch) and the resulting graphs are drawn using Matlab. The work stops at the graphs level and the architectures are not evaluated. In [69], automated physical modelling and filtering is added. The topologies are automatically transformed into dynamic models using Simscape. These works will be recalled in Chapter 3 and compared to the proposed methodology.

2.3. OVERVIEW OF THE PROPOSED METHODOLOGY

To solve the (P)HEV design problem, we propose the methodology shown in Figure 27. It consists of four main parts (A, B, C and D) that will be overviewed hereafter. The detailed explanation can be found in chapters 3 and 4.

A: Design Inputs

The problem inputs are defined here, the '*what we have*' and the '*what we want'*.

- '*what we have'*: here are defined the available powertrain components that can be from different physical domains: thermal, electrical or oleo-pneumatic (the available engines, electric machines, batteries,..). In addition, the available connecting elements (clutches, gears, synchronizers) that can be used in the powertrain are selected.

- '*what we want*': here are defined the vehicle specification (performance requirements: maximum speed, acceleration 0-100km/h, electric range…) and the objectives to optimize (fuel consumption on different driving cycles, powertrain cost,…)

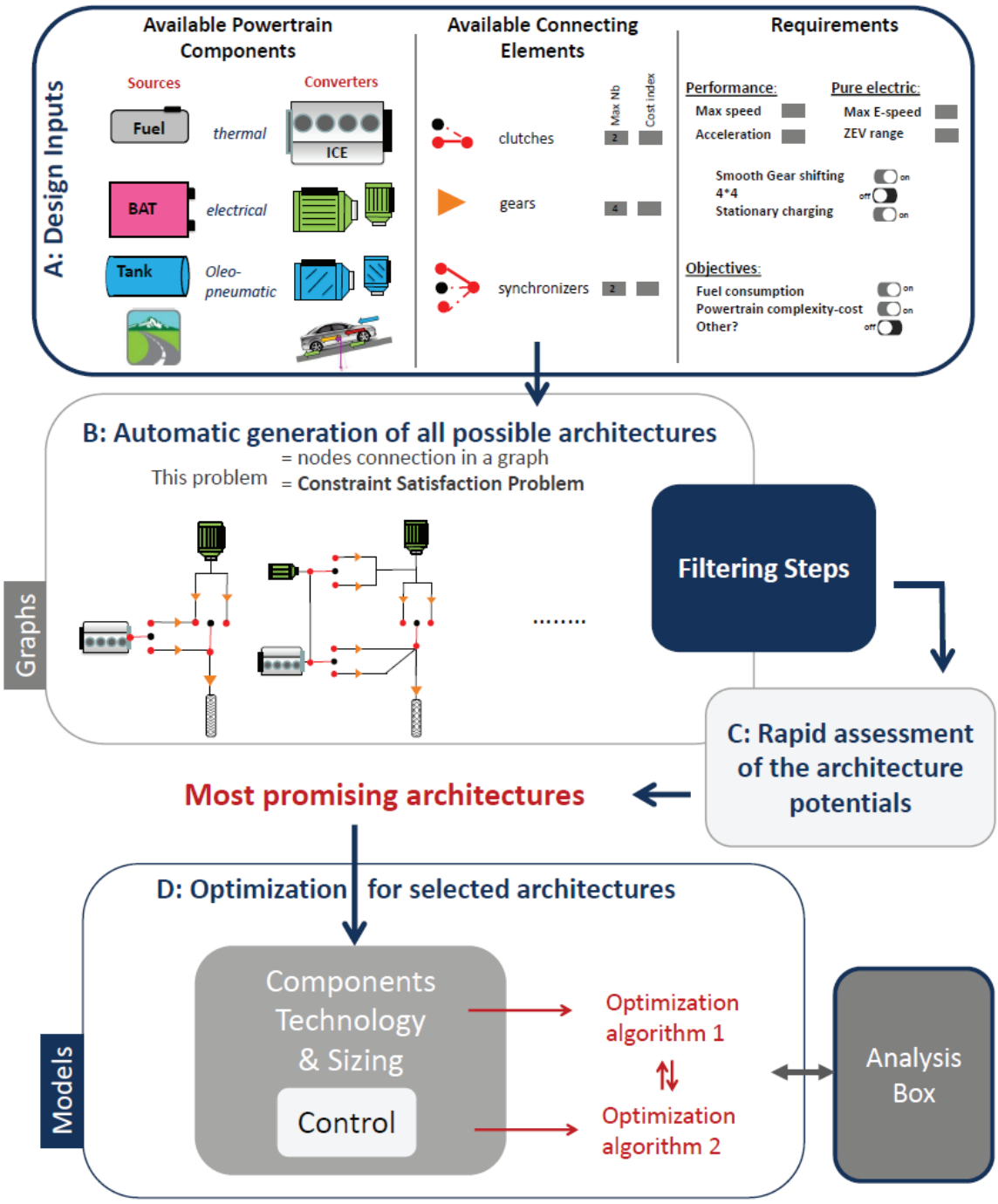


Figure 27: The proposed methodology

B: Automatic generation of all possible architectures:

The powertrain components and the connecting elements that can be used have been defined in A. Now, instead of fixing the powertrain architecture based on experts' knowledge or supplier constraints, the idea of this part is to generate all the possible architectures that can be realized to connect the components. These architectures are represented as graphs. Finding all the possibilities of connecting the nodes of a graph can be written as a constraint satisfaction problem and solved. This will be explained in details in chapter 3. The numerous generated graphs need to be filtered and only feasible ones are to be selected.

C: Rapid assessment of the architecture potentials:

Some of the generated graphs might be feasible but not presenting a good potential to be the best architecture because of the absence of some essential operating mode for example. This should be detected in this part of rapid assessment of the architecture potentials. The idea is to kill at early stages the useless architectures before arriving to the next steps of assessment by numerical simulation because of its computation time burdens. In a utopian case where the numerical evaluation of the architectures is rapid, this part C can be removed. It should be mentioned that it is crucial not to make wrong decisions in this step. In fact, killing by mistake any good architecture can make the process lose its optimality.

D: Optimization for selected architectures:

The most promising architectures need to be compared based on the objective functions in order to select the optimal. The assessment and optimization of the architectures is therefore needed. Numerical models of the architectures are required in this step. Developing numerical models for each of the architecture is infeasible. The target is to have a general model that can simulate all the generated architectures. Each architecture will be a special case of this ready model and will be optimized on the two levels of components technology & sizing, and control. Two optimization algorithms are needed here. The developed numerical models and the used optimization algorithms will be explained in details in chapter 4.

An application of the entire methodology is performed in chapter 5 to show its capability.

Contribution of the PhD:

1- Integration of the architecture level (Figure 28) in the optimization by exploring the architecture level instead of selecting already known architectures for comparison. This is done thanks to the automatic generation of architectures. It is to be mentioned here that power-split architectures are excluded from the methodology.

2- Development of a general hybrid model that can allow the assessment of all the generated graphs and therefore can connect the exploration work on the architecture level with the optimization work on the components technology & sizing and control levels.

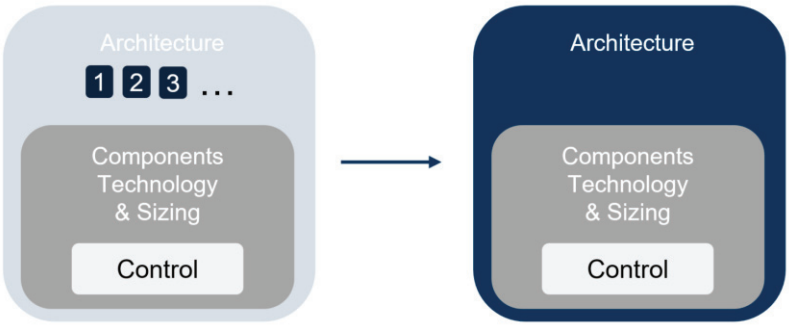


Figure 28: The covered area of the design space before and after the PhD

Conclusion:

This second chapter stated the problem of (P)HEV design optimization, overviewed the state of the art, and introduced the authors' proposed methodology to solve it. The explanation of this methodology will be the material of chapter 3 and 4, and the application of the methodology will be done is chapter 5.

CHAPTER 3: AUTOMATIC GENERATION AND FILTERING OF ARCHITECTURES

Abstract - This chapter explains the part of the methodology that explores the architecture level. There is a need to generate all the possible architectures that can be made when connecting the defined powertrain components and the defined connecting elements. This chapter explains in details how this is performed: from the graphical representation that is used to visualize the architectures, to the description of the called constraint satisfaction problem method, to the way it was solved, to the automatic filtering. The usage of the outcomes is presented in chapter 4 and a case study of the complete methodology is presented in chapter 5.

The objective of this part in the methodology is to generate all the possible architectures that can be made when connecting the defined powertrain components and the defined connecting elements. For that, these components and generated architectures need to be visualized.

3.1. GRAPHICAL REPRESENTATION FOR HYBRID ARCHITECTURES

A representation is needed in order to visualize the hybrid powertrain components and the generated architectures.

3.1.1. Representations found in the literature

In the literature, two remarkable representations are found in [68]and [70]:

A. In [68], an interesting methodology that deals with the automatic generation of hybrid powertrain topologies is proposed. A set of nodes (components) is defined (Table 4): Engine (ICE), Electric Machine (EM), Gearbox, Planetary Gear Set (PG), Differential+Wheel (FD), Clutch, Brake, and a 3-node connector. The connections are done from node to node. Each node can have one or more connections (edges). Powertrain components (power sources and wheels) that are on the extremity of the graph have one connection and the transmission components (gearbox, clutch) in the center of the graph have two connections. The gearbox is considered as one node that can have two connections (1 input, 1 output) and that can be used once.
   In this representation, three edges can never be connected, expect through the Planetary Gear Set that is allowed to have three edges and through the 3-node connector, which is also allowed to have three edges. The 3- node connector can be seen as a torque coupler with two inputs and one output.
   An example on this representation is shown in Figure 29. The nodes are numbered as V_{xy} where x is the component number and y is the instance number.

Table 4: Library of nodes used in [68]

Component Number, τ	Component Name	Maximum Number of Instances	Number of Edges
1	Engine	1	1
2	Electric Machine	2	2
3	Gearbox	1	2
4	Planetary Gear Set	2	3
5	Differential+Wheels	1	1
6	Clutch	3	2
7	Brake	3	1
8	3-node connector	3	3

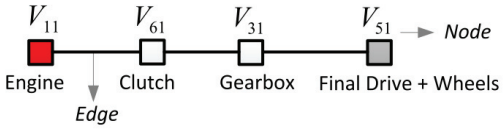


Figure 29: The graph of a 5-speed manual gearbox powertrain with the representation in [68]

B. In [70] a tool is developed to automatically generate gearboxes. In contrast with A, the gearbox here is not one node or one component. The gearbox is displayed in a lower level of details. It is made of gears and synchronizers and these are the nodes. Another dissimilarity with A is that the graph here is bipartite, meaning that 2 types of nodes exist: components (drawn rectangular) and shafts (drawn circular). The components cannot be interconnected; they connect only to the shafts.

An example on this representation is shown in Figure 30; e: the input shaft, s: output shaft, Gs: Gear set, S2: 2-positions synchronizer, and S3: 3-positions synchronizer.

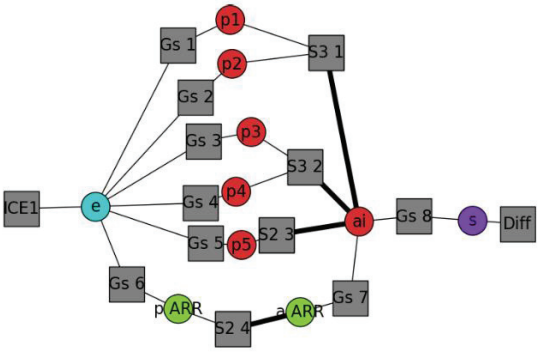


Figure 30: The graph of a 5-speed manual gearbox powertrain with the representation in [70]*, missing the clutch.*

3.1.2. The proposed representation

Inspired by these two works, a graphical representation is proposed to be used in this PhD (Figure 31). It will be compared to the A and B representations through an example in Figure 33. In this proposed representation, a powertrain is visualized as a graph. This graph is made of nodes and connections. The nodes are of two types (bipartite graph): components (circles in colors) and connectors - i.e. shafts - (small circles in black). No connection can be done between the components. Connections are only done through the shafts.

The components nodes are:

- Powertrain components: the Internal Combustion Engine (ICE), the Motor Generators (MGs), the Final Drive output (FD)
- Gearbox components: the clutches and the 'synchro' unit (Synchronizer + 2 gear pairs)

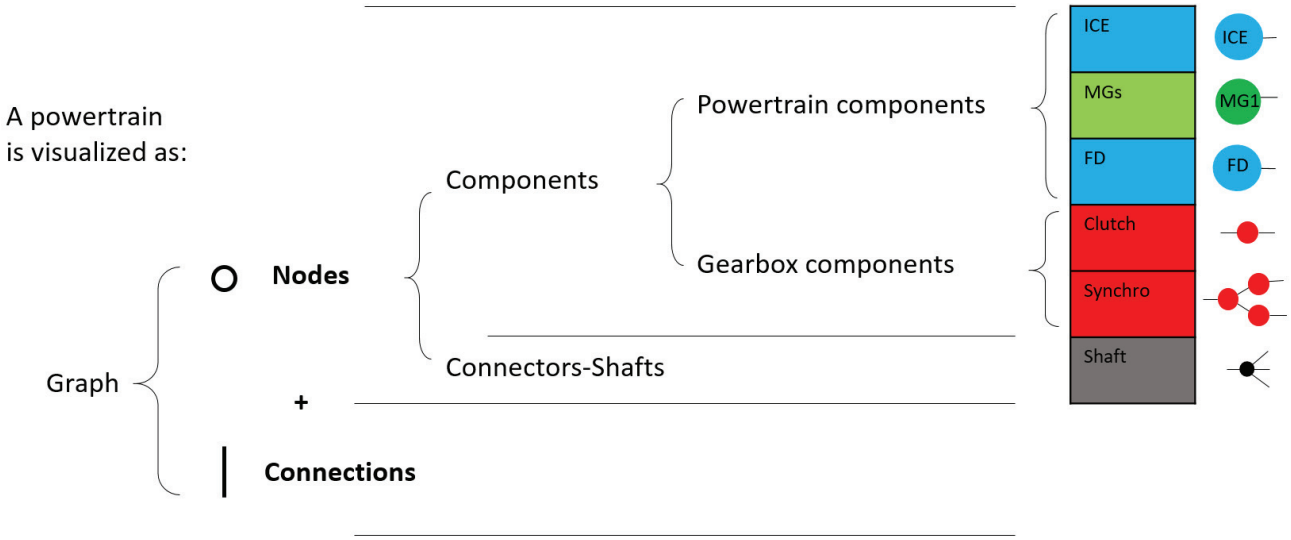


Figure 31: The proposed representation

A comparison between the proposed representation and the ones found in A and B is done in Figure 32: the proposed representation has a level of details appropriate to generate the new-trend dedicated hybrid gearboxes (example: Renault E-Tech, Eolab1 and Eolab2 that can't be generated by A), without making the problem too complex like in B (see next section).

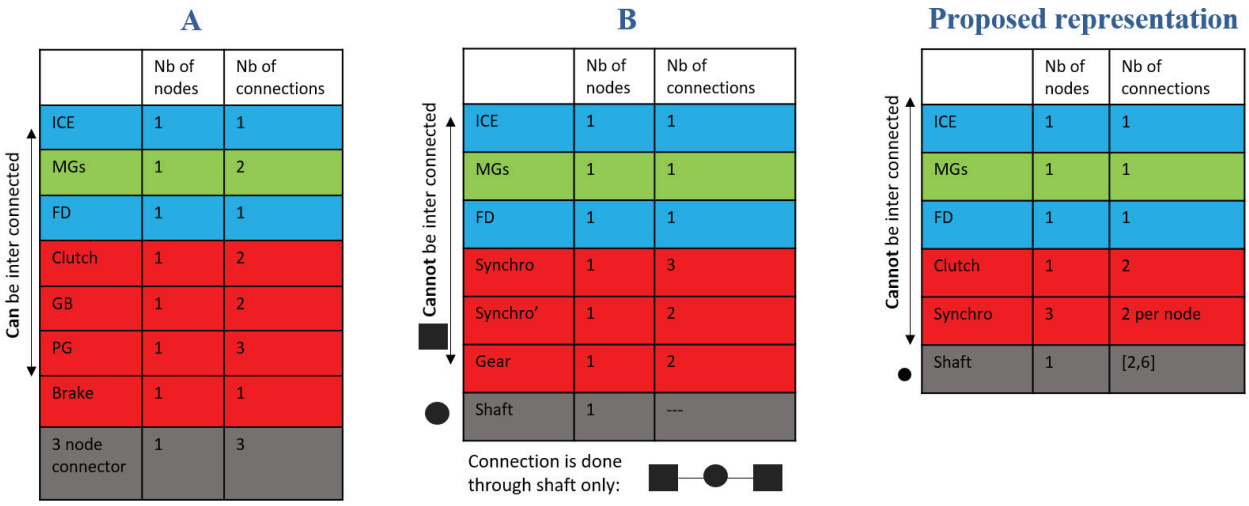


Figure 32: Comparison between the proposed and other representations

The powertrain architecture of a 5-speed manual gearbox vehicle will look as follows in each of the above representations (Figure 33).

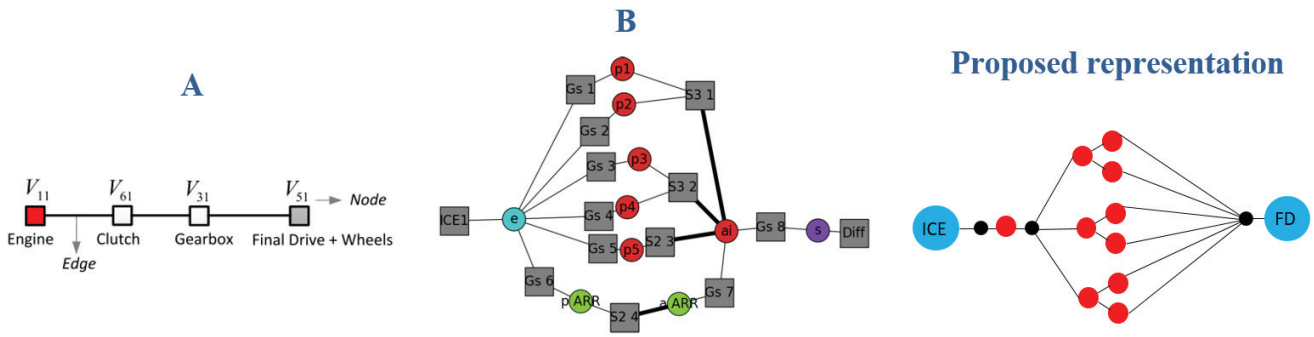


Figure 33: 5-speed gearbox example

3.1.3. The 'synchro' unit in detail

In the proposed representation, a 'synchro' unit is used. Such units are seen in manual gearboxes and consist of a synchronizer and 2 pairs of gears. In each pair of gears, one gear is fixed to a shaft and one is free spinning and connects to a shaft through the synchronizer. Functionally, this unit allows 3 options: disconnection, connection through a gear ratio 1, connection through a gear ratio 2. An example of a 4-speed gearbox connecting an ICE to FD is shown in Figure 34. It consists of 2 'synchro' units. Each unit has a synchronizer with dog clutches (shown in violet) and 2 pairs of {free-spinning gears (shown in blue) + fixed gear (shown in green)}.

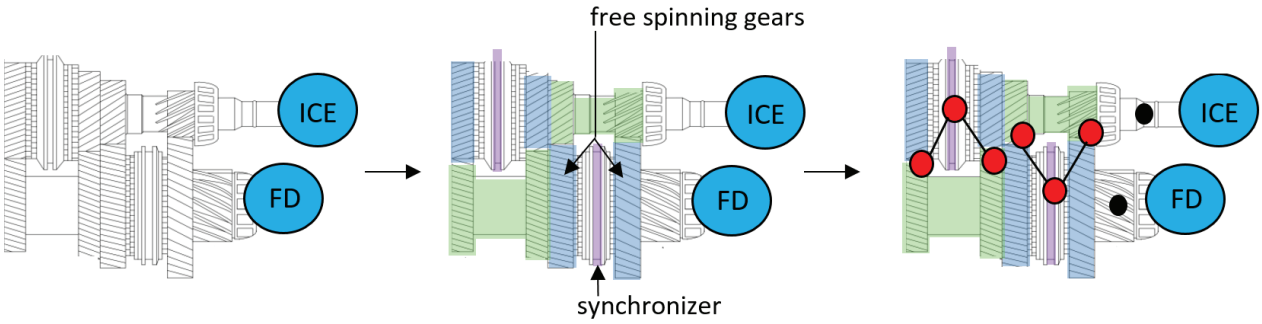


Figure 34: The 'synchro' unit

This ‘synchro’ unit is visualized as an entity of 3 red nodes (Figure 34, right). This entity has three external connections. From one side it is connected to one shaft only (Figure 35), this is the side of the unit where the synchronizer is placed and moves between the two free spinning gears. From the other side, the two gears are connected to one shaft or two shafts (Figure 35).In conventional gearboxes that connect 1 input shaft (engine shaft) to 1 output shaft (final drive shaft), the 2 gears of the synchro unit are always connected to 1 same shaft (Figure 34). However, in the new-trend dedicated hybrid gearboxes (example Renault E-Tech, Eolab1 and Eolab2), the EM can be placed on a separate shaft and can be given devoted gear ratios not shared with the ICE. Such gearboxes cannot be generated unless our representation takes into account the option ‘Synchronizer Case 2 shafts’ (Figure 35).

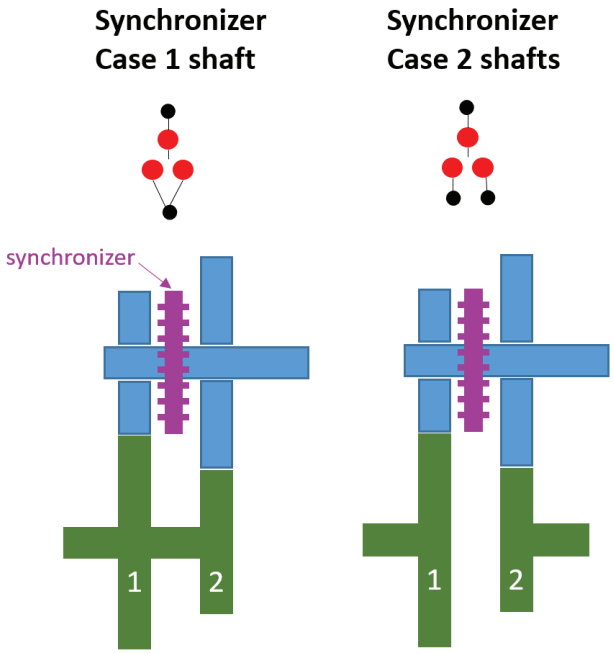


Figure 35: Synchronizer 1 shaft and 2 shafts cases

The synchronizer can be in 3 states (Figure 36):

- in the center and not connected to any free-spinning gear (position Nutral)
- to the left and connected to the left gear (position Gear 1)
- to the right and connected to the right gear (position Gear 2)

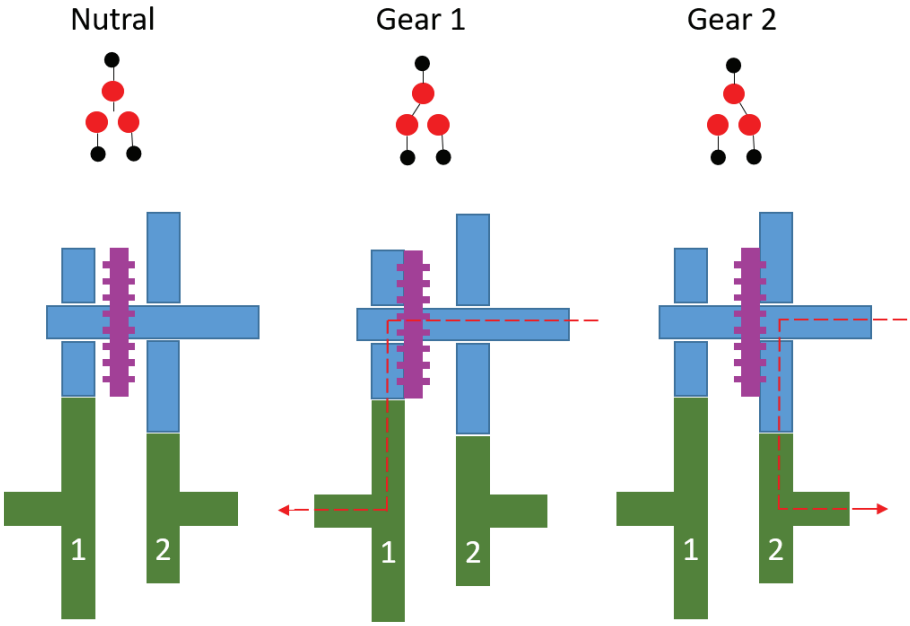


Figure 36: The 3 positions of the synchronizer

3.1.4. Gears placement in the representation

Gears are not shown in our representation, in contrast with representation B. However, each component edge and each of the right and left node of the 'synchro' unit have an attribute gear ratio. An example is shown in Figure 37: the green circles are the location of 'synchro' attribute gears and the green lines are the location of the components attribute gears.

By default, the value of the gear ratios is 1 for the components attribute gears (green lines). However, later on at the sizing optimization step, these values can be chosen to be different than 1. In the latter case, a corresponding complexity/cost/volume for the gear is added.

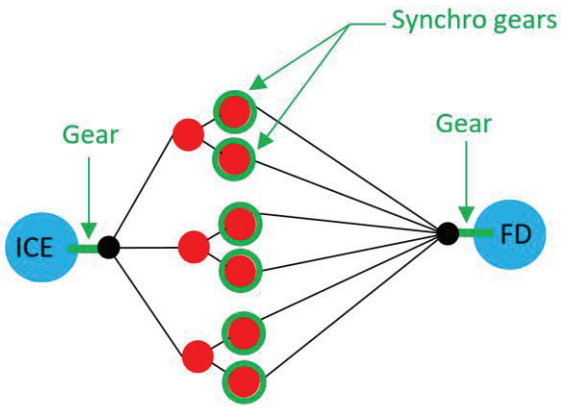
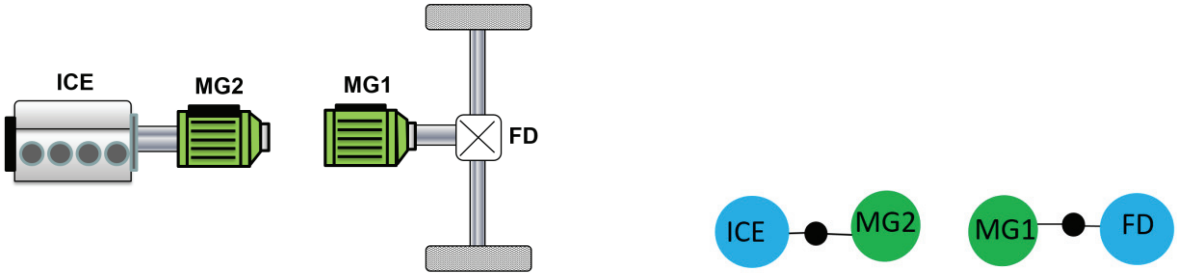


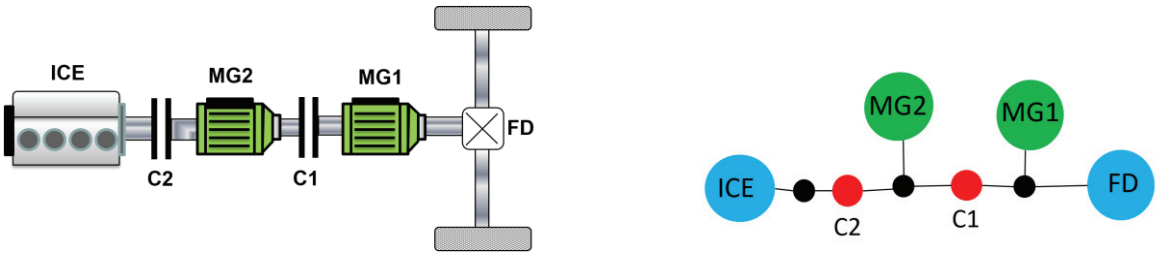
Figure 37: Gears are attributes not shown in the representation, the green arrow show where a gear attribute is present

Below are examples on how some known architectures can be seen using the proposed representation:

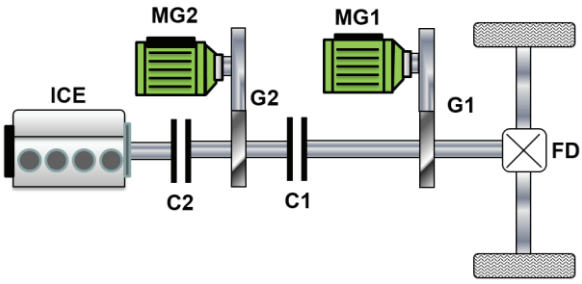
- A series architecture:

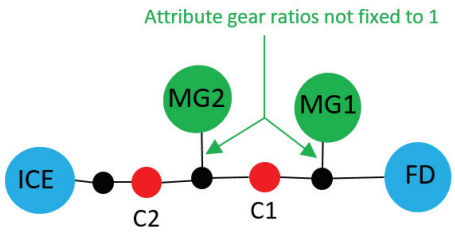


- A simple series-parallel architecture:



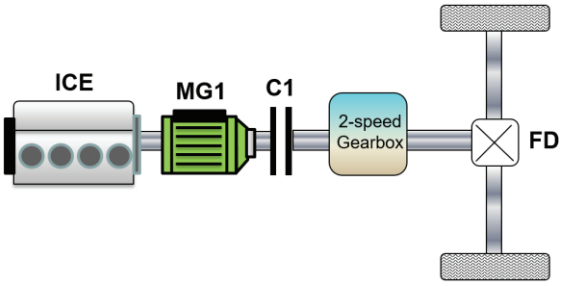
- A simple series-parallel architecture with gears for the electric machines:

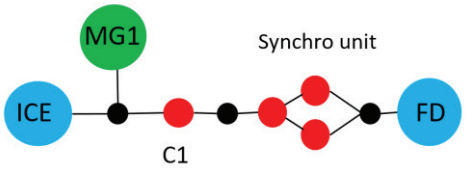




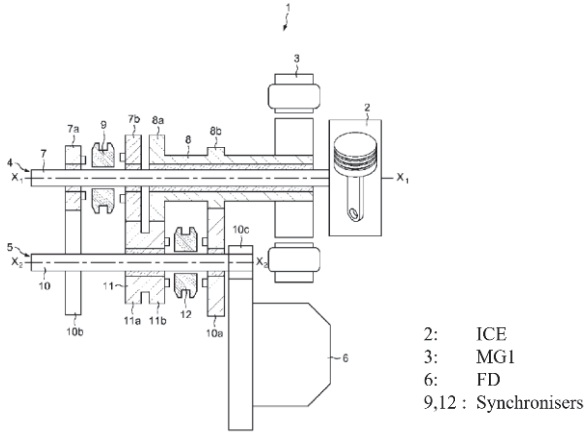
This architecture and the 'simple series-parallel architecture' have the same representation. Yet, here the electric machines have attribute gears non-equal to 1 whereas in the 'simple series-parallel architecture', they are fixed to 1.

- A parallel architecture with a 2-speed gearbox:

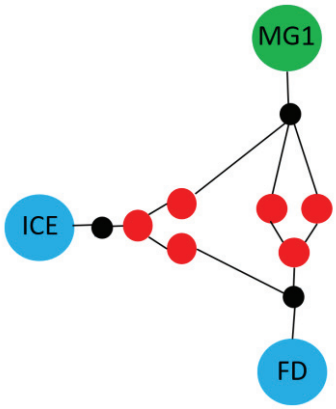




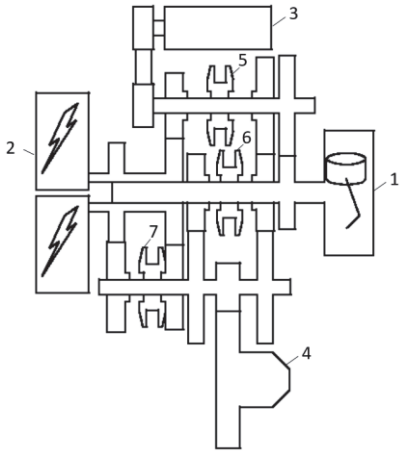
- A Renault patented architecture [71], it will be referred to as Eolab1:



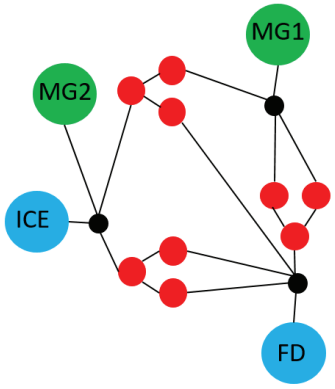
2: ICE
3: MG1
6: FD
9,12 : Synchronisers



- A Renault patented architecture [72], it will be referred to as Eolab2:



1: ICE
2: MG1
3: MG2
4: FD
5,6,7: Synchronisers



It is to mention that after we had defined our representation shown above, other representations were found in the literature. For instance, in continuation of the work in [68] (representation A), in [36] an updated list of components was used. It is presented in Figure 38.

Component type, ϕ	Abbre-viation	Max number of instances	Number of edges
1: Internal combustion engine	ICE	1	1
2: Electric machine	EM	2	1
3: Wheels	W	1	1
4: Gear pair	GP	n.l.	2
5: Planetary gear set	PGS	n.l.	3
6: Clutch	C	n.l.	2
7: Clutch pair	CP	n.l.	3
8: Brake	B	n.l.	1
9: Ground	G	n.l.	1
10: Virtual node	VN	n.l.	3

n.l. = not limited

Figure 38: Library of components in [36]

It should be also mentioned that in our proposed representation no planetary gear component was used. That was Groupe PSA and the authors' decision to exclude for the moment the power-split architectures from the entire methodology. The target was to simplify the problem and create as soon as possible a complete methodology that can generate and assess architectures with discrete gearboxes components. This means that for the moment, the methodology explores the architecture level while excluding the part corresponding to power-split architectures. The addition of the planetary gear components will be a next step after end of this PhD.

3.2. CONSTRAINT SATISFACTION PROBLEM

In this section, the problem of automatic generation of architectures is explained. Before this, a definition of what is called a Constraint Satisfaction Problem (CSP) is done.

A CSP is a problem that can be stated as follows [73]:

- A set of variables $X = \{ x_1,\dots, x_n\}$,
- Each x_i has a domain of possible values D_i
- A set of constraints $C_{i,j,k..}$ restricting the values that the variables x_i, x_j, x_k, ... can take simultaneously

Finding a feasible solution to a CSP consists in assigning a value to every variable from its domain, while respecting all the constraints. One might want to find one feasible solution, all feasible solutions, or an optimal solution if an objective function is defined.

The problem of automatic generation of architectures can be formulated as a CSP. The involved variables, domains and constraints are defined hereafter.

3.2.1. Problem variables and their domain

As seen before, the architecture of a vehicle powertrain can be visualized as a graph: a set of nodes (components) connected by a set of edges (connections). This graph is connected and undirected because the connections do not have directions. Let us consider a graph G composed of a set of nodes $\boldsymbol{V}$ and connected by a set of edges $\boldsymbol{E}$. In software that handle graphs creation, manipulation and visualization, G can be handled as [74], [75] :

- Adjacency list:
  List of the n nodes $\boldsymbol{V} = \{ V_0, \dots V_{n-1}\}$ and list of the edges $\boldsymbol{E} = \{ \{V_0, V_1\}, \dots \{V_{n-2}, V_{n-1}\} \}$
- Adjacency matrix:
  n by n matrix where the columns are the nodes $V_0, \dots V_{n-1}$, the rows are the nodes $V_0, \dots V_{n-1}$, and each cell at index (i, j) is 0 or 1 value describing the absence (0) or presence (1) of connection between the node V_i and V_j

The adjacency lists avoid the big matrices size and are useful especially when the graph is not dense (the case where the adjacency matrix will be mostly made of zeros, with few ones).

However, in this work, the graphs will be handled through their adjacency matrix because it will simplify the understanding and solving of the CSP, and because no problem of memory was detected in our case.

Based on a review of the commonly used hybrid powertrains and the hybrid-dedicated gearboxes found in patents, we will consider in this PhD that the powertrain can include a maximum of:

- 1 ICE
- 2 MGs
- 1 FD (Final Drive output)
- 4 shafts (X1 to X4 in Figure 39)
- 3 Clutches (C1_R to C3_L)
- 3 Synchronizers (S1_0 to S3_L)

To solve the generation problem, the case of a powertrain including all these components is considered. The number of nodes is 23. The corresponding adjacency matrix is shown in Figure 39.

		ICE	MG1	MG2	FD	clutches						synchro									shafts			
						C1_R	C1_L	C2_R	C2_L	C3_R	C3_L	S1_0	S1_R	S1_L	S2_0	S2_R	S2_L	S3_0	S3_R	S3_L	X1	X2	X3	X4
		V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
ICE	V0	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
MG1	V1	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
MG2	V2	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
FD	V3	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C1_R	V4	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C1_L	V5	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C2_R	V6	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C2_L	V7	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C3_R	V8	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C3_L	V9	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_0	V10	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_R	V11	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_L	V12	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_0	V13	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_R	V14	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_L	V15	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_0	V16	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_R	V17	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_L	V18	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X1	V19	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X2	V20	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X3	V21	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X4	V22	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]

Figure 39: Adjacency matrix of the powertrain including all the defined components

The rows and columns correspond to the nodes of all the components $V_0, \dots V_{22}$. In blue and green are the powertrain components' nodes (ICE, MGs, FD), in red are the gearbox components' nodes (clutches, synchronizers), and in grey the (shafts) nodes. The matrix has $23 * 23 = 529$ cells that can have a value of 0 or 1. When a node is not present, its corresponding rows and columns are set to 0. The case of all nodes present is considered hereafter. The initial number of solutions is:

$$2^{529} = 1.7574 \times 10^{159} \quad (1)$$

Generating the graphs of the architectures that can be made out of these components while respecting the design constraints is equivalent to finding all the adjacency matrices that respect the constraints. This means that the problem of graph generation is a CSP problem where the variables are the 529 cells of this 23*23 matrix, the domain of all variables is {0, 1}, and the solving target is to find all the feasible solutions.

3.2.2. Problem constraints

Any solution to this CSP problem should respect some constraints regarding the components connection, mechanical feasibility, powertrain functionality and the non-repeatability of the components.

In the early works on the generation tool, some basic constraints were defined. Then, based on the generated graphs and by trial and error, these constraints were enhanced and others were added. At the tool final stage, the following constraints were the ones considered.

- C000: The adjacency matrix is symmetric because the graph of mechanical connections is an undirected graph: node 1 is connected to node 2 is equivalent to node 2 is connected to node 1.

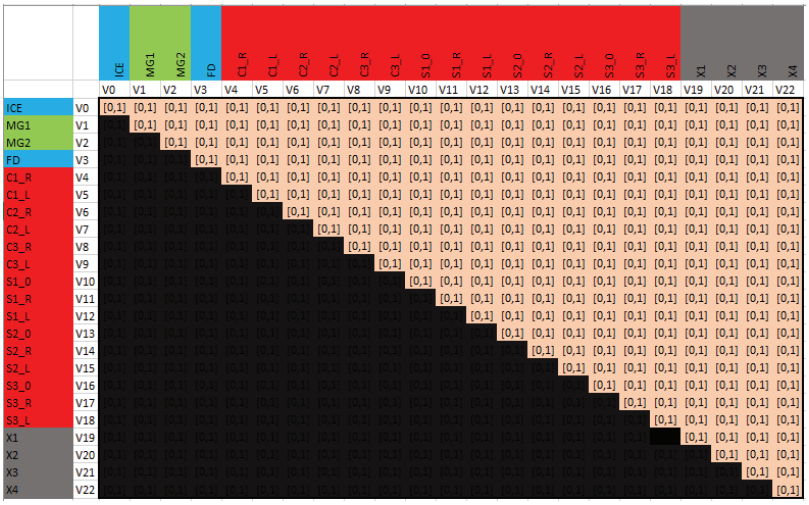


Figure 40: The adjacency matrix after the addition of C000

The number of solutions is reduced to:

$$2^{(\frac{23*23}{2}+\frac{23}{2})} = 2^{276} = 1.2142 \times 10^{83} \quad (2)$$

- C001: Fixed values
  - C001a: The nodes cannot be self-connected. Hence, the values in the diagonal should be set to 0.

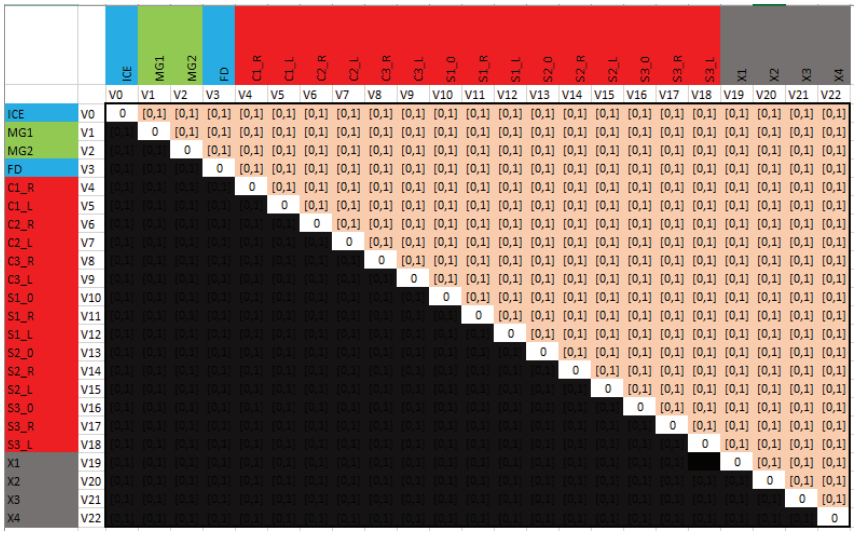


Figure 41: The adjacency matrix after the addition of C001a

The number of solutions is reduced to:

$$2^{276-23} = 2^{253} = 1.4474 \times 10^{76} \quad (3)$$

- C001b: The nodes of the components can only connect to the shafts nodes. They can not connect to other component nodes (bipartite graph). All the values of the variables corresponding to connection between components nodes are therefore 0. An exception is made for the synchronizer (For visualization purposes, the synchro unit is made of 3 nodes and these nodes are always connected).

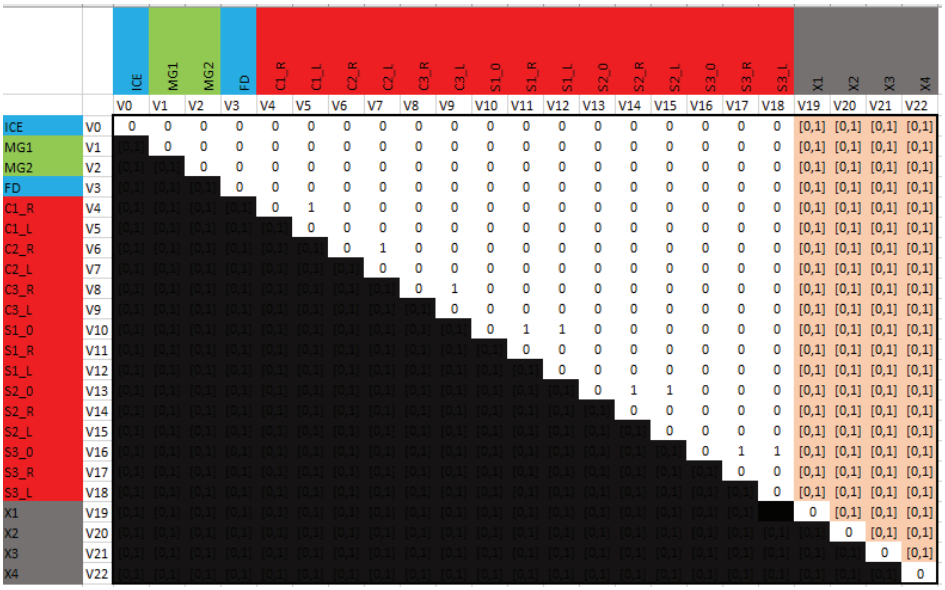


Figure 42: The adjacency matrix after the addition of C001b

The number of solutions is reduced to:

$$2^{82} = 4.8357 \times 10^{24} \tag{4}$$

- C001c: The shafts cannot be connected together, because 2 connected shafts = 1 shaft with more connections.

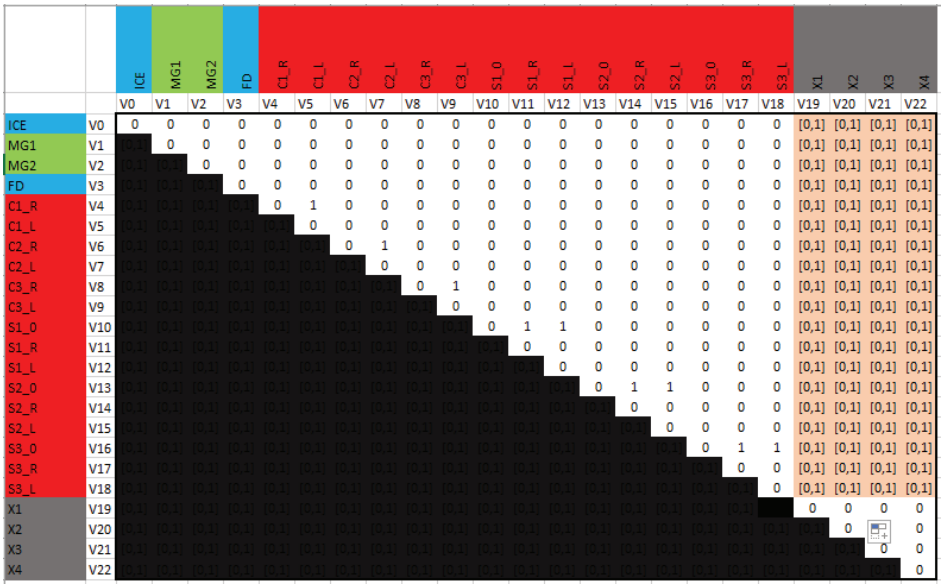


Figure 43: The adjacency matrix after the addition of C001c

The number of solutions is reduced to:

$$2^{76} = 7.5558\times 10^{22} \tag{5}$$

- C002 : minimum and maximum number of connections per node (Figure 44). The shafts are allowed to have between 2 and 6 connections. The ICE, EMs and FD are on the extremity of the graphs and are allowed to have only 1 connection which will be made with a shaft. The gearbox elements (clutch and synchro) have 2 connections, except for the S_0 node (the node where the synchronizer is placed) which is making 3 connections (see the representation in Figure 31).

	Minimum nb of connections	Maximum nb of connections
ICE	1	1
EMs	1	1
FD	1	1
Clutch	2	2
Synchro S_0	3	3
Synchro S_R, S_L	2	2
Shaft	2	6

Figure 44: Minimum and maximum number of connections per node

- C003 : ICE and FD cannot be connected to same shaft (idling and minimum rotation speed constraint)
- C004 : MG1 and MG2 cannot be connected to same shaft (equivalent to one MG)
- C005:
  - Clutch nodes cannot be connected alone to 1 same shaft unless a component is connected to this shaft (clutches cannot be in series)
  - Synchro nodes cannot be connected alone to 1 same shaft unless a component is connected to this shaft
- C006: Shafts should be connected to at least 1 powertrain component (ICE,MG,FD)
- C007: Synchro S_0 and S_L or S_R cannot be connected to same node (otherwise it makes a loop)
- C008: 2 clutches cannot be in parallel
- C009: 1 clutch and 1 synchro cannot be in parallel (otherwise the clutch will be useless)

A summary of the considered constraints is presented in Figure 45:

C000	Symmetric matrix (undirected graph)
C001	Nodes can be connected to 'shafts' only
C002	[Min, Max] # connections
C003	ICE and FD cannot be connected to same shaft
C004	MG1 and MG2 cannot be connected to same shaft
C005	Clutches nodes can't be connected alone to 1 same shaft; Same for synchro nodes
C006	Connectors should be connected to at least 1 (ICE-MG-FD) component
C007	Synchro S_0 and S_L or S_R cannot be connected to same node
C008	2 clutches can 't be in parallel
C009	1 clutch and 1 synchro cannot be in parallel

Figure 45: Summary of the problem constraints

3.2.3. Problem implementation

Now that the problem variables, domains and constraints have been defined, the next step is to choose an environment where this CSP can be implemented and a solver that can solve the problem. In [68], this was done using SWI-Prolog [76].

After a screening and handling of the available tools that can be used to solve CSP problems (SWI-Prolog [76], Python-Constraint [77], Numberjack [78], facile [79], Google OR-Tools [80]), the Python-Constraint module on Python environment was selected. This choice was made because the problem implementation in Python-Constraint was simple and because working on Python is convenient: later on, finding modules on Python for graph visualization or other purpose will be easy, knowing the numerous available libraries and modules with open access on Python.

An example on how a CSP problem can be implemented and solved in Python-Constraint is presented in annex 1. In a similar manner, and based on a code initially done for Sudoku solving, the problem of graph generation was coded. The main stages in the developed code can be found in annex 1.

3.3. AUTOMATIC GENERATION OF THE ARCHITECTURES

After having implemented the problem as shown above, the automatic generation of architectures consists now in solving this problem to get all the solutions of adjacency matrices and then in transforming these matrices into graphs and visualizing them.

3.3.1. Problem solving

In general, CSP problems with integer values and finite domains can be solved using Integer Programming (IP) techniques or Constraint Programming (CP) techniques.

The authors' understanding of the difference between IP and CP techniques is presented below in Figure 46.

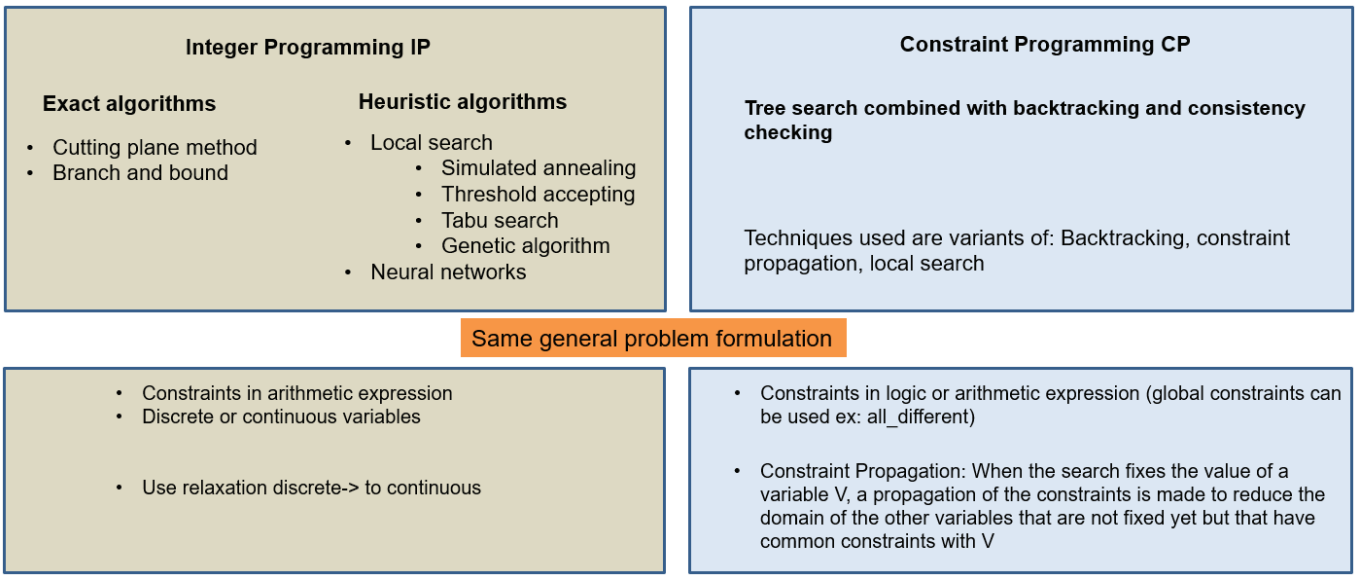


Figure 46: Techniques to solve CSP

Their understanding of the difference between Branch and Bound BB and CP as tree search techniques is shown below.

BB and CP are both tree search techniques	
BB	**CP**
Used in Linear programming with integer variables. We branch = We divide the domain in 2. We bound if the solution of the node is worse than the current solution. We don't propagate constraints, we calculate the objective function on the node	We branch on all the values of the variables. But we constantly do constraint propagation to reduce the domain of the variables and remove unnecessary values.

Figure 47: Comparison between Branch and Bound (BB) and Constraint Programming (CP)

Based on this understanding and on information found in [68] and [73], CP was selected to be used because our problem is highly constrained and in this case CP performs better than IP [73]. Another reason for choosing CP is that our logical constraints can be easily stated in CP, precisely using Python-Constraint defined functions.

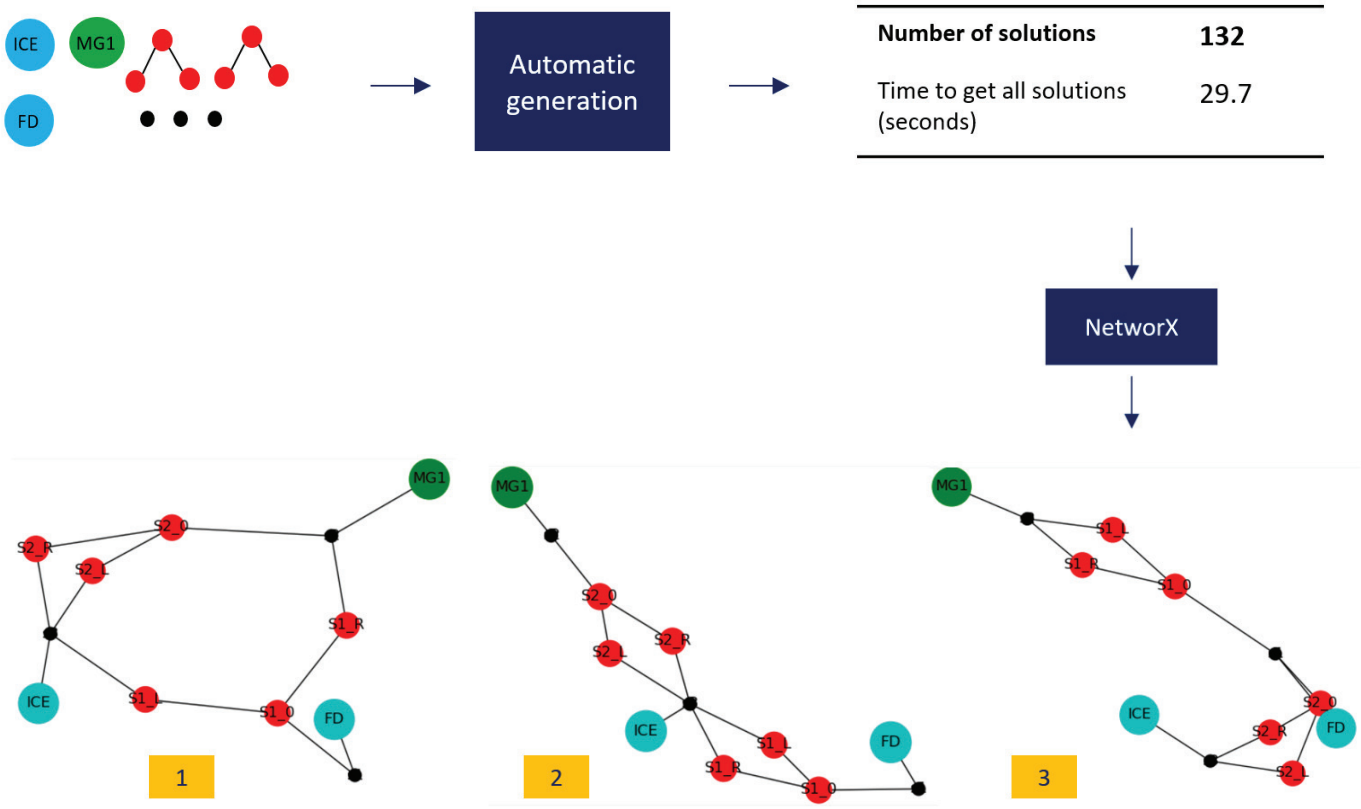
After having implemented the problem using Python-Constraint module, the default CP solver named "backtracking" in Python-Constraint is selected. This solver has backtracking capabilities and forward checking capabilities (constraint propagation) and can generate all possible solutions. It is suited for our problem and solved it with no issues. For example, the Eolab1 (one of the two Renault patented architectures presented earlier) components were used

as input of the tool and the generation of graphs was launched. At the end of the process, 132 solutions were found in 29.7 seconds (Figure 48). These results are found using a computer equipped with a processor Intel® Core™ i7-6820HQ CPU @ 2.70GHz 2.71GHz and a 32.0Go RAM.

Figure 48: Example on the resolution of the problem, case Eolab1 components

3.3.2. Generated graphs

These solutions are in form of adjacency matrices. NetworkX module [81] on Python is used to transform these adjacency matrices into graphs that are visualized to the user. The example of Eolab1 components is continued here and 6 of the generated graphs are shown (Figure 49).



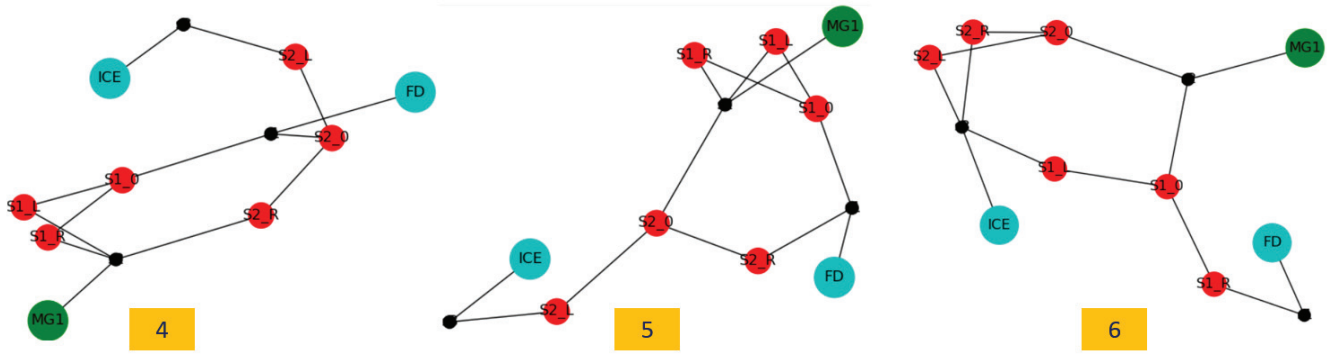


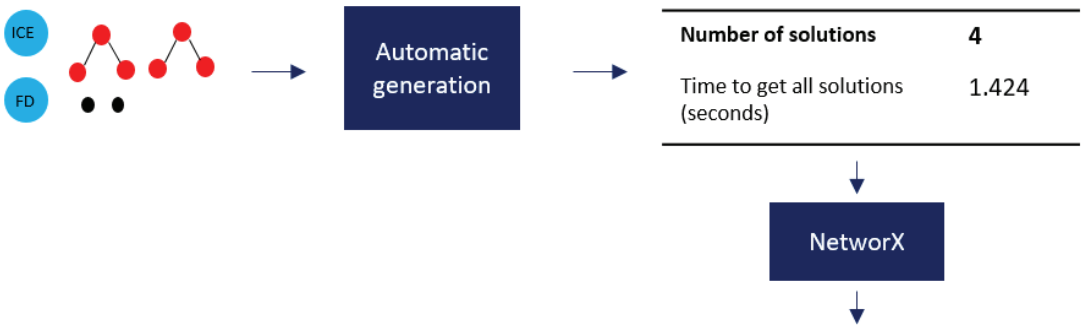
Figure 49: Example of generated graphs, case Eolab1 components

3.4. AUTOMATIC FILTERING AND ANALYSIS

After having generated the graphs, steps of filtering and analysis were needed. They are presented in this section.

3.4.1. 0ABC Table

In the example of Figure 49, 6 solution graphs were shown. As it can be seen, the placement of the synchro and components in these graphs is very different from graph to graph. That was not the case for all the 132 solution graphs. In fact, some graphs were found to be redundant or isomorph due to a symmetry problem. This is explained hereafter using the simple example of the components of a conventional powertrain with a manual 4 speed gearbox (Figure 50).



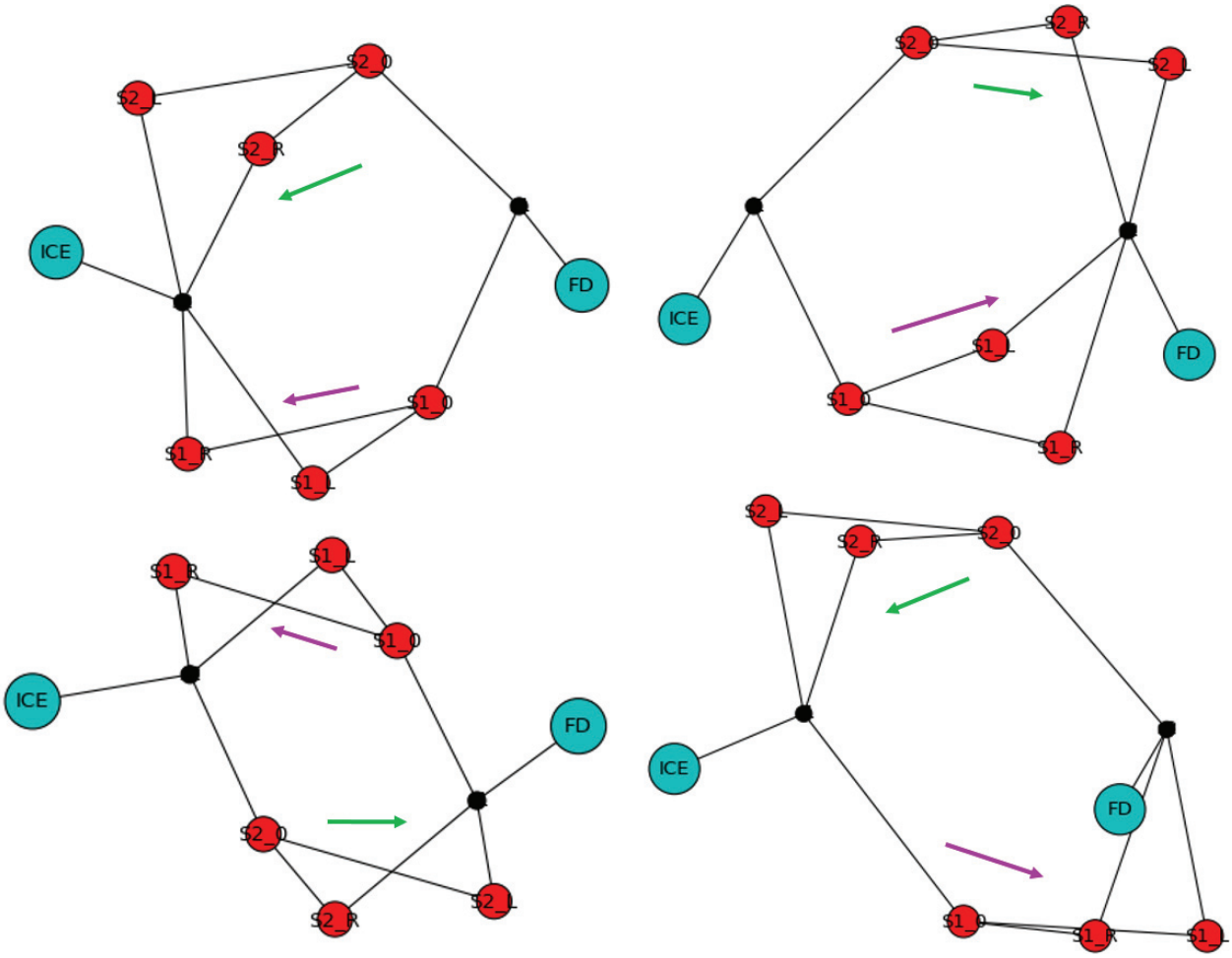


Figure 50: Example on the redundancy

The difference in these 4 graphs is only the orientation of the 'synchro' unit between ICE and FD. The orientation of each unit is shown in the colored arrow, violet for the 'synchro1' and green for 'synchro2'. 4 combinations of orientation exist leading to these 4 graphs. However, the 4 graphs result in a same kinematic linking between the components: disconnection or connection through 4 gear ratios. The 4 graphs will have a same energetic model leading to same energy efficiency. The only small difference might be in their compactness.

A new representation called '0ABC Table' was developed to detect these redundancies. It is a table describing the type of 1-stage connection between all the powertrain components. A 1-stage connection means that a maximum of 2 shafts exist in this connection between the 2 considered powertrain components (Figure 51).

Figure 51: Maximum length of a 1-stage connection

These connections in the graphs can be classified into 4 types:

- 0: no connection between the 2 powertrain components.
- A: direct connection, the 2 powertrain components are connected to 1 same shaft.
- B: connection through synchro placed between 2 shafts.
- C: connection through clutch placed between 2 shafts.

The 4 graphs in Figure 50 have the same 0ABC table shown below in Figure 52:

	0	A	B	C
ICE<->FD	0	0	4	0
MGs<->FD	0 0	0 0	0 0	0 0
ICE<->MGs	0 0	0 0	0 0	0 0
MGs<->MGs	0	0	0	0

Figure 52: 0ABC table of the 4 graphs shown in Figure 50

In the case of graphs having same 0ABC table, the decision is to keep the 1st graph and kill the others. This is the first considered filtering step that resulted in reducing the number of considered solutions. For the example shown in Figure 49, the 132 graphs are reduced to 12 after 0ABC filtering step.

3.4.2. State graphs

The graphs have been generated and filtered. Now it is time to start analyzing them. The first added step is to automatically determine the different states that the architecture can have. In fact, each architecture has gearbox elements that can take different states. More precisely, a clutch if present can have 2 states (open or closed) and a 'synchro' if present can have 3 states (disconnection, connection gear 1, connection gear 2). Therefore, each generated architecture

equipped with $N\_clutches$ and $N\_synchros$ will have $2^{N\_clutches} \times 3^{N\_synchros}$ states. These states will correspond to different operating modes or different gear selections.

The developed tool on Python considers each generated graph as a 'Parent Graph'. Then for each Parent Graph, the $2^{N\_clutches} \times 3^{N\_synchros}$ 'State Graphs' are derived. For each of these state graphs, and based on the components present and the connections between them, the operating mode that corresponds to this architecture state is automatically detected. Nine operating modes exists:

- **ICE only:** only the ICE is connected to the wheels
- **Pure 1MG:** only 1 MG is connected to the wheels
- **Pure 2MG:** 2 MGs are connected to the wheels
- **Parallel 1MG:** ICE and 1 MG are connected to the wheels
- **Parallel 2MG:** ICE and 2 MGs are connected to the wheels
- **Series:** 1 MG is connected to the wheels and the ICE is connected to another MG
- **Neutral:** the wheels are connected to none of the components, no connections between the other powertrain components
- **Stand-still charging 1MG**: the wheels are connected to none of the components, the engine is however connected to 1 MG
- **Stand-still charging 2MG:** the wheels are connected to none of the components, the engine is however connected to 2 MGs

An example of the state graphs generation from a parent graph is shown in Figure 53. 6 different states exist for this parent graph. The corresponding powertrain mode is automatically detected and can be seen on top of the state graphs.

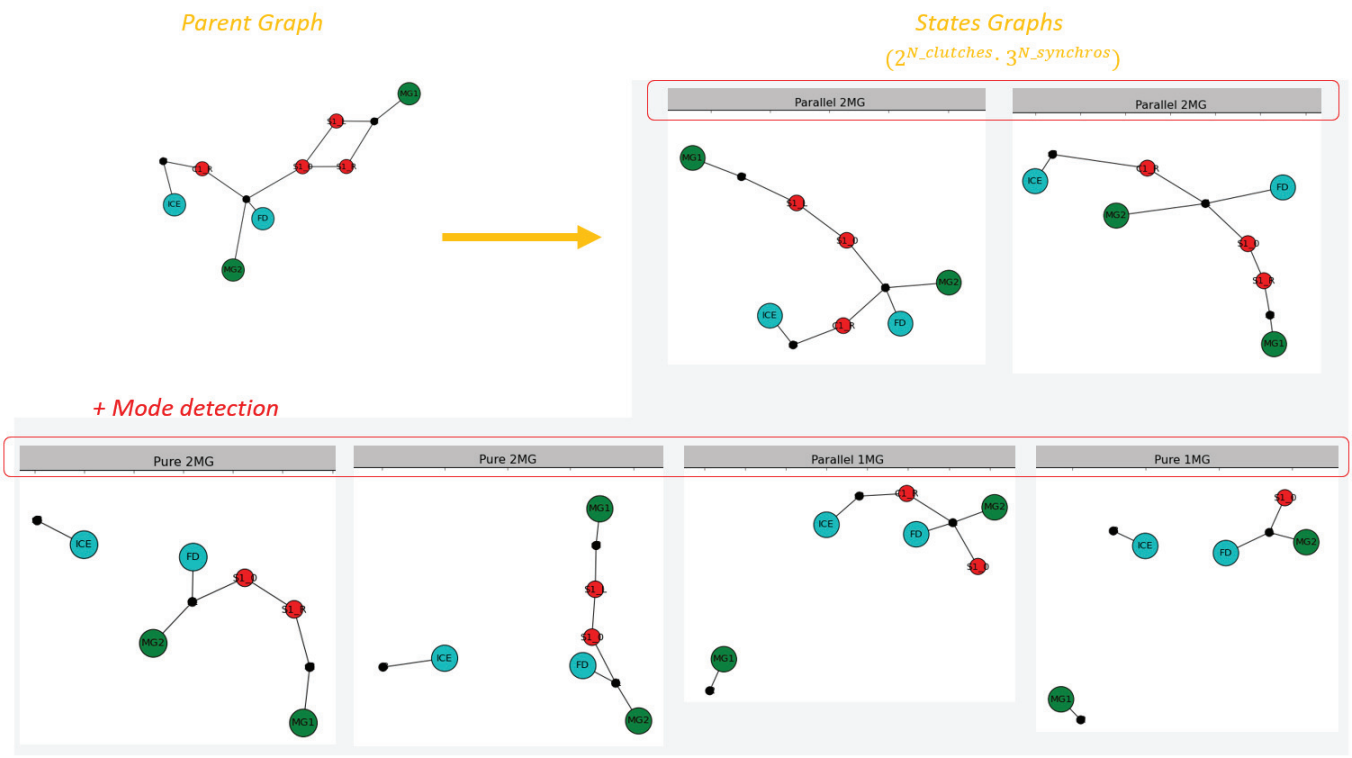
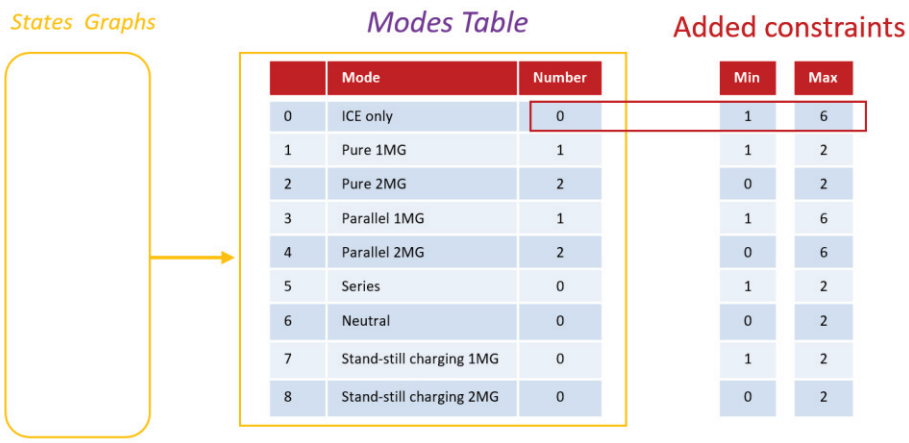


Figure 53: State Graphs generation and modes detection

3.4.3. Modes Table

The modes have been detected for all the state graphs of the parent graph. The next step is to create the 'Modes Table' for each of these parent graphs (Figure 54). This table lists the number of different modes for this parent graph. At this step, the graphs can be compared based on their Modes Table.

In addition, added constraints can be considered here on the minimum and maximum number of instances for each mode. The values of these constraints are discussable and adjustable. An example of values is shown Figure 54. In this example, any architecture not having at least one pure ICE mode to be used in highway driving is to be killed; any architecture not having at least 1 pure electric mode to be used in urban driving is to be killed. This is an example of how constraints can be added. These example constraints are not applied by default in the methodology.



	Mode	Number	Min	Max
0	ICE only	0	1	6
1	Pure 1MG	1	1	2
2	Pure 2MG	2	0	2
3	Parallel 1MG	1	1	6
4	Parallel 2MG	2	0	6
5	Series	0	1	2
6	Neutral	0	0	2
7	Stand-still charging 1MG	0	1	2
8	Stand-still charging 2MG	0	0	2

Figure 54: Modes table for the parent graph in Figure 53

3.4.4. Modes Table +

More information is needed to compare between the architectures linked to the efficiency of the powertrain in each of the listed mode in Modes Table. In addition, the next step in the methodology is to automatically assess and optimize the architectures (it will be presented in next chapter). Hence, energetic models of the architectures will be needed. For those two reasons, it was realized that a description of the power flows between the components is needed at this level.

Each possible power flow between the components will be described using two values:

- The global efficiency of the path
- The global gear ratio of the path

To calculate these values, each node of the state graph was assigned an attribute ratio and efficiency. An example is shown in Figure 55.

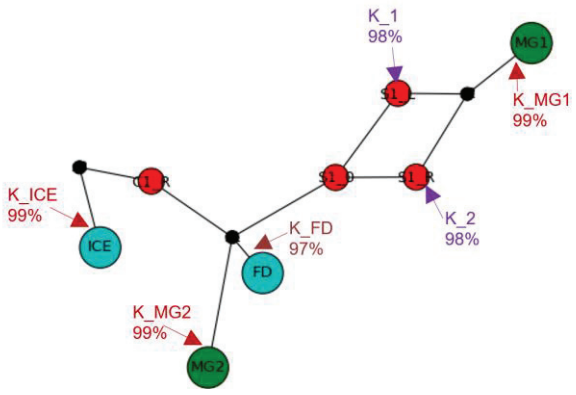


Figure 55: Attributes assignment for the nodes

Then, when a power flow between 2 components is considered, the nodes between these components are listed and the global efficiency and ratio are calculated. The global efficiency is a multiplication of the nodes efficiencies on the path. The global ratio is a multiplication or division by the nodes ratios on the path. A convention for the ratio direction is considered and shown in Figure 56. If the flow is in this considered direction, we multiply by the node ratio. If the flow is opposite to this considered direction, we divide by the node ratio.

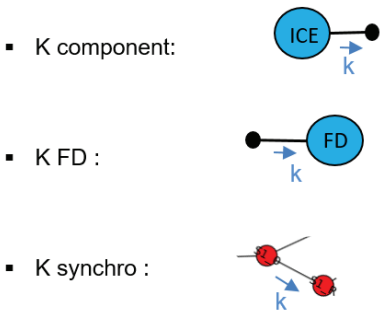


Figure 56: The used convention for ratio direction

These global values are calculated for the power flows between all the components in all the state graphs and visualized. An example on how this information is visualized is presented in Figure 57. In this example, the gears were assigned efficiency values for simplification purpose. In the methodology, all these information will be in function of the nodes efficiency and ratio. These values are chosen later on at the sizing optimization step. If a gear ratio is 1, a 100% efficiency is assigned to the gear. If a gear ratio is different than 1, the efficiency is reduced to a chosen value (98% for example). A corresponding complexity/cost/volume for the gear is also added.

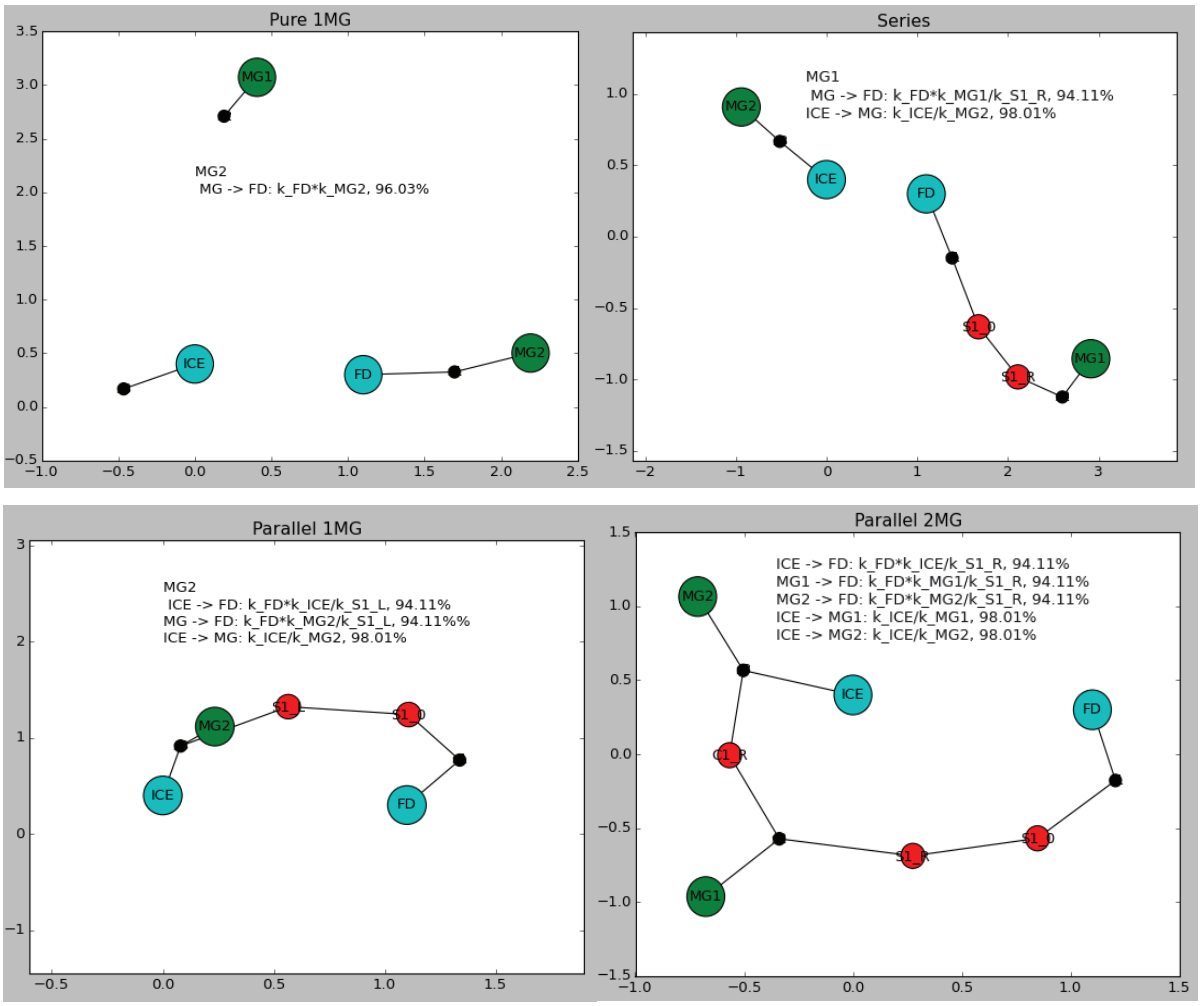


Figure 57: Example of state graphs with the added information to be used in Modes Table +

Finally, this information is collected and listed inside a table called Modes Table + (Figure 58). In this table, the modes available for each architecture are listed. For each mode, the description of the global efficiency and ratio of the power flows between all the components is found.

		Description
	Mode	1st instance
1	ICE only	K_ICE_FD, η_ICE_FD
2	Pure 1MG	MG = 1 or 2 K_MG_FD, η_MG_FD
3	Pure 2MG	K_MG1_FD, η_MG1_FD K_MG2_FD, η_MG2_FD K_MG1_MG2, η_MG1_MG2
4	Parallel 1MG	MG = 1 or 2 K_ICE_FD, η_ICE_FD K_MG_FD, η_MG_FD K_ICE_MG, η_ICE_MG
5	Parallel 2MG	K_ICE_FD, η_ICE_FD K_MG1_FD, η_MG1_FD K_MG2_FD, η_MG2_FD K_ICE_MG1, η_ICE_MG1 K_ICE_MG2, η_ICE_MG2 K_MG1_MG2, η_MG1_MG2
6	Series	MG= 1 or 2 K_MG_FD, η_MG_FD K_ICE_MG, η_ICE_MG
7	Nutral	-
8	Stand-still charging 1MG	MG=1 or 2 K_ICE_MG, η_ICE_MG
9	Stand-still charging 2MG	K_ICE_MG1, η_ICE_MG1 K_ICE_MG2, η_ICE_MG2 K_MG1_MG2, η_MG1_MG2

Figure 58: Modes Table + describing the efficiency and ratio of the power flows in each mode

The details and order of this information are shown Figure 59. For each detected mode, a row is created. The first cell in the row is the mode type name (ICE only, Pure 1MG,…). Then, depending on the mode type, different amount of information is placed in each row. Pairs of cells with same color are found in the rows (Figure 59). These are the ratio (k) and efficiency (η) of a path. In the lower diagram of the general hybrid model, the path in question is highlighted with the same color of the considered cells. For example, for the ICE only row, a pair of cells is found with grey color. One is k and the other is η of the path ICE_FD which has the same color as the cells (grey). For the Pure 1MG, Parallel 1MG and Series mode, the first cell take a 1 or 2 value: 1 if MG1 is connected to the wheels, 2 if MG2 is connected to the wheels.

ICE only	K_ICE_FD	η_ICE_FD	-	-	-	-	-	-	-	-		
Pure 1MG	MG = 1 or 2	K_MG_FD	η_MG_FD	-	-	-	-	-	-	-		
Pure 2MG	K_MG1_FD	η_MG1_FD	K_MG2_FD	η_MG2_FD	K_MG1_MG2	η_MG1_MG2	-	-	-	-		
Parallel 1MG	MG = 1 or 2	K_ICE_FD	η_ICE_FD	K_MG_FD	η_MG_FD	K_ICE_MG	η_ICE_MG	-	-	-		
Parallel 2MG	K_ICE_FD	η_ICE_FD	K_MG1_FD	η_MG1_FD	K_MG2_FD	η_MG2_FD	K_ICE_MG1	η_ICE_MG1	K_ICE_MG2	η_ICE_MG2	K_MG1_MG2	η_MG1_MG2
Series	MG= 1 or 2	K_MG_FD	η_MG_FD	K_ICE_MG	η_ICE_MG	-	-	-	-	-		
Neutral	-	-	-	-	-	-	-	-	-	-		
Stand-still charging 1MG	MG=1 or 2	K_ICE_MG	η_ICE_MG	-	-	-	-	-	-	-		
Stand-still charging 2MG	K_ICE_MG1	η_ICE_MG1	K_ICE_MG2	η_ICE_MG2	K_MG1_MG2	η_MG1_MG2	-	-	-	-		

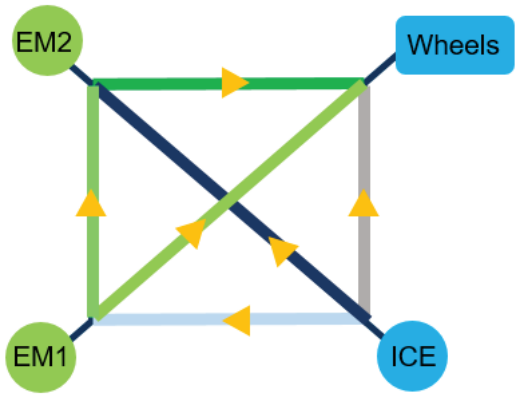


Figure 59: Details of the information found in Modes Table +

Based on the information found in Modes Table +, the architectures can be compared and graded. Based on such grading, the most promising architectures can be selected to continue in the process, while the other architectures are killed. This grading and selecting should be done with unquestionable rules and is a critical step. Killing a wrong architecture at this level can make the methodology lose its optimality. For this reason, the grading and selection is not considered here and all the Modes Table + are considered to continue to process.

Conclusion:

An overview of the works presented in this chapter is given in Figure 60. The aim of this part of the methodology is to explore the architecture level. This is done through an automatic generation and filtering process. The process starts in step 1 by choosing the components to include in the powertrain. In parallel, constraints are pre-defined on how these components can be connected. The chosen components and the pre-defined constraints are injected inside a Constraint Satisfaction Problem solver that generates all the possible graphs of realizable

architectures (Step 2). Each generated ‘Parent graph’ has different states. The inherited ‘State Graphs’ are derived and the corresponding modes are detected (Step 3). The graphs are transformed into 0ABC Tables in Step 4 where redundancies are detected and eliminated. Also in Step 4, for each Parent graph, the modes are listed in a table called Modes Table. Constraints on the minimum and maximum instances of a mode can be added for more filtering of the graphs. In step 5, Modes Table + are created. In these tables, a description of the global efficiency and ratio of the power paths in each mode is added.

The Modes Table + of the architectures are the outcomes of this part of the methodology. These tables include the information needed for the energetic assessment of the architectures. This automatic assessment and optimization of the generated architectures will be the content of the next chapter.

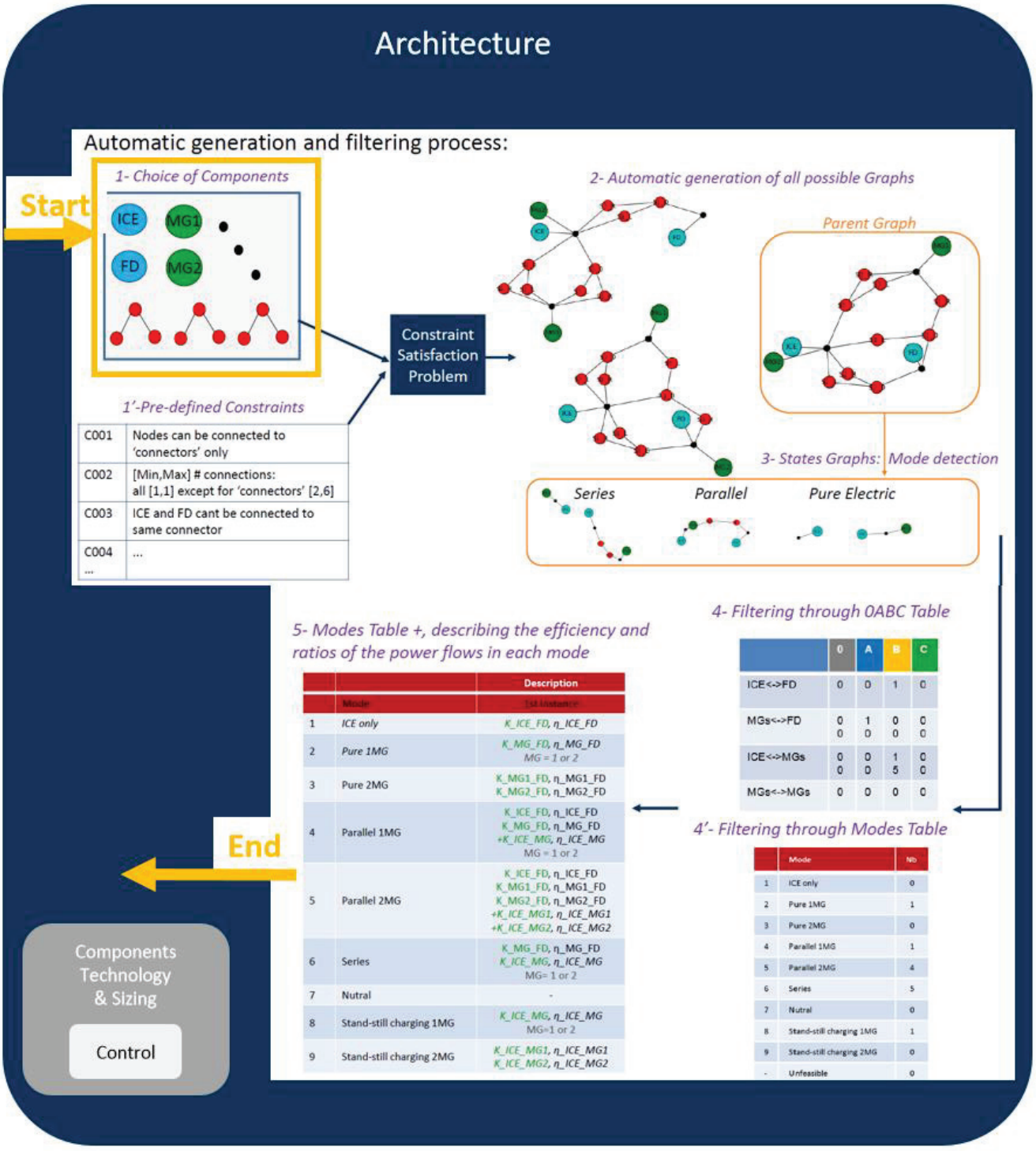


Figure 60: Overview of the works done in the architecture level

CHAPTER 4: AUTOMATIC ASSESSMENT AND OPTIMIZATION OF THE GENERATED ARCHITECTURES

Abstract

Abstract - The generated architectures need to be assessed, optimized and compared. Therefore, energetic models that calculate the fuel consumption and the performance of the powertrains are needed. When the number of assessed architectures is limited, the traditional way is to manually develop one model per architecture. However, when the automatic generation of architectures is included in the methodology, manually modelling all these generated architectures becomes infeasible. To solve this, a general model for (P)HEV is developed. This model is inserted inside an optimization methodology that allows optimizing the sizing and components based on optimal control. An already existing bi-level optimization strategy is selected. Genetic Algorithm GA is used on the sizing and components level, while Dynamic Programming DP is used on the control

level. The capability of the optimization methodology is verified by optimizing and comparing some architectures in the next chapter.

4.1. INTRODUCTION

As seen in the previous chapters, the works on the architecture level consider two approaches. The first approach (Figure 61, left) is to select few known architectures based on previous experience, benchmarking, experts' knowledge and suppliers constraints. The second approach is to explore the architecture level by automatic generation of all the possible architectures that can be realized (Figure 61 , right).

In both cases, the next step is to optimize these selected or generated architectures on the levels of Components Technology & Sizing and Control, and then to compare between them. For this, energetic models for the powertrains of these architectures are needed. In the first approach, the number of assessed architectures is limited, thus the traditional way is to manually develop one model per architecture. However, in the second approach, the number of generated architectures is high and manually modelling all of them becomes infeasible. To solve this, a general model for (P)HEV can be developed and used to assess and optimize all the architectures.

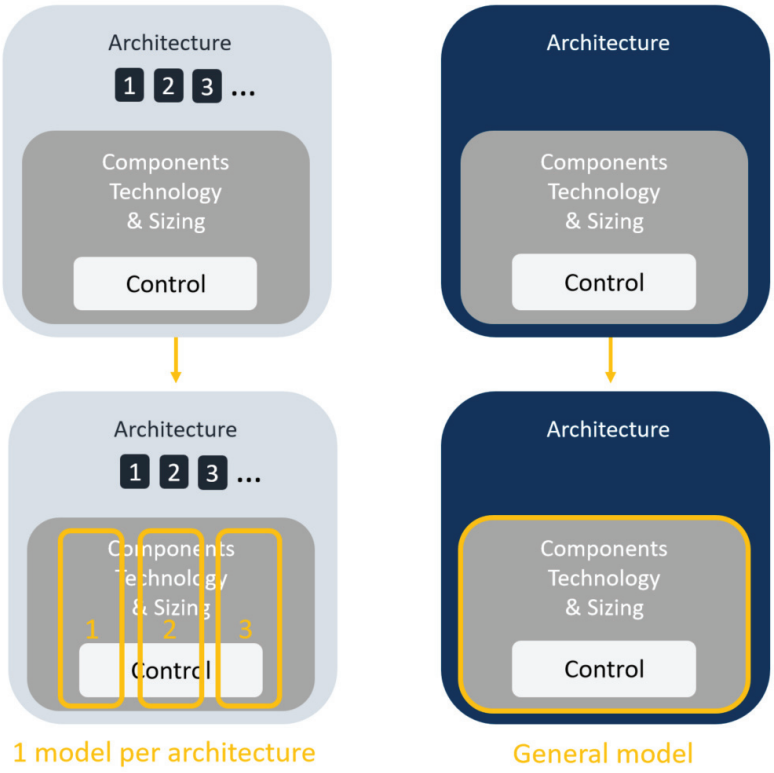


Figure 61: The models needed

The first part of this chapter explains the models used to assess the powertrains, for the case of one model per architecture then for the case of a general model for all the architectures. In the second part of the chapter, the bi-level optimization methodology used to optimize all of these models is explained.

4.2. ASSESSMENT OF THE POWERTRAINS

In order to assess the different powertrains, the vehicle in its environment needs to be modelled.

4.2.1. Powertrain modelling

Energetic models are developed in MATLAB using the VEHLIB [82] library of vehicle components and the longitudinal dynamics law. The backward approach is used to calculate the fuel consumption, while the forward approach is used to calculate the vehicle performance. The vehicle model consists of components models linked by kinematic equations describing the mechanical connections between the components. More precisely, it is the description of how the speed and torque are translated from the components to the wheels (forward) or from the wheels to the components (backward).

The backward approach used to calculate the fuel consumption of the powertrain is shown in Figure 62. The blocs are explained after. EMS stands for the Energy Management Strategy.

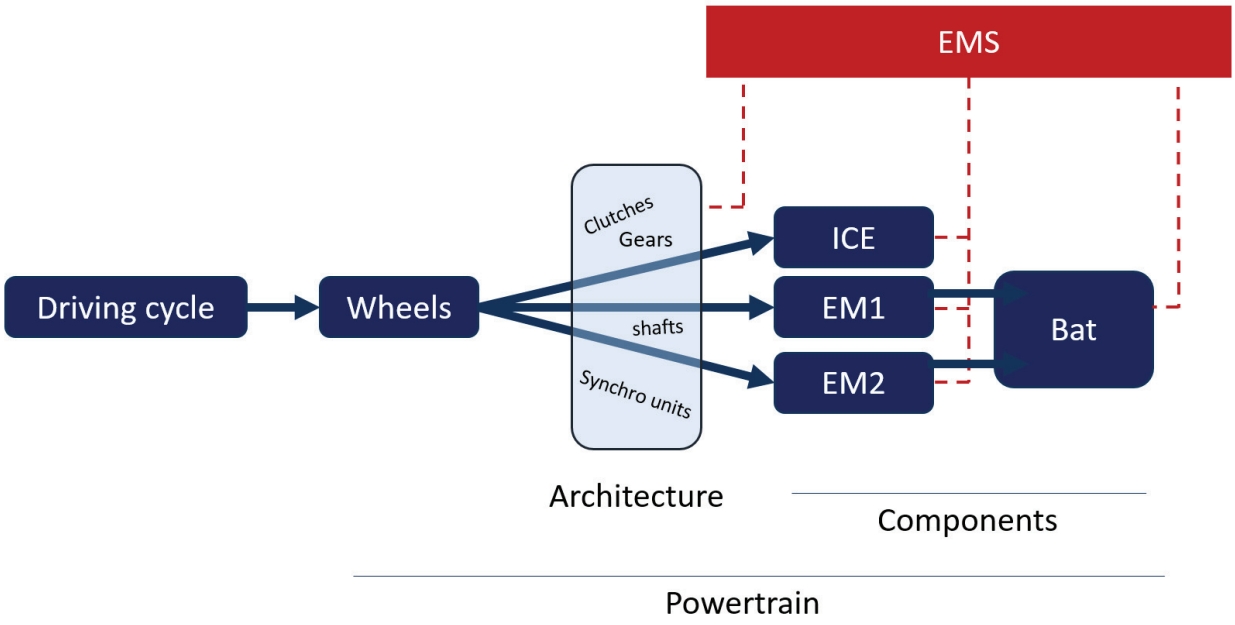


Figure 62: Modelling approach used to calculate the fuel consumption

1) Driving cycle

The system should be evaluated in an environment. To evaluate the powertrain operation and fuel consumption, the vehicle is considered to follow a speed and acceleration demand in function of time. This is called the driving cycle imposed on the vehicle. It is an input to the methodology. Any driving cycle can be inserted and used, for example homologation driving cycles like WLTC. In addition, the methodology considers three real driving conditions: urban, rural road and highway. The ARTEMIS European driving cycles [83] are used to simulate these three conditions. The fuel consumption can be calculated as a weighted sum of the fuel consumptions in these three cycles. The weights are calculated from the yearly mean traveled distance in these conditions.

The driving cycle speed and acceleration demands are then transformed into rotational speed and torque demand on the wheels via the vehicle longitudinal model.

2) Powertrain architecture

Physically, the powertrain architecture bloc contains all the connecting elements and gearbox elements that are shaping the connection between the wheels and the components. This

can include links, shafts, clutches, gears, gearbox, synchro units…Some of these elements are actuators and thus have different states and they need control (EMS intervention).

From the modelling point of view, this bloc contains the kinematic equations governing the translation of the speed and torque from the wheels to the components. In the one model per architecture approach, these equations are defined by describing the architecture scheme, going from the wheels to the components, step by step. The clutch, gears, and gearbox models are added when these elements are encountered. In the general model case, these equations will be a simple description of the virtual global power paths between the different components. This will be explained in details in section 4.2.2.

3) Powertrain components

The speed and torque demands are transferred to the components. Models for the components are needed here. They are gathered from the VEHLIB [82] library. Experimental Brake-Specific Fuel Consumption (BSFC) maps are used for the engine. The electric machines and their converters are modelled using losses maps. The losses of the EMs are generated from reluctance network models that were validated by a finite element model [84] and experimental data [85]. The losses of the inverters are calculated using an analytical model based on inverter parameters (switching losses, switching frequency, recovery charge,…). The global losses (EMs + inverters) will be considered to be independent from the battery voltage. A classical equivalent electric circuit model is used for the battery which is considered to be modules in series, for simplicity reason. The choice of series or parallel configuration for the battery modules will not affect the converters and EMs losses. Gear losses are modelled by assigning a constant efficiency to each gear.

The sizing of these components is not fixed and will be optimized. The involved sizing variables can be:

- the maximum power of the ICE, of EM1 and of EM2
- the number of battery modules in series
- the ratios of the gears, if existing
- the ratios of the gearboxes, if existing

When the sizing variables are chosen, the component characteristics are updated. A scaling technique is performed for the ICE and EMs. The scaling factor is the ratio between the sized power and a reference power. The maximum torque curves, the fuel consumption, the power losses maps, and the weight are multiplied by this factor. The inertia is multiplied by this factor to the power 5/3. For the EMs, the thermal boundaries and the geometry were not considered here but were considered in [86]. Concerning the battery, the sizing is performed by changing the number of cells in series. The voltage, maximum power, stored energy and weight are recomputed. The influence of the battery voltage on the EM and converter losses is neglected. For the gears, their ratios are replaced by their sized values, without changing their efficiency.

4) Energy Management Strategy EMS

The powertrain model has degrees of freedom (DoF) because it has more than one energy converter that can provide the needed power to the wheels and because it has some connecting elements that can take different states.

A high level control supervises the components and the connecting elements and decides the corresponding control variables during the vehicle operation (Figure 62). This control is referred to as Energy Management Strategy.

The involved control variables can be:

- Choice of the operating mode: induced by the state of the clutches and state of the gearbox and synchro units (disconnection or connection)
- Gear selection for a mode: induced by the selection of gears in the gearbox or synchro unit.
- Power sharing between the components

According to the chosen values for the control variables, the resulting operation of the components is determined: the fuel consumption in the engine, the losses of the machines, and the discharge/charge in the battery.

4.2.2. General model for all the generated architectures

In the second approach of the works done on the architecture level (Figure 61, right), the number of generated architectures is high. Manually modelling all the architectures is infeasible. Alternatives are needed.

1) Literature works

Two methods are found in the literature. The first is to do automatic modelling of the generated architectures. This is the procedure adopted in [87]. Knowing the graph of the architecture, the methodology lists the equations of kinematic relations between the model variables, then determines the DoF (degree of freedom) variables to be controlled, and finally orders the equations into a model: on each line, the variables only depends on the previously defined variables (Figure 63). This can be seen as an automation of the manual modelling that is done in the first approach.

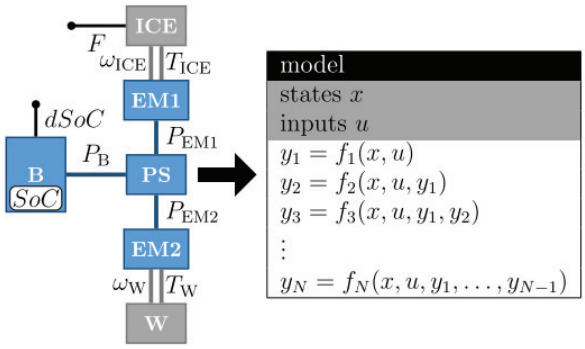


Figure 63: Automatic modelling of a graph, methodology found in [87]

The second method is to develop a generic model having parameters that are automatically determined for each architecture. This approach is adopted in [36] where a method is developed to automatically determine the parameters of a general transmission model (Figure 64). To determine these parameters, the authors list the set of linear equations that describe the kinetic relations of the architecture. Significant analysis and processing work is done on the matrices generated by these equations. The works finish by detecting and classifying the feasible, non-redundant modes of the architecture and the parameters of the generic transmission model. The parameter determination is performed analytically, resulting in low computational time (less than half a second). A case study on two architectures is done in [36]. For the architectures optimization, no sizing optimization is done. Control optimization is performed using DP with 2

states variables. Losses of gears, clutches, and brakes are neglected. Around 5 hours were required for the control optimization of a fixed sizing for a 5-modes architecture (including power-split mode).

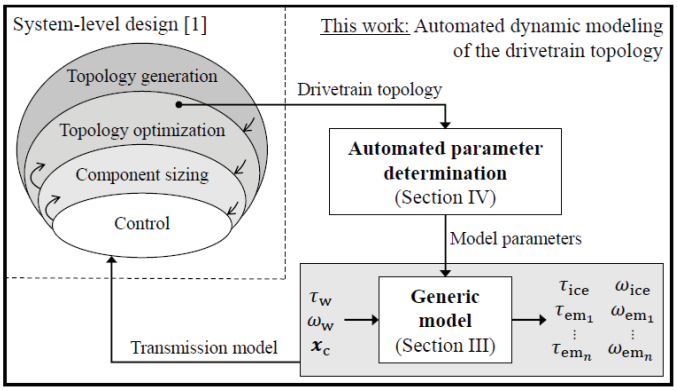


Figure 64: The generic transmission model and parameters determination in [36]

Similar work on system matrix generation and filtering is also made in [66][48].

2) Proposed method

In the case of our methodology, any generated architecture will have a combination of 9 possible modes: ICE only, Pure 1MG, Pure 2MG, Parallel 1MG, Parallel 2MG, Series, Neutral, Stand-still charging 1MG, Stand-still charging 2MG.

Knowing this, we consider that the complex work of automatically modelling the architectures can be avoided by developing a general model for the (P)HEV. This model can be used to simulate and optimize all the generated architectures.

The generic model has different functions used to calculate the performance and the fuel consumption ('Performance Function', 'Limits Function', and 'Edge Cost Function'). They will be explained in 4.3. In each of these functions, a call is made for all the available modes in the architecture. Modes models are developed for the 9 possible modes. Whenever a mode is present in an architecture, its model is called with its corresponding information (global ratios and efficiencies) found in the Modes Table + presented in the previous chapter. In case two instances of a same mode are present, the mode model is called twice, each time with different information. An example on how the modes models are called inside the general hybrid model for the

'Performance Function' (function that evaluate the maximum dynamic performance of the vehicle) is shown in Figure 65.

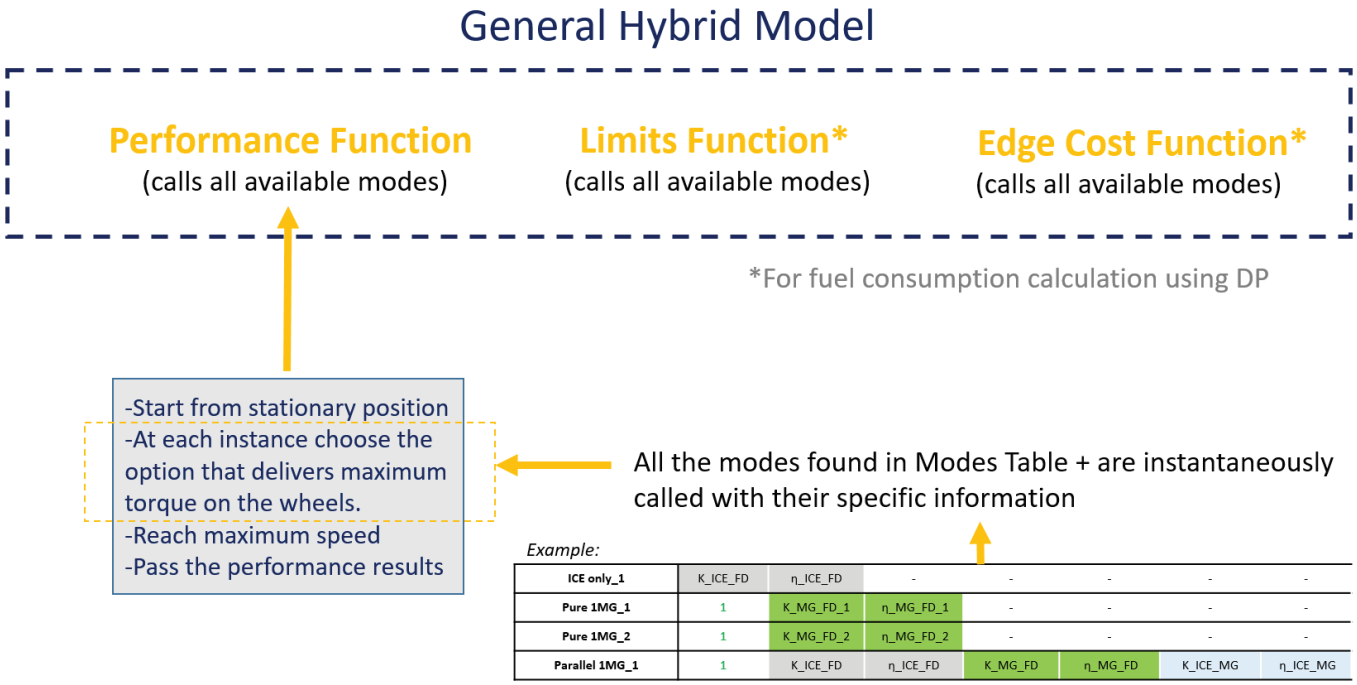


Figure 65: Example on how the modes models are called inside the general hybrid model, performance function

In contrast with [36] where a generic transmission model is developed, in this PhD the notion of real transmission is removed in the general hybrid model (Figure 66). Examples on the mode models of this general hybrid model are shown in Figure 67. In each mode model, no real transmission exist. All the components are connected by simplified virtual links that have a gear ratio and an efficiency. These are the global ratio and efficiency of the considered power path. This information is found in the Modes Table + of the architecture (detailed in chapter 3). Each of the possible modes in the architecture is a function called using this information.

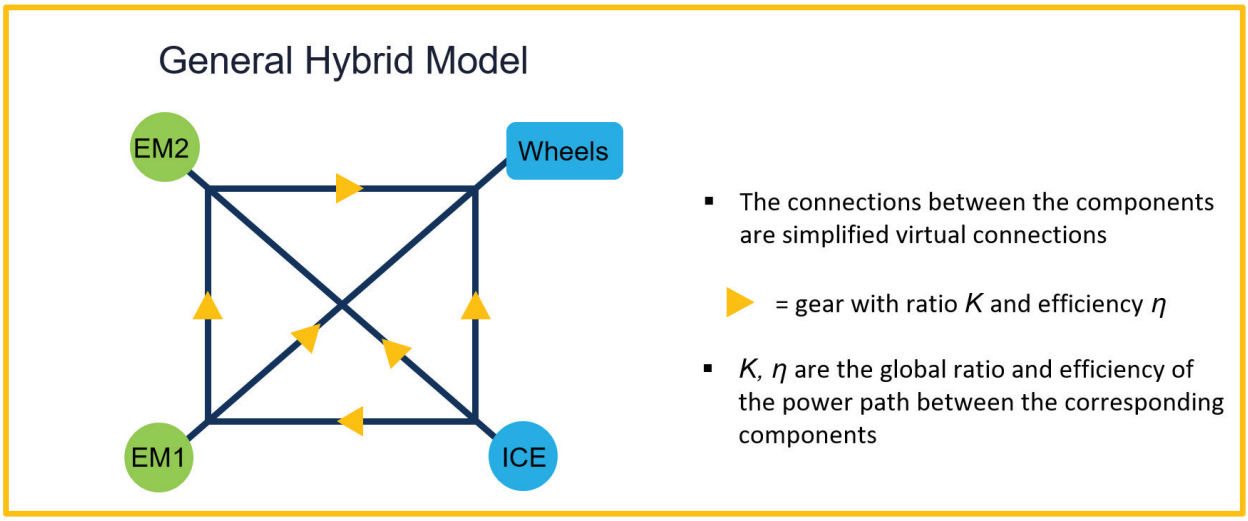


Figure 66: The developed General Hybrid Model in this PhD

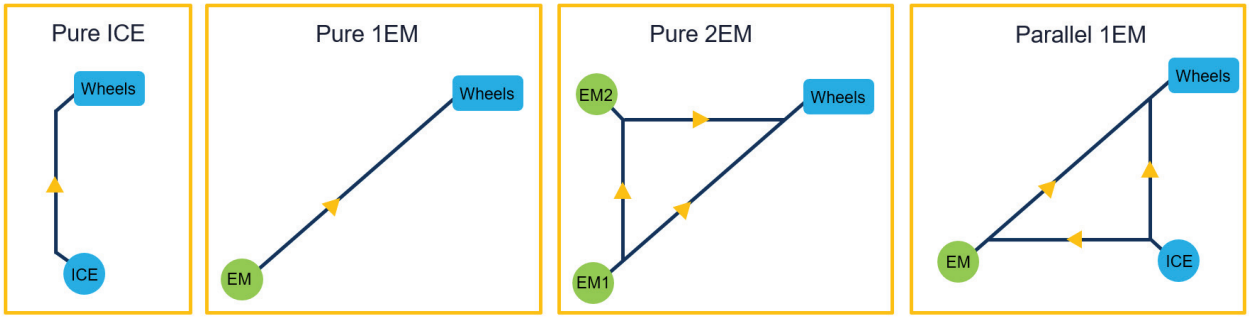


Figure 67: Examples on the mode models

The linking between the automatic generation of architectures and the general model is done through the Modes Table + (Figure 68).

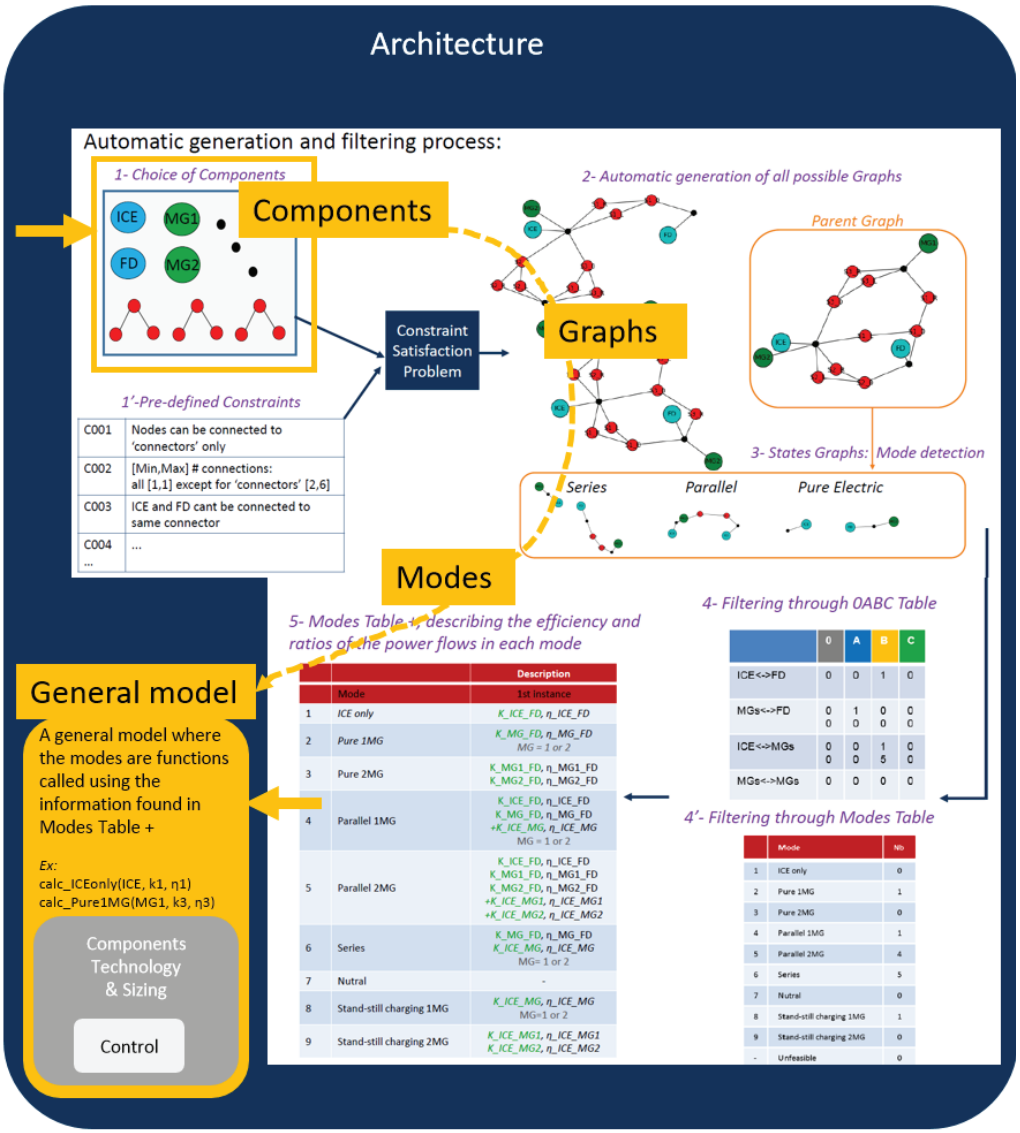
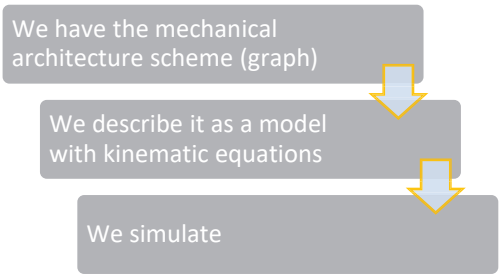


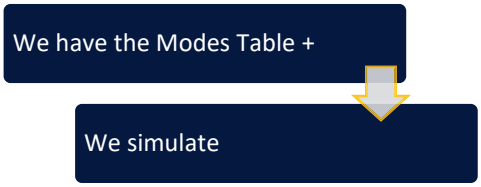
Figure 68: Connection between the automatic generation of architectures and the general model

3) *Difference with the commonly used models*

This general model is done with a novel perspective in modeling, functional more than physical. In the previously developed models by the authors and in the commonly found models in the literature, the connections between the components (kinematic equations) are written by describing the physical mechanical architecture scheme. The paradigm would be:



With the developed general model, the description of these physical connections is not needed. There is no more need to know the mechanical architecture scheme. Knowing the Modes Table + of an architectures is now enough to simulate this architecture.



This is done through an unusual torque translation inside the powertrain. In this general hybrid model, the torque provided by a component is divided into different portions, each toward a destination (Figure 66). For example, the engine delivered torque is divided into a portion going to the wheels, a portion going to EM1 and a portion going to EM2. Each portion will be multiplied by the ratios and efficiencies of the path it takes.

4) Validation of the general hybrid model

This general hybrid model was validated by doing simulation and optimization of architectures using their old hand-developed model first, then using the general hybrid model, and comparing the results.

An example is shown in Figure 69. An architecture called SPHEV 1 was optimized in [53] using a specific model for this architecture. The simulation and optimization parameters can be found in [53]. The architecture was then optimized using the general hybrid model and the Modes Table + of SPHEV1. The same simulation and optimization parameters were used. The resulting Pareto fronts of the 2 considered objective functions are shown in Figure 69. The results coincide, validating the correctness of the general hybrid model for the SPHEV1 case. In a same

manner, the general model was validated for the previously studied architectures in [88] [89] [53].

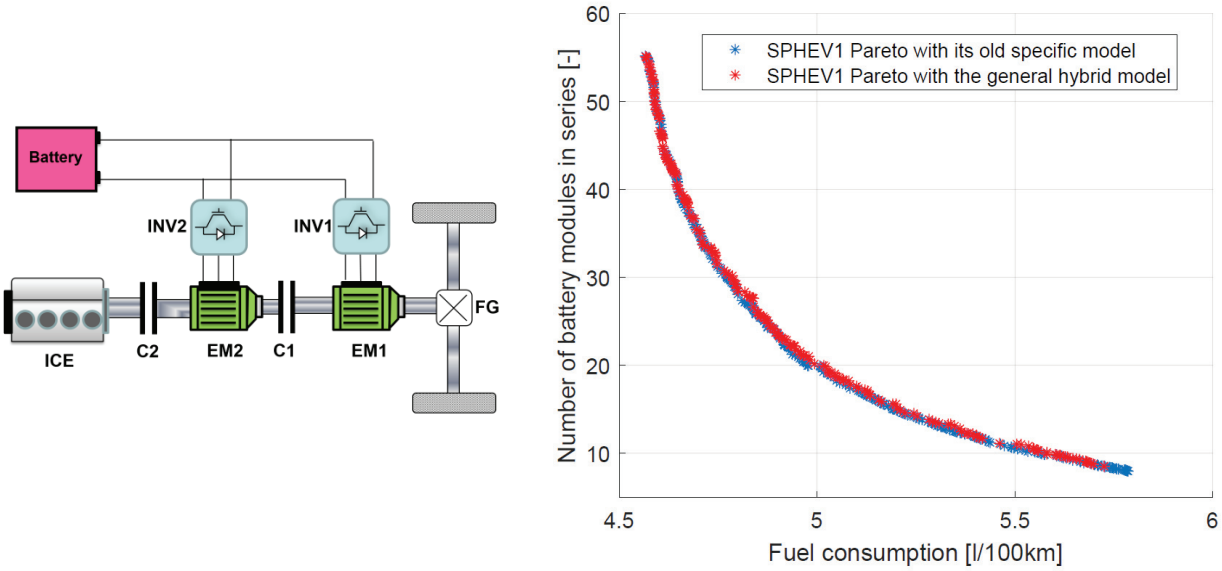


Figure 69: Example on the validation of the general hybrid model, case of SPHEV1 architecture

4.3. BI-LEVEL OPTIMIZATION OF THE POWERTRAINS

To guarantee a fair comparison between the architectures, an optimization process is performed for each of them to ensure that they are compared based on their optimal potential. The adopted optimization methodology is described in this section.

As explained before, the vehicle model has components and sizing variables that need to be fixed when the powertrain is being designed, and control variables that need to be fixed instantaneously when the powertrain is being operated. A bi-level optimization process (Figure 70) is adopted [89], [90], [32] following the nested approach (presented chapter 2): an upper level optimization of the components technology & sizing (the design) and a lower level control optimization (for the operation). This process yields to a Pareto front presenting the tradeoff between the objective functions.

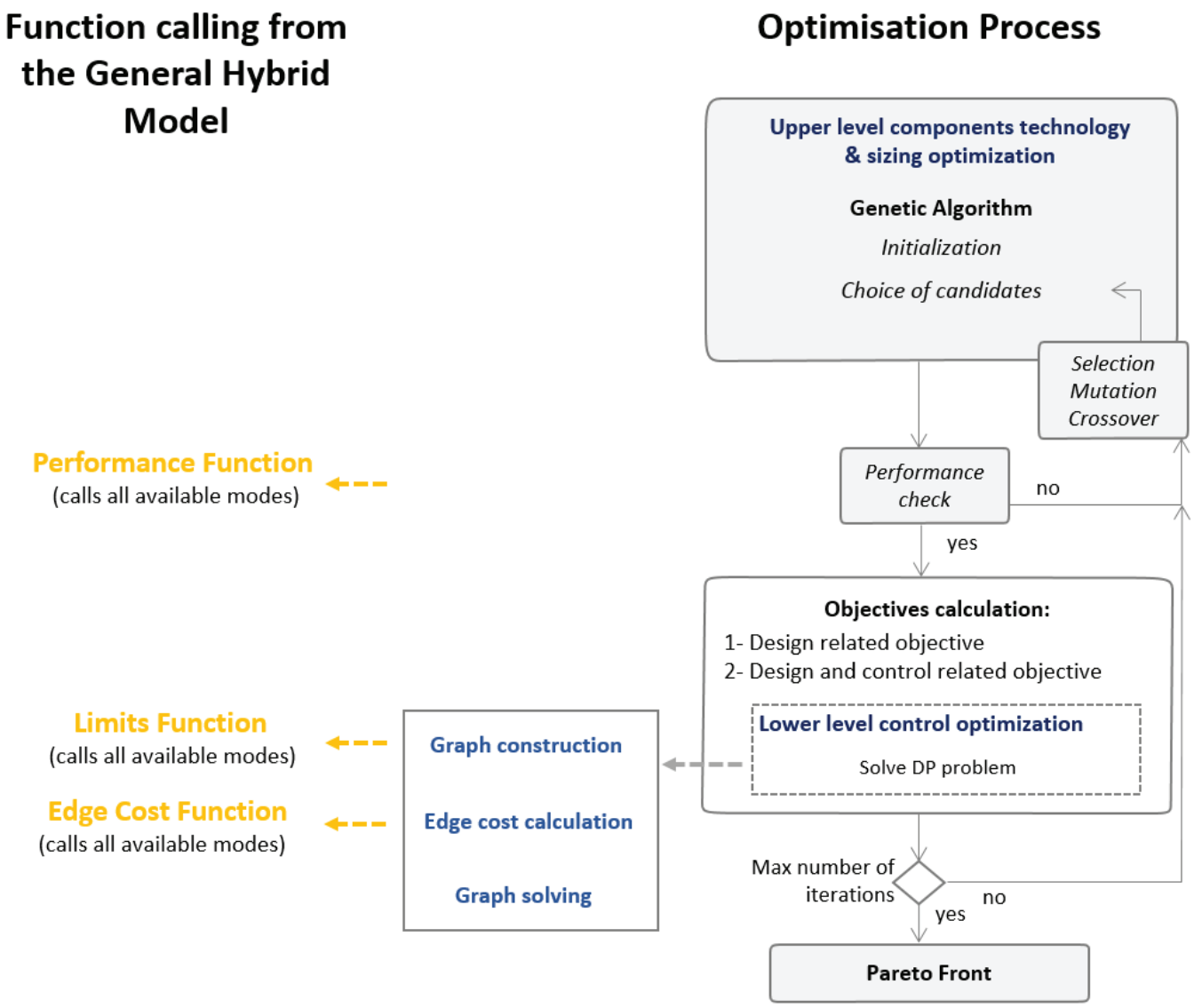


Figure 70: Optimization process and function calling from the General Hybrid Model

4.3.1 Upper level components technology & sizing optimization

The optimization variables on this level are the components choice variables and the sizing variables of the powertrain. At the current stage of the works, the choice of components was fixed (engine and electric machines technologies, battery cells type…) and the variables in this level were limited to the sizing variables. The freedom in components choice is to be added in future works.

For an architecture that can have up to N sizing variables, the design space of this optimization will have N dimensions. The genetic algorithm (GA) NSGA-II [91] is used as the optimization algorithm. GA performs the exploration of this design space and generates at the end of the optimization the optimal tradeoff (Pareto fronts) between the objective functions.

The algorithm chooses at each step the sizing candidates to be assessed (set of values for the sizing variables). GA initializes a random population of sizing candidates. After an evaluation of the objective functions of this population, GA performs an evolution process (Selection, Crossover, and Mutation) to produce better generations. For each generated individual, the performance constraint is checked. If it is respected, the objectives are assessed. The process is repeated until a maximum number of iterations is reached. More explanations can be found in [91]. In previous works, the Pareto fronts were compared from 200 to 1000 generations for different optimizations. No change in the Pareto was seen after 500 generations. Thus, no more than 500 generations will be considered.

In the performance check, a test is performed to ensure that the sizing candidate respects the constraints. The methodology allows constraining four indicators:

- The acceleration time from 0 to 100 km/h:
  $t_{0->100} < value$
- Acceleration time from 80 to 120 km/h:
  $t_{80->120} < value$
- The maximum speed of the vehicle on a flat road: $V_{max} > value$
- Minimum electric range > *value*

To calculate the first three indicators (2 accelerations and maximum speed), the ‘Performance Function’ is called. The working principle of this function is shown in Figure 65. The vehicle starts from a stationary position. Then at each instant, the function calls all the available modes to calculate for each of them the torque that can be provided to the wheels. The mode and mode instance that maximizes the torque on the wheels is chosen. For the parallel hybrid mode, clutch slipping is allowed when the engine speed is below a threshold value, which is higher than the idle speed. At the vehicle maximum speed in the performance test, the battery power is not forced to be null.

The 4th indicator (the minimum electric range) is calculated by running the vehicle in pure electric operation on WLTC driving cycle.

For the objective functions, a choice of objectives exists:

- Design related objectives:
    - **The number of battery modules in series:**

This is an indicator of the battery stored energy and is one of the sizing variables. This objective function reflects the amount of electrification. It has an influence on the energy management and thus the fuel consumption. A high battery size leads to reduction in fuel consumption because the powertrain can recuperate more energy when needed and has more energy freedom when choosing between electric and hybrid modes. In addition, the importance of the architecture on the fuel consumption might decrease when the battery size is big.

- **Powertrain Cost index PCost:**

In order to have a representation of the cost of the powertrain induced by the sizing process, an index called Powertrain Cost index PCost is created and calculated as follows:

$$PCost = Cost_{ICE} + Cost_{EMs,conv} + Cost_{HVB} + Cost_{GB}$$

$$Cost_{ICE} = \alpha_0 + \alpha_1 \cdot (P_{ICE} + \alpha_2 \cdot T_{ICE})$$

$$Cost_{EMs,conv} = \beta_1 \cdot P_{EMs}$$

$$Cost_{HVB} = \gamma_0 + \gamma_1 \cdot N_{bat}$$

$$Cost_{GB} = \delta_0 + \delta_1 . N_{clutches} + \delta_2 . N_{synchro\ units} + \delta_3 . N_{gears}$$

PCost index is equal to the summation of:

- $Cost_{ICE}$: the cost of ICE which is calculated from its maximum power P_{ICE} and maximum torque T_{ICE} (based on Groupe PSA internal formulas)
- $Cost_{EMs}$: the cost of the EMs with their converters. These costs are assumed to be directly proportional to the maximum power P_{EMs} [92].

- $Cost_{HVB}$: the battery cost assumed to be equal to an initial cost, plus a cost proportional to the battery stored energy (kWh) calculated from the number of battery modules N_{bat}.
- $Cost_{GB}$: the cost of the gearbox elements, assumed to be equal to an initial cost for the differential, final drive and shafts, plus an added cost per added clutch, added cost per added synchro unit, and added cost per added component gear.

The values of the α , β , γ, and δ coefficients are based on assumptions, Groupe PSA internal data and literature readings [92] [93].

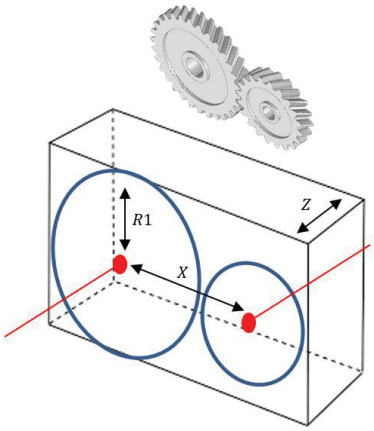
- Powertrain Compactness index PCompactness:

In order to have a representation of the compactness of the powertrain, an index called Powertrain Compactness index PCompactness is created and calculated as follows based on Groupe PSA formulas:

$$PCompactness = V_{ICE} + V_{EM} + V_{GB}$$

$$V_{ICE}(L) = P_{max,ICE}\ (\text{kW}) \,.\, a_1 . a_2$$
$$V_{EM}(L) = P_{max,EM}(\text{kW}) . b_1 + b_0$$
$$V_{GB}(L) = \sum V_{Gpair}(L)$$

$\boldsymbol{V_{Gpair}}$ for a pair of gear with a ratio 'Ratio':



$$V_{Gpair}(L) = (2X) . 2R_1 . Z . 10^{-6}$$

$$X\ (mm) = (11.5) \,.\, \sqrt[3]{C_{max}}$$

$$R1\ (mm) = \frac{X}{1+\frac{1}{Ratio}}$$

$$Z\ (mm) = \frac{2.34^2 \,.\, \cos\alpha_h \,.\, C_{max}}{R_e \,.\, R1 \,.\, M}$$

The values of the a and b are based on regressions done using internal data from Groupe PSA. α_h helix angle, R_e yield strength, and M module are gathered from the literature. C_{max} is the maximum input torque.

- Design and control related objectives that cannot be calculated without simulating the vehicle operation on a chosen driving cycle:
    - The charge sustaining fuel consumption:

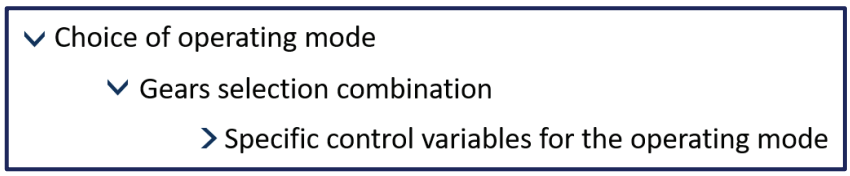
To calculate this fuel consumption, the vehicle is run on a driving cycle. Here, a control optimization problem arises. It is explained in the following section 4.3.2

4.3.2 Lower level control optimization

The fuel consumption at the end of the drive cycle is strongly dependent on the instantaneous choice of control variables during the entire vehicle operation. The control problem is defined as follows:

- The powertrain has 1 state variable: the battery SOC.
- The powertrain state has constraints: initial state, final state, minimum and maximum boundaries.
- The powertrain has different control variables that need to be decided instantaneously in a way to reach the final state with the minimum cumulative fuel consumption.

These different control variables are:



The first level control variable is the choice of operating mode. In fact, a given speed and power demand of the vehicle can be accomplished by the powertrain through different possible operating modes (maximum of 9). For each operating mode, if synchro units are involved, different instances of the mode exist corresponding to each gear selection combination. This is the second level control variable. Finally, specific control variables might also exist for each mode. They are listed in Table 5.

Table 5: Modes specific control variables

Mode	Specific control variables
ICE only	none
Pure 1MG	none
Pure 2MG	1 power sharing variable between EM1 and EM2
Parallel 1MG	1 power sharing variable between ICE and EM
Parallel 2MG	2 power sharing variables between ICE, EM1 and EM2
Series	1 power sharing between Pbat and ICE speed
Neutral	none
Stand-still charging 1MG	none
Stand-still charging 2MG	1 power sharing variable between EM1 and EM2

For the electric mode 2 and the parallel hybrid mode, the control variables are the power sharing between the involved components. In the series mode, the control variables are the battery power and the speed of ICE. For a given battery power and a given power in propelling EM, the ICE power is known. The engine speed is discretized and the value that optimizes the ICE-EM2 system is chosen.

This control problem can be solved by different methods, or Energy Management Strategies (EMS). They differ in their optimality, computation time and ability to be implemented or not in real time vehicle operation. In this work, the chosen EMS is the Dynamic Programming (DP) which guarantees the global optimal fuel consumption of each candidate on the entire cycle [38], [94], [95]. The choice of DP implies that the architectures are compared based on their optimal potential of fuel saving and preserves consequently the fairness of comparison. A time step of 1 second is used and the SOC is discretized with around 1000 points between the minimum SOC and maximum SOC. More details can be found in [39].

The calculation of DP in the general hybrid model is explained in Figure 71. The control problem can be represented as a graph of battery SOC versus time. This graph is regularly sampled, with a time step t and SOC step dSOC. The graph is also limited by the maximum SOC and minimum SOC allowed in the battery. The points in 2 consecutive columns of this graph are connected through an edge associated with a cost that is the fuel consumption corresponding to

this edge.

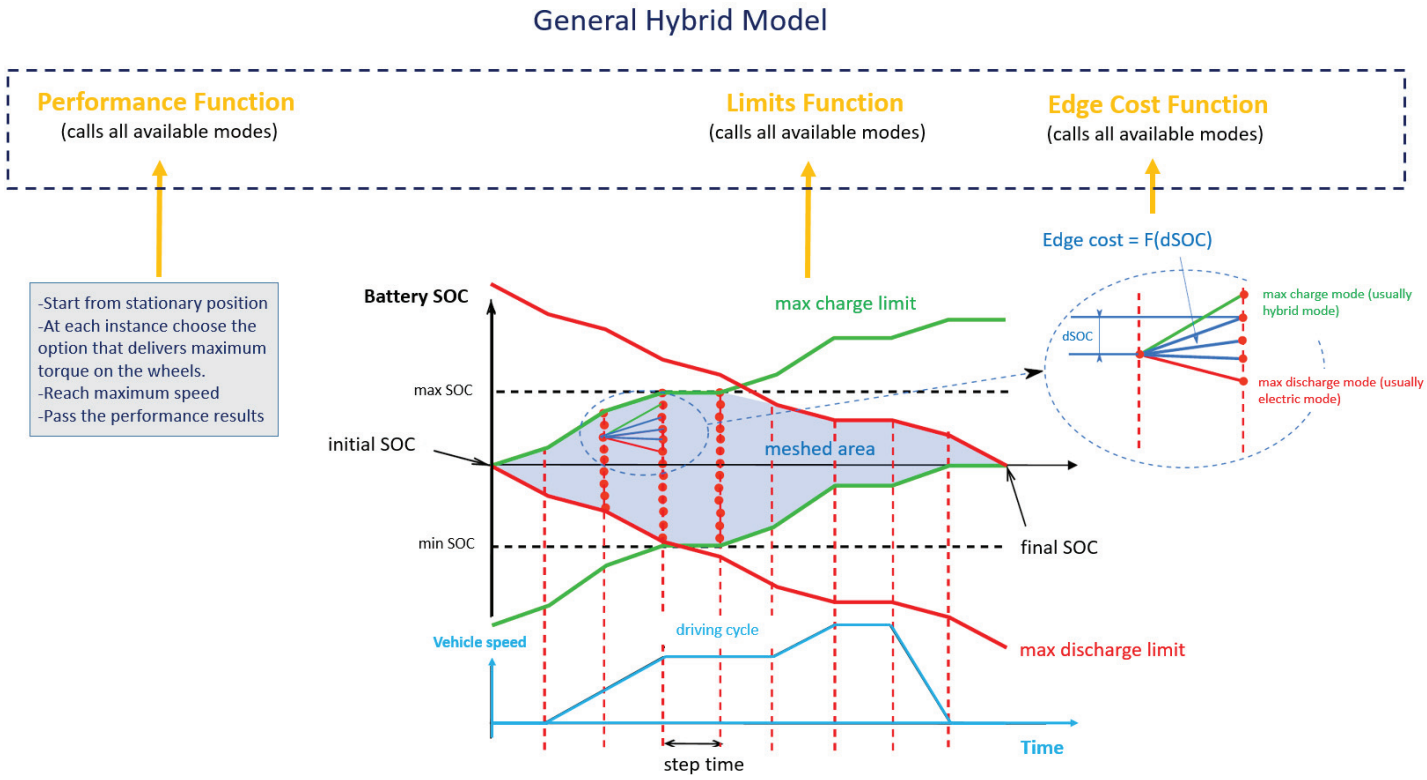


Figure 71: The calculation of DP inside the general hybrid model

DP starts by construction this graph and by determining the max charge limit and max discharge limit shown in green and red in Figure 71. The 'Limits Function' of the general hybrid model is used here. Inside this function all the modes are called and the ones that will lead to a maximum discharge or charge of the battery are chosen. The graph is then limited to a stricter area between these limits, called meshed area in Figure 71.

The second step is to calculate the cost of all the edges. For this, 'Edge Cost Function' is used. It calls all the modes and mode instances that can allow the vehicle to go to this edge. For each mode, a fuel consumption is calculated after having optimized the specific control variables inside the mode. The edge cost selected (and consequently the corresponding mode) will be the minimum of these fuel consumptions.

Once all the edge costs are known, DP will determine the optimal trajectory allowing the system to go from the initial SOC to the final SOC with the minimal global fuel consumption.

This global optimal fuel consumption calculated by DP is sent back to the upper level where the genetic algorithm continues the optimization process until a maximum number of generations is reached. If we consider two particular objective functions, a Pareto front of the fuel consumption versus the number of battery modules can be plotted for each architecture. This is presented in the following chapter.

Conclusion:

The generated or selected architectures need to be assessed, optimized and compared. For this, the powertrains should be modeled and the models should be inserted inside optimization loops. This chapter presented this modelling and the bi-level optimization methodology used. The capability of the optimization methodology will be verified by optimizing and comparing some architectures in the next chapter.

CHAPTER 5: APPLICATION AND RESULTS

Abstract

Conclusion

Abstract - In this chapter, an application of the entire methodology is performed to show its capability. Starting from the powertrain components of a simple SPHEV powertrain, all the feasible architectures are generated and filtered. The most promising are automatically optimized and compared.

5.1. THE INTEREST IN SPHEV ARCHITECTURES

According to [28], series-parallel (SPHEV) architectures benefit from the advantages of series and parallel, but they have relatively a more expensive design and complicated control. The most common SPHEV powertrain is the power-split that uses a Planetary Gear (PG) as a power-split unit. The Toyota Prius was the first adopter of this architecture in the Toyota Hybrid System (THS) [30], (Figure 72). Chevrolet Volt and Opel Ampera also adopted a power-split architecture [31].

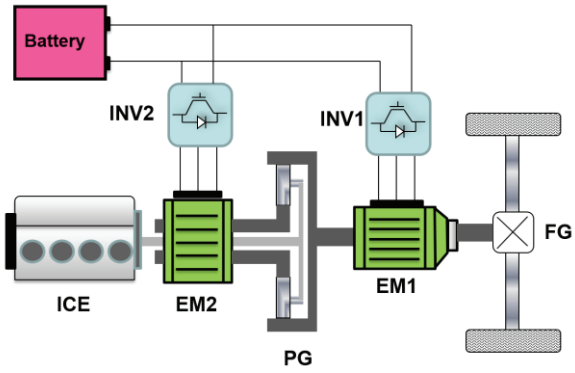


Figure 72: THS power-split architecture

In power-split, moving the ICE operation in its high efficiency area is made simpler by the presence of a speed degree of freedom (DoF), in addition to the torque DoF present in any parallel architecture. This speed DoF is made possible by the PG, which decouples the ICE speed from the wheel speed. The cost of this double freedom is that the power going to the wheels is always split between a parallel path and a series path, the latter having a lower efficiency due to the added energy conversion stages. Still, power-split SPHEV remain one of the most efficient mass produced HEV [32], [8] and the THS is widely used as a reference powertrain.

Nevertheless, other SPHEVs can be realized without a planetary gear system [97]. A simple SPHEV powertrain with no PG was studied in [33] (Figure 73). It consists of 2 Electric Machines (EM) mounted on the ICE shaft and separated by a clutch. It is relatively a simple architecture that allows vehicle operation in pure electric, series hybrid or parallel hybrid mode. The switching between the modes is done through clutch engaging or disengaging. In contrast to power-split SPHEV, the powertrain does not operate in power-split between series and parallel modes. Its advantage is the possibility to operate in pure parallel mode, avoiding the losses of the series path.

On the other hand, it is more difficult to operate the system in its best efficiency areas compared to power-split SPHEV because the speed DoF is only available in the series mode. This simple SPHEV architecture has thus been proven to be less efficient than the power-split SPHEV [32]. Still, the addition of gears or a gearbox to this simple architecture can improve its efficiency.

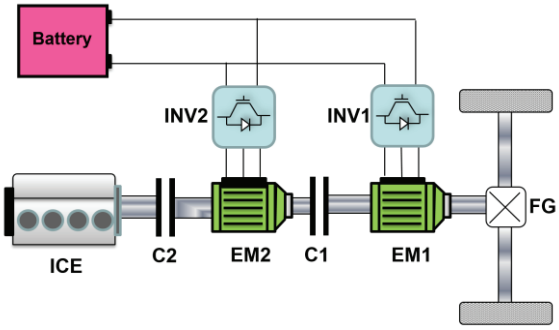


Figure 73: SPHEV 1 architecture

5.2. OBJECTIVES

The objective of this chapter is to search the Architecture design space in the aim of discovering new efficient SPHEV architectures with simple transmissions. To do this, the entire automatic methodology presented in chapter 4 will be used.

The searching process was done in a non-automatic way in [53] where the potential of improvement of the simple SPHEV powertrain is investigated by doing topology modifications. The work starts from the simplest series parallel architecture that can be realized (SPHEV 1, Figure 73). Then, knowing this architecture's weak points, 3 additional SPHEV variants are proposed, they are shown in Figure 74. The four versions of SPHEVs and the reference THS are then optimized and compared. The bi-level optimization methodology presented in Chapter 4 is applied on the architectures developed models.

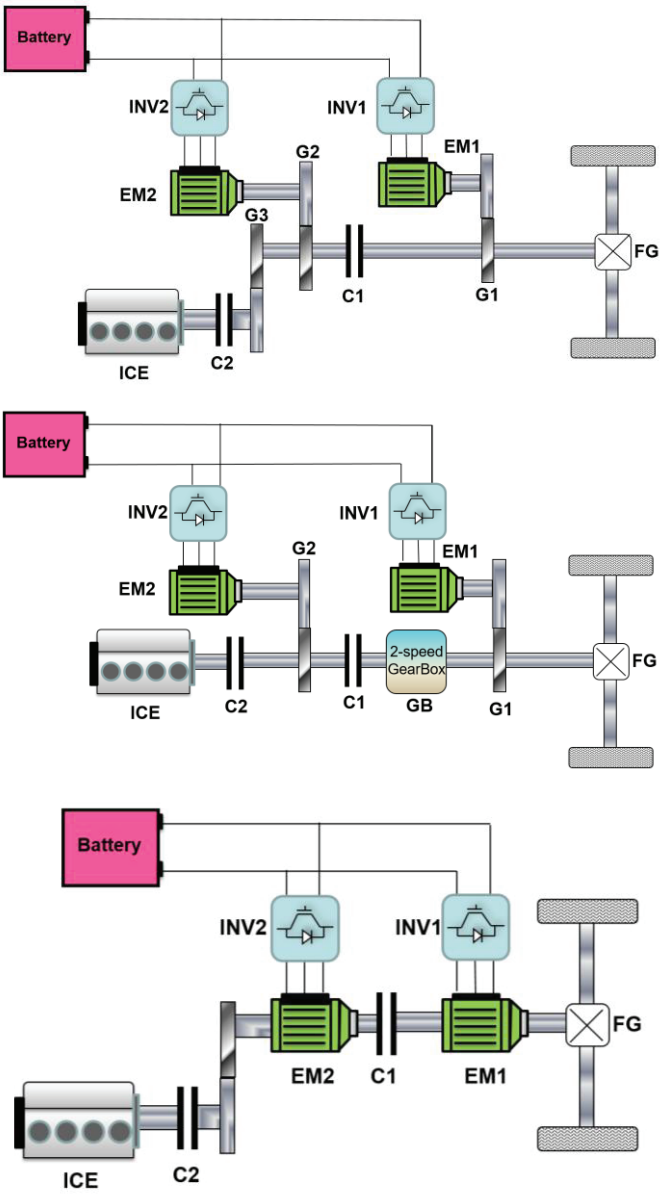


Figure 74: The three proposed SPHEV architectures in [53]

The variant with a gearbox showed good improvements and a potential of fuel saving close to the THS architecture. It was the most interesting architecture and merits more investigations.

For this reason, the searching process will be performed in this manuscript with the components of this architecture. However, this time the searching is done using the entire automatic methodology presented in the previous chapters.

5.3. *STARTING COMPONENTS*

The starting components that are injected in the automatic generation tool are the components of the SPHEV architecture that has a 2-speed gearbox in Figure 74. They are shown in Figure 75 and are listed below:

- 1 ICE
- 1 FD
- 2 MG
- 4 shafts (black dots)
- 2 clutches (red dots)
- 1 synchro unit

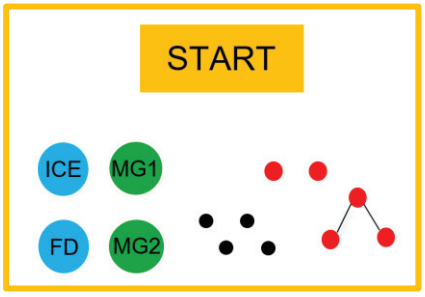


Figure 75: Starting components

The problem constraints are already predefined inside the architecture generation tool. The Constraint Satisfaction Problem solving is launched and the tool generates all the possible solutions that are visualized as graphs.

5.4. *THE AUTOMATICALLY GENERATED ARCHITECTURES*

The tool generated 480 graphs of architectures in 807.1 seconds.

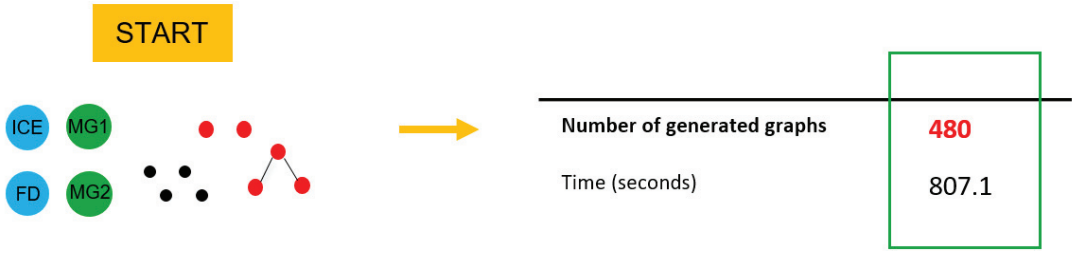


Figure 76: Number of generated graphs

Four examples of these generated graphs are shown below:

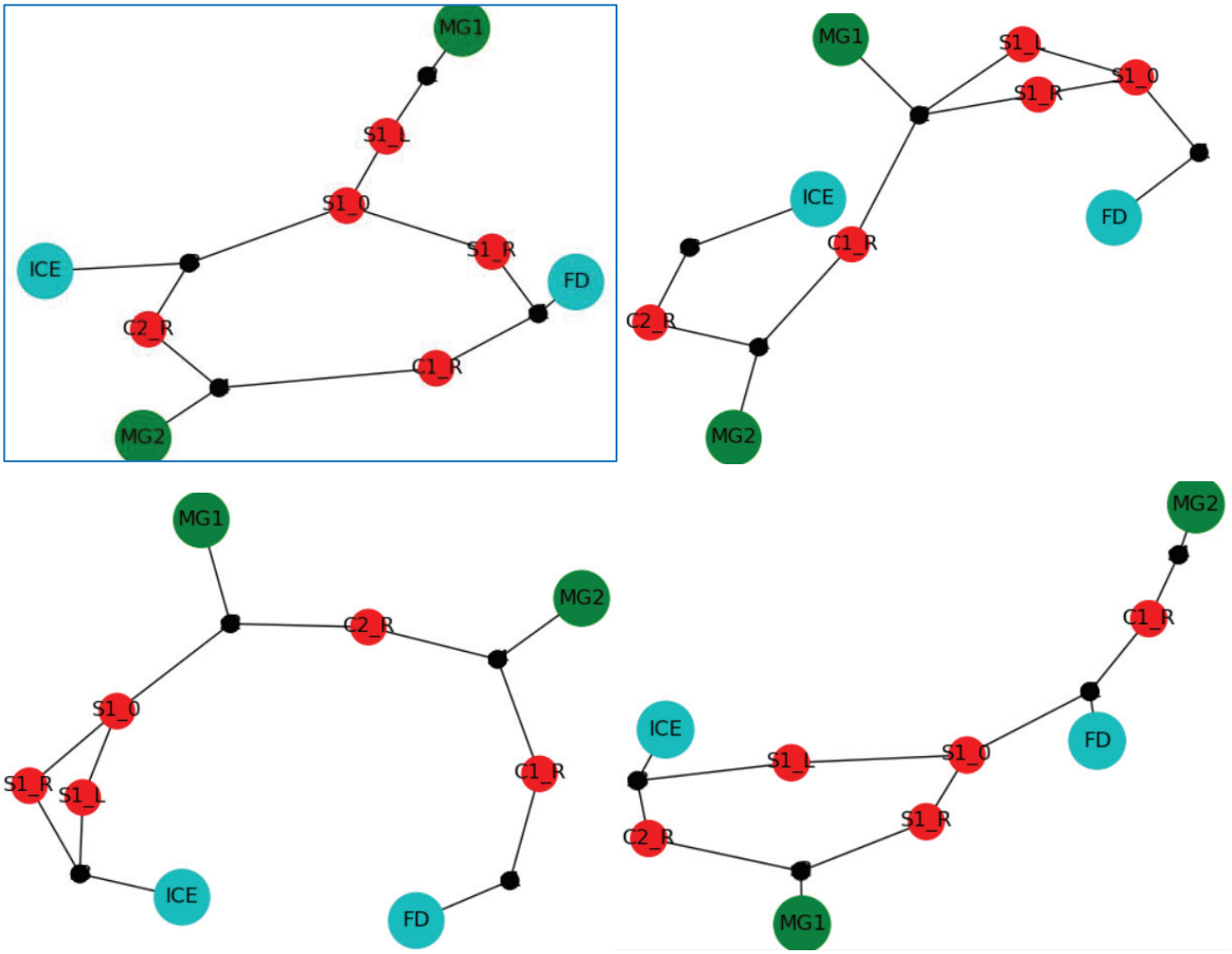


Figure 77: Four examples of generated graphs

Each of these parent graphs has $2^{N_{clutches}} \times 3^{N_{synchros}} = 2^2 \times 3^1 = 12$ state graphs created by the combinations of states of the clutches and synchro unit. These state graphs are also automatically generated and the corresponding modes are detected as seen in Chapter 4. The global efficiencies and ratios of the flows between components are also automatically issued.

This allows the creation of a Modes Table and Modes Table + for each of the parent graph. The graph with bleu frame in Figure 77 is reshown in Figure 78 along with its 0ABC Table (Figure 79), its Modes Table (Figure 80) and its Modes Table + (Figure 81).

In Figure 79, 1st line of the 0ABC Table indicates that the ICE is connected to FD through a synchro. The 2nd line: MG1 cannot be connected to FD. The 3rd: MG2 connected to FD through

a clutch. The 4th: ICE connected to MG1 through a synchro. The 5th : ICE connected to MG1 through a clutch, and the 6th: MG1 and MG2 are not connected (1 stage connection).

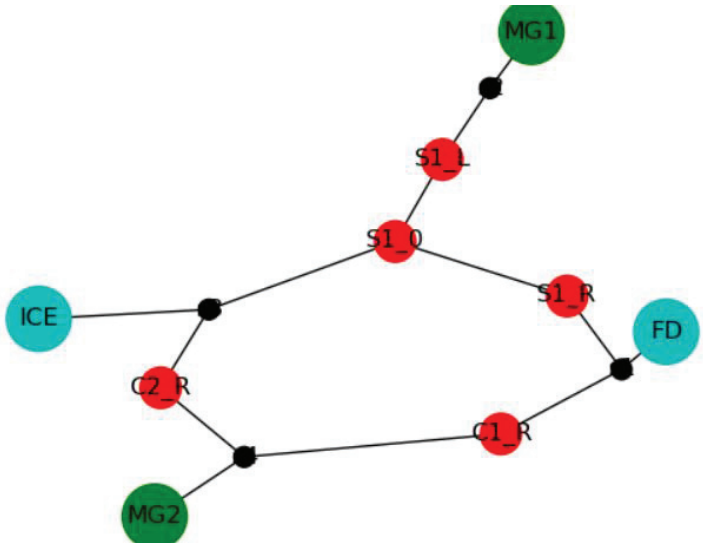


Figure 78: An example graph

	0	A	B	C
ICE<->FD	0	0	1	0
MGs<->FD	0	0	0	0
	0	0	0	1
ICE<->MGs	0	0	1	0
	0	0	0	1
MGs<->MGs	0	0	0	0

0 : no connection
A: direct connection on same shaft
B: connection through synchro
C: connection through clutch

Figure 79: The corresponding 0ABC Table for the graph in Figure 78

Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	1
Unfeasible	1
Unknown	0
Total	12

Figure 80: The corresponding Modes Table for the graph in Figure 78

Modes	Information
Parallel 2MG	[k_FD*k_ICE, '0.96', k_FD*k_MG1/k_S1_L, '0.94', k_FD*k_MG2, '0.96', k_ICE*k_S1_L/k_MG1, '0.96', k_ICE/k_MG2, '0.98']
Unfeasible	
Parallel 1MG	[2, k_FD*k_ICE, '0.96', k_FD*k_MG2, '0.96', k_ICE/k_MG2, '0.98']
Stand-still charging 2MG	[k_ICE*k_S1_L/k_MG1, '0.96', k_ICE/k_MG2, '0.98']
Parallel 1MG	[2, k_FD*k_ICE*k_S1_R, '0.94', k_FD*k_MG2*k_S1_R, '0.94', k_ICE/k_MG2, '0.98']
Stand-still charging 1MG	[2, k_ICE/k_MG2, '0.98']
Series	[2, k_FD*k_MG2, '0.96', k_ICE*k_S1_L/k_MG1, '0.96']
Parallel 1MG	[2, k_FD*k_ICE*k_S1_R, '0.94', k_FD*k_MG2, '0.96', k_ICE*k_S1_R/k_MG2, '0.96']
Pure 1MG	[2, k_FD*k_MG2, '0.96']
Stand-still charging 1MG	[1, k_ICE*k_S1_L/k_MG1, '0.96']
ICE only	[k_FD*k_ICE*k_S1_R, '0.94']
Nutral	

Figure 81: The corresponding Modes Table + for the graph in Figure 78

The 12 modes of the architecture are listed in the Modes Table (Figure 80). In the Modes Table +, the description of the power flows in each mode is found. For example, the 3rd line states that the mode is 'Parallel 1MG'. On the right is found the information of this mode:

['2, k_FD*K_ICE, '0.96', k_FD*k_MG2, '0.96', k_ICE/K_MG2, '0.98']

- 2: meaning that the MG2 is connected to the wheels in this parallel mode
- k_FD*K_ICE: the global ratio of the flow between the ICE and FD
- '0.96': the global efficiency of the flow between the ICE and FD
- k_FD*k_MG2: the global ratio of the flow between the MG2 and FD
- '0.96': the global efficiency of the flow between the MG2 and FD
- k_ICE/K_MG2: the global ratio of the flow between the ICE and MG2
- '0.98': the global efficiency of the flow between the ICE and MG2

In a same way, the other modes are also described with their information.

All these tables and information are automatically generated for all the 480 graphs and can be visualized or not depending on the user preference. They will be filtered in the next section.

5.5. *THE AUTOMATIC FILTERING*

A filtering is required for those 480 graphs. Three steps of filtering are done in this example and are explained hereafter.

5.5.1. Redundancy filtering

Some of these graphs are redundant and should be removed. Ideally, these redundancies could have been avoided if appropriate special constraints for them were defined prior to the start of

the generation process. That was not the case. Expressing these special constraints was complicated and the authors preferred to generate these redundant graphs and filter them later. This filtering is done now at this stage.

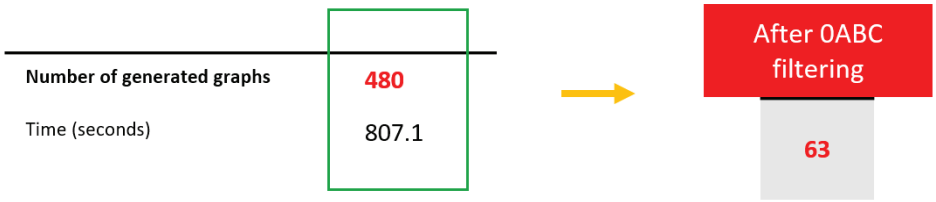
The conversion of the graphs into 0ABC Tables allows the detection of these redundancies. A first group of the redundant graphs are different graphs having same 0ABC Tables. The difference between the graphs is only in:

- the direction of the synchro unit between 2 nodes
- the index of the clutch
- the index of the synchro unit

The first step in removing the redundant graphs is to compare the 0ABC Table of the generated graph to the 0ABC Tables of the previously generated graphs. If the new generated graph has same 0ABC Table than any of the previous graphs, the new generated graph is dismissed from the process. This was explained in Chapter 3.

Another redundancy type exists. In this type, the redundant graphs are different graphs having different 0ABC Tables. The difference between the graphs is only that EM1 in one is EM2 in the other and vice versa. The difference between their 0ABC Tables is that the lines of EM1 in one corresponds to the lines of EM2 in the other and vice versa. For this reason, the 0ABC Table of any generated graph is compared to not only the 0ABC Tables of the previously generated graphs, but also the 0ABC Tables with EM1 and EM2 lines switched.

After this automatic filtering, the number of architectures is reduced to 63.



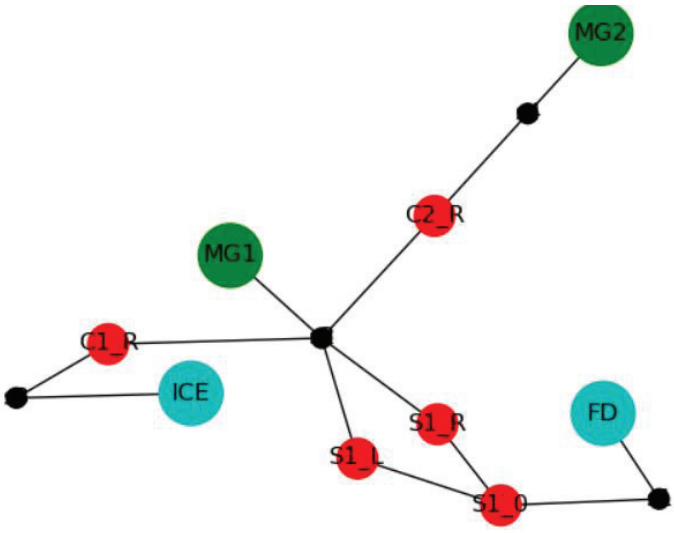
5.5.2. Modes filtering

Each of these 63 architectures has a Mode Table. They might be missing a mode that the design engineer estimates vital. For example, the architectures might be required to have at least

one pure electric driving mode, or at least one pure ICE mode. These constraints might differ from a vehicle design project to other and from OEM to other. In the example considered in this chapter, the authors consider the following modes constraints. Each architecture should have at least:

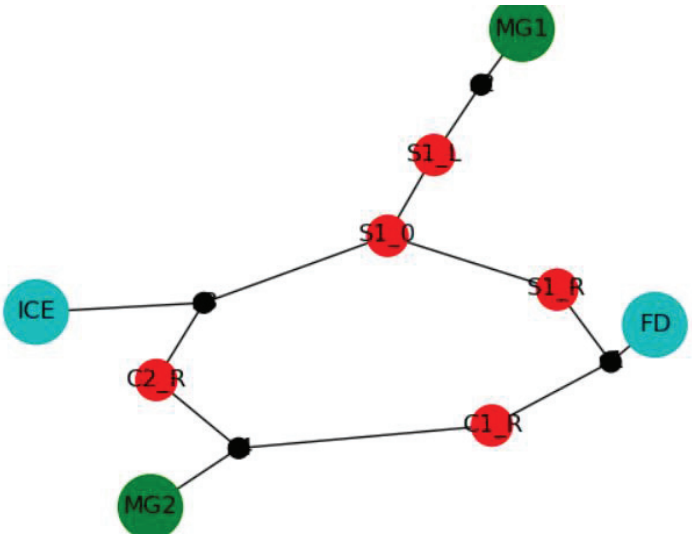
- 1 ICE only mode
- 1 (Pure 1MG or Pure 2MG) mode
- 1 Series mode
- 1 (Parallel 1MG or Parallel 2MG) mode
- 1 (Series or Stand-still charging 1MG or Stand-still charging 2MG) mode

Out of these 63 architectures, only 17 pass this modes filtering step and 46 do not pass and are therefore dismissed. An example of these dismissed architectures is shown in Figure 82. It lacks the ICE mode and the Series mode. An example of architecture that passes the modes filtering is shown in Figure 83. It has all the required modes.



Modes	Nb
ICE only	0
Pure 1MG	2
Pure 2MG	2
Parallel 1MG	2
Parallel 2MG	2
Series	0
Nutral	2
Stand-still charging 1MG	1
Stand-still charging 2MG	1
Unfeasible	0
Unknown	0
Total	12

Figure 82: Example of a dismissed architecture



Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	1
Unfeasible	1
Unknown	0
Total	12

Figure 83: Example of architecture that passes the modes filtering

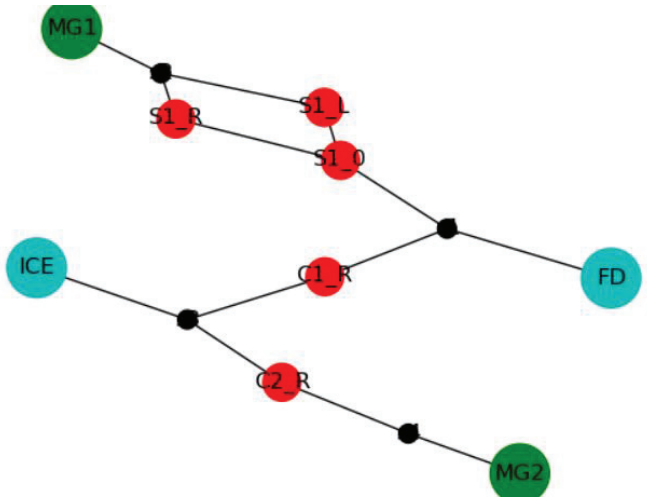
5.5.3. Filtering based on the number of paths from a component to the wheels

These 17 architectures have the same gearbox components: two clutches and one synchro unit. However, the allocation of these gearbox components to which powertrain components is different from an architecture to other. This is why the architectures have a same total number of modes (=12) but they do not have the same modes or the same number of mode instances. Moreover, this is why the number of paths between two chosen components might be different from an architecture to other.

Design engineer might favor to have multiple paths between two specific components. This is the concept of this filtering step. In fact, electric vehicle powertrains require only one path from the electric machine to the wheels. This is why usually electric vehicles does not have a gearbox. Only one gear ratio or one path between the electric machine and the wheels can be enough. However, this cannot be applied in conventional ICE powertrains that require a gearbox with multiple gear ratios between the ICE and the wheels. Fitting the wheel speed and torque demands in the good efficiency zone of the engine speed and torque map requires a freedom in gear choices.

6 out of the 17 architectures happen to have only one path between the engine and the wheels. One of these 6 architectures is shown in Figure 84. The synchro unit is placed between MG1 and FD, which offers the path MG1<—>FD 2 gear ratios and leave the path ICE—>FD

with only 1 gear ratio. This means that the ICE has only one gear choice in pure ICE and parallel hybrid modes.



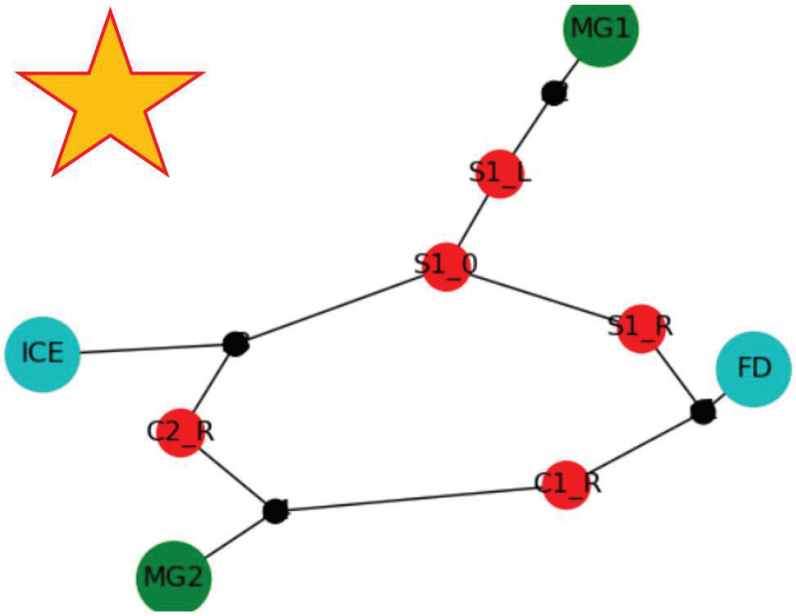
Modes	Nb
ICE only	1
Pure 1MG	2
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	2
Series	2
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	0
Unknown	0
Total	12

Figure 84: Example of architectures with 1 path from the ICE to the wheels

These 6 architectures are considered to have low potential for being the best in energy efficiency. They will be dismissed from the process and 11 architectures are left. They are the most promising architectures and are listed in the coming section.

5.6. MOST PROMISING ARCHITECTURES

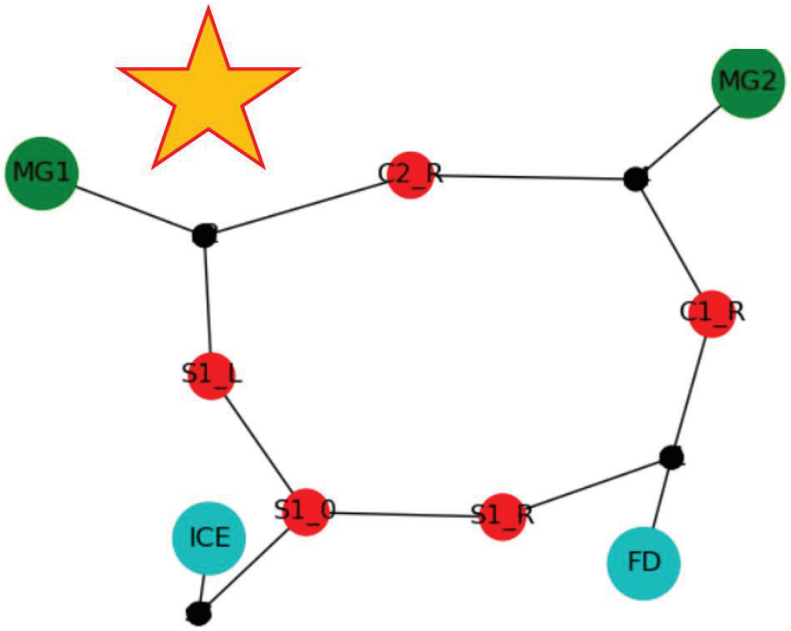
The most promising architectures are the 11 architectures that passed all the filtering steps. They are shown along their Mode Table in Figure 85 to Figure 95.



1

Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	1
Unfeasible	1
Unknown	0
Total	12

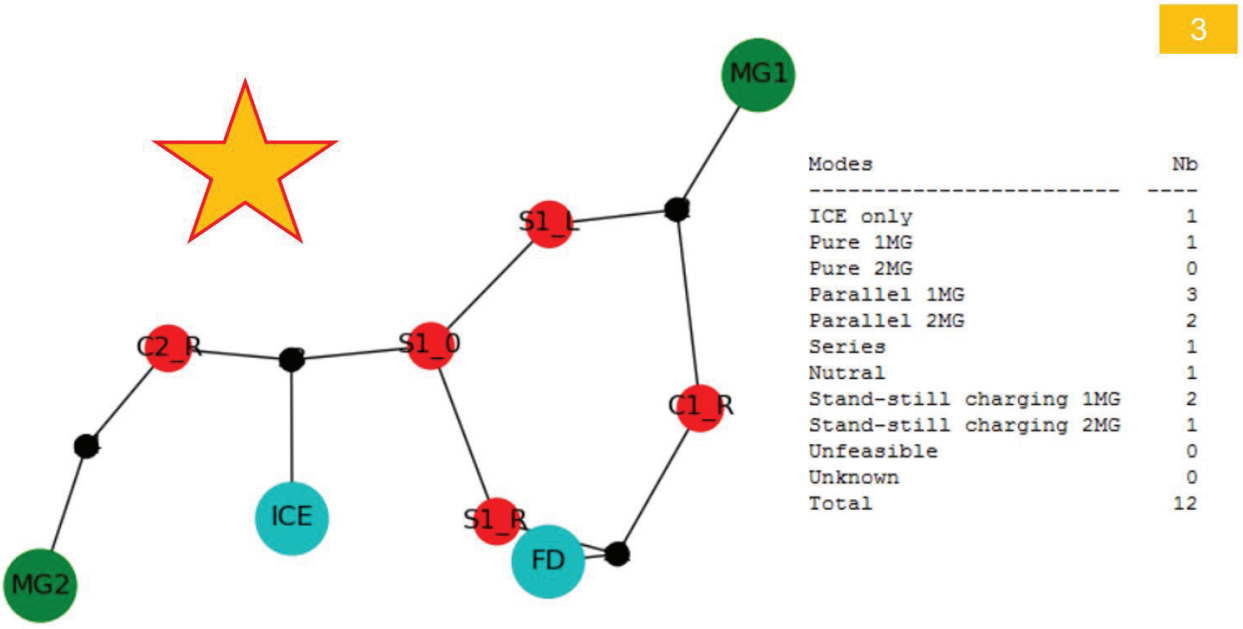
Figure 85: Architecture 1



2

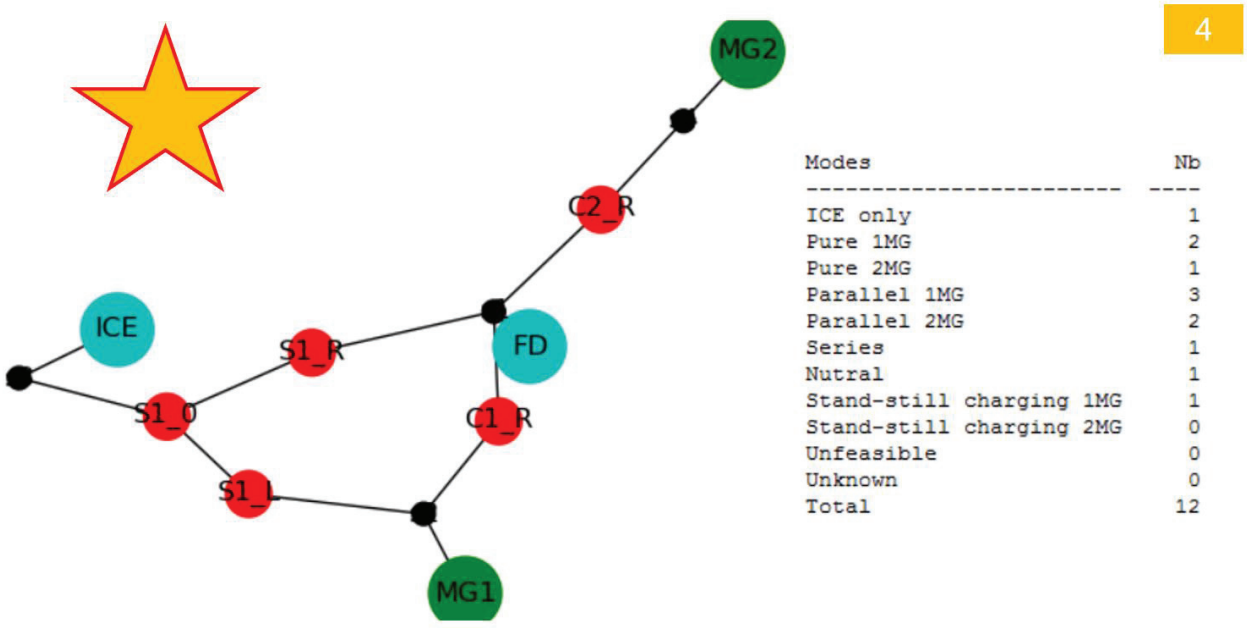
Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	1
Parallel 1MG	1
Parallel 2MG	2
Series	1
Nutral	2
Stand-still charging 1MG	1
Stand-still charging 2MG	1
Unfeasible	0
Unknown	1
Total	12

Figure 86: Architecture 2



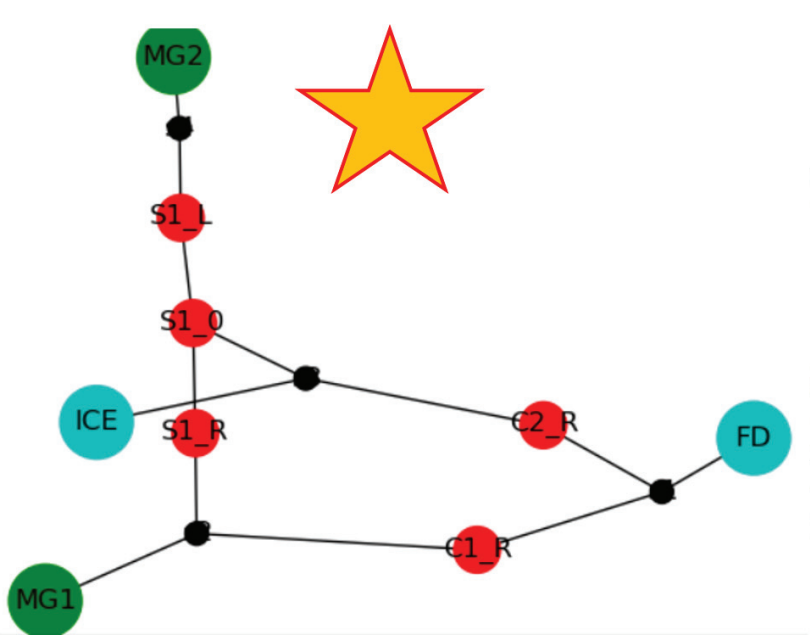
Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	2
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	1
Unfeasible	0
Unknown	0
Total	12

Figure 87: Architecture 3



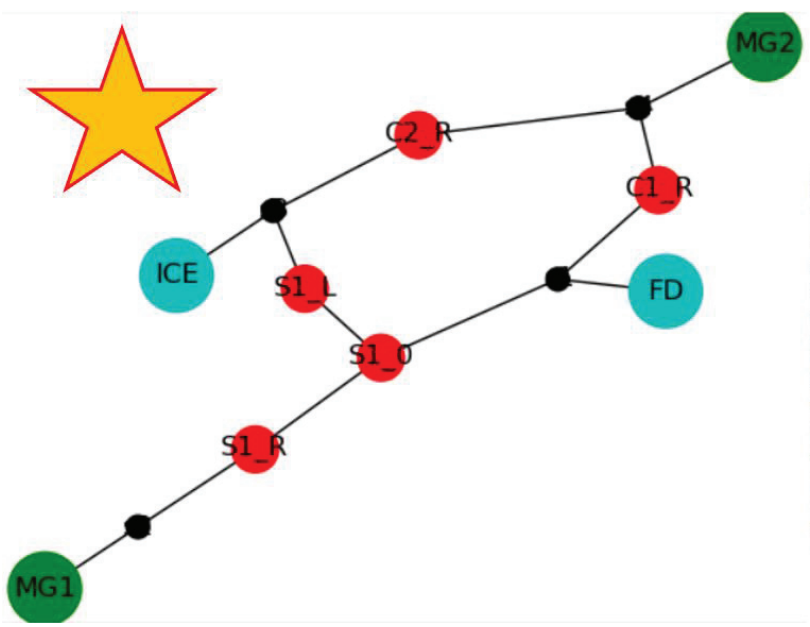
Modes	Nb
ICE only	1
Pure 1MG	2
Pure 2MG	1
Parallel 1MG	3
Parallel 2MG	2
Series	1
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	0
Unknown	0
Total	12

Figure 88: Architecture 4



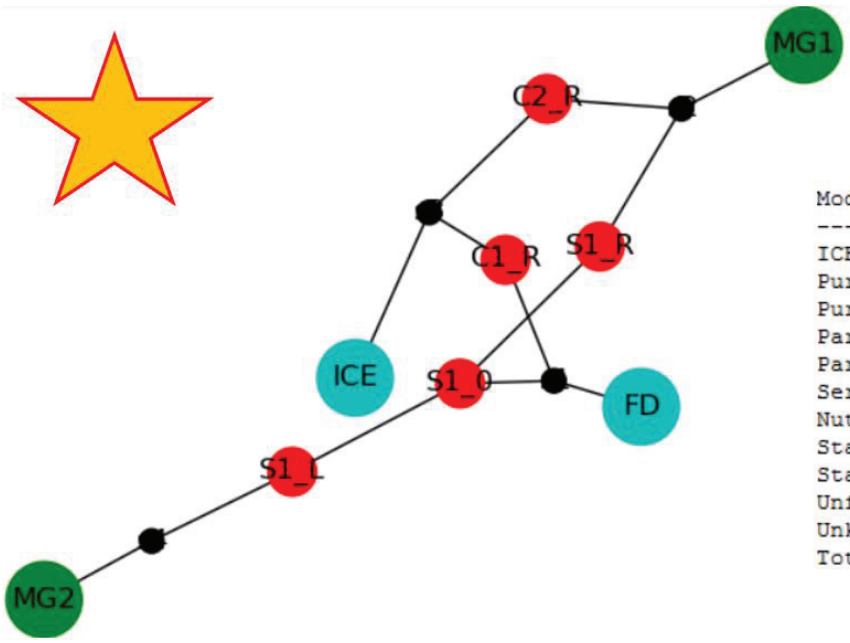
Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	4
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	0
Unfeasible	1
Unknown	0
Total	12

Figure 89: Architecture 5



Modes	Nb
ICE only	1
Pure 1MG	2
Pure 2MG	1
Parallel 1MG	3
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	1
Unknown	0
Total	12

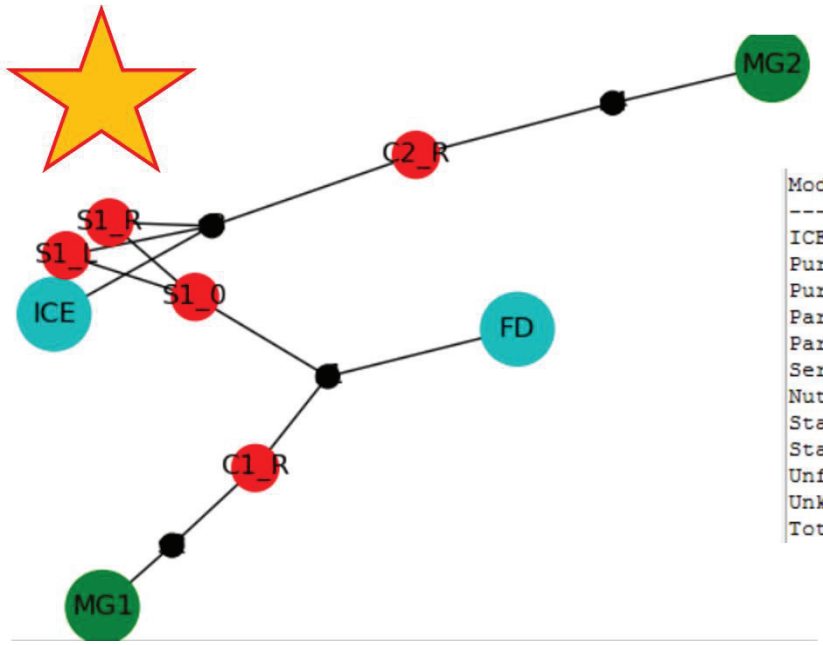
Figure 90: Architecture 6



7

Modes	Nb
ICE only	1
Pure 1MG	2
Pure 2MG	0
Parallel 1MG	4
Parallel 2MG	1
Series	1
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	1
Unknown	0
Total	12

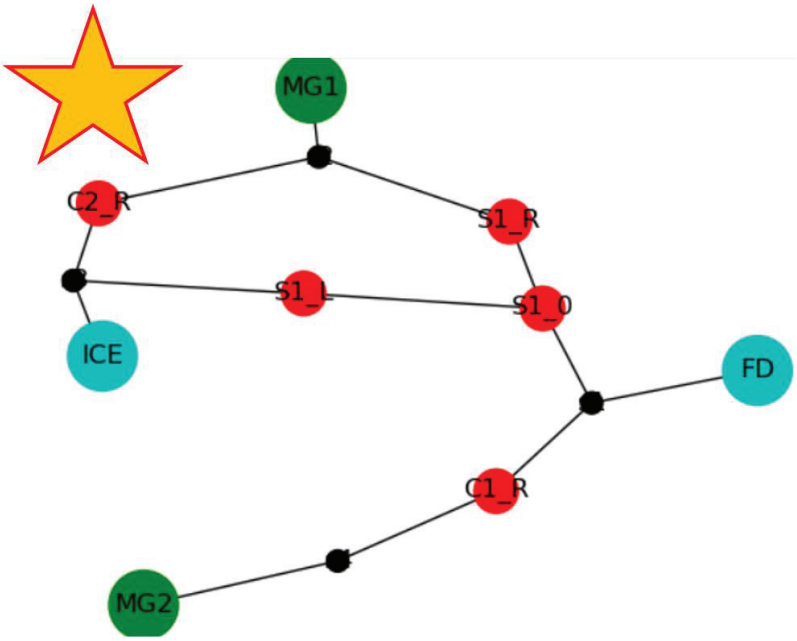
Figure 91: Architecture 7



8

Modes	Nb
ICE only	2
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	4
Parallel 2MG	2
Series	1
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	0
Unknown	0
Total	12

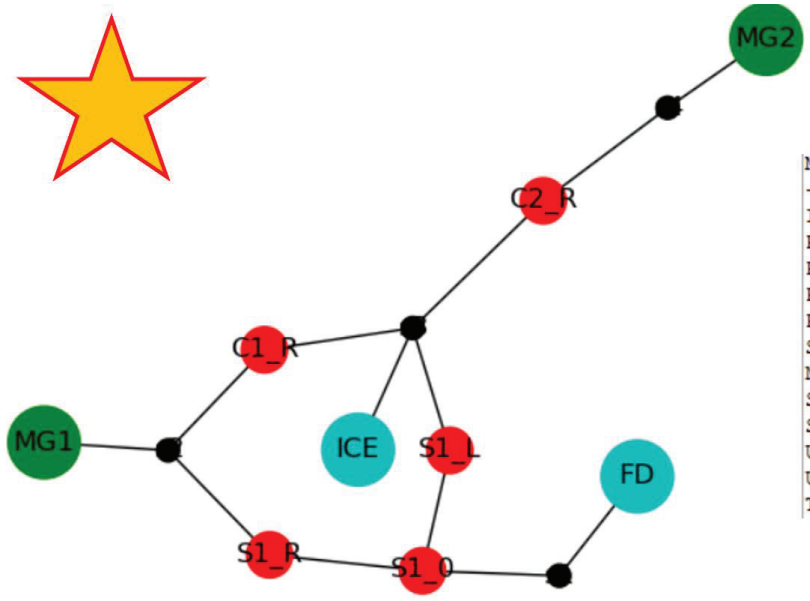
Figure 92: Architecture 8



9

Modes	Nb
ICE only	1
Pure 1MG	2
Pure 2MG	1
Parallel 1MG	3
Parallel 2MG	2
Series	1
Nutral	1
Stand-still charging 1MG	1
Stand-still charging 2MG	0
Unfeasible	0
Unknown	0
Total	12

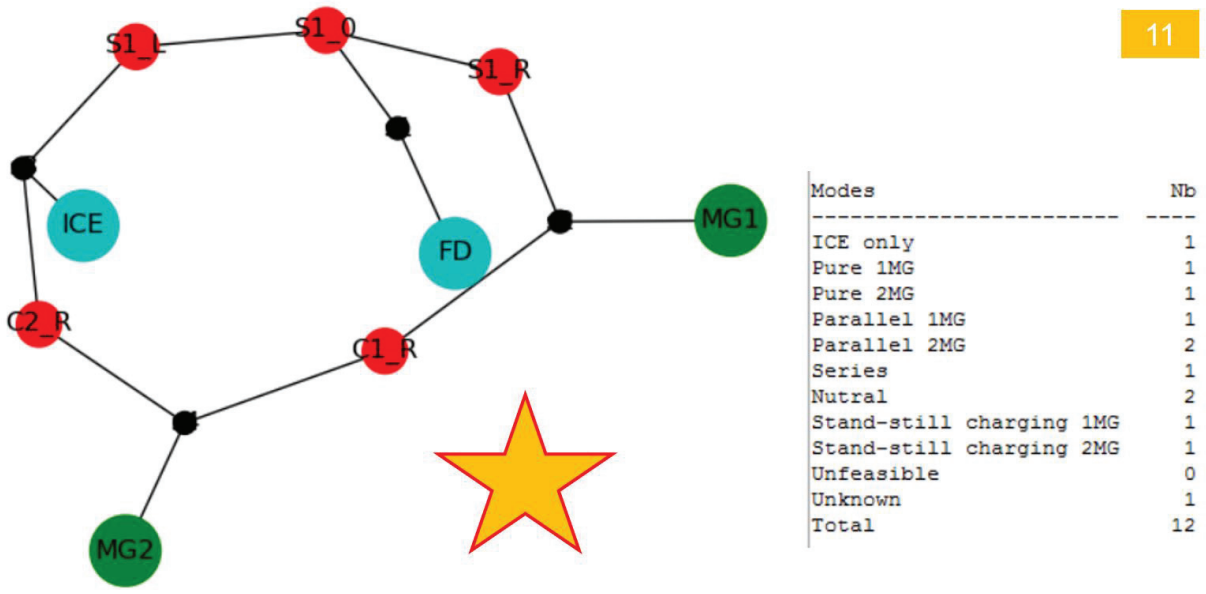
Figure 93: Architecture 9



10

Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	0
Parallel 1MG	3
Parallel 2MG	2
Series	1
Nutral	1
Stand-still charging 1MG	2
Stand-still charging 2MG	1
Unfeasible	0
Unknown	0
Total	12

Figure 94: Architecture 10



Modes	Nb
ICE only	1
Pure 1MG	1
Pure 2MG	1
Parallel 1MG	1
Parallel 2MG	2
Series	1
Nutral	2
Stand-still charging 1MG	1
Stand-still charging 2MG	1
Unfeasible	0
Unknown	1
Total	12

Figure 95: Architecture 11

5.7. THE AUTOMATIC OPTIMIZATION

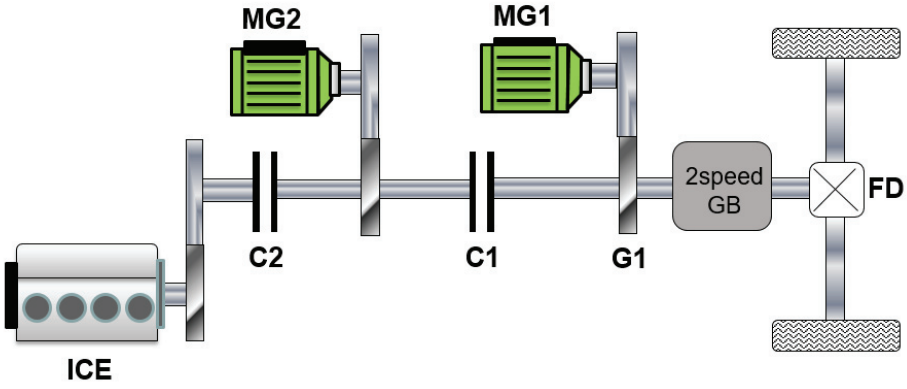
These 11 selected architectures need to be assessed, optimized and compared. Their Modes Tables + are passed to the general hybrid model that was explained in the previous chapter.

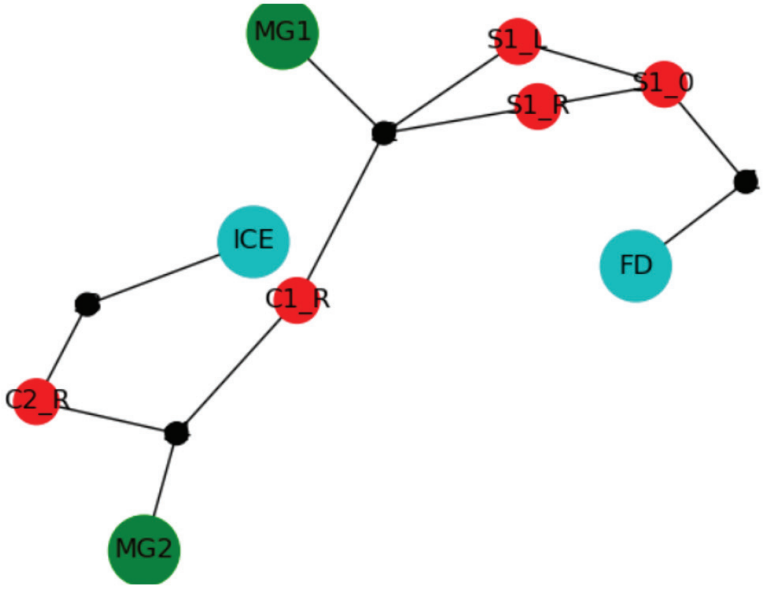
5.7.1. Two added architectures

Here we consider that the proposed methodology is not present. If we want to imagine the architectures that can be realized with the chosen components (1 ICE, 1 FD, 2 MGs, 4 shafts, 2 clutches, and 1 synchro unit of 2 gears), the first two architectures that come to mind are shown in Figure 96 and Figure 97. They will be called architecture 12 and 13. The synchro unit is positioned as a conventional gearbox between 1 input shaft and 1 output shaft, between the MG and the wheels or between the 2 MGs.

These architectures are two of the 63 generated graphs. However, they were dismissed in the Modes filtering step because they lack an 'ICE only' mode. Still, it is interesting to consider them because they were proposed to be studied by the authors as improved versions of the simple

series-parallel architecture in [53], before the realization of the proposed methodology. Architecture 13 is functionally the same as an architecture proposed by Denso Corporation in [98] and studied in [53].





12

Modes	Nb
ICE only	0
Pure 1MG	2
Pure 2MG	2
Parallel 1MG	0
Parallel 2MG	2
Series	2
Nutral	2
Stand-still charging 1MG	1
Stand-still charging 2MG	1
Unfeasible	0
Unknown	0
Total	12

Figure 96: Architecture 12

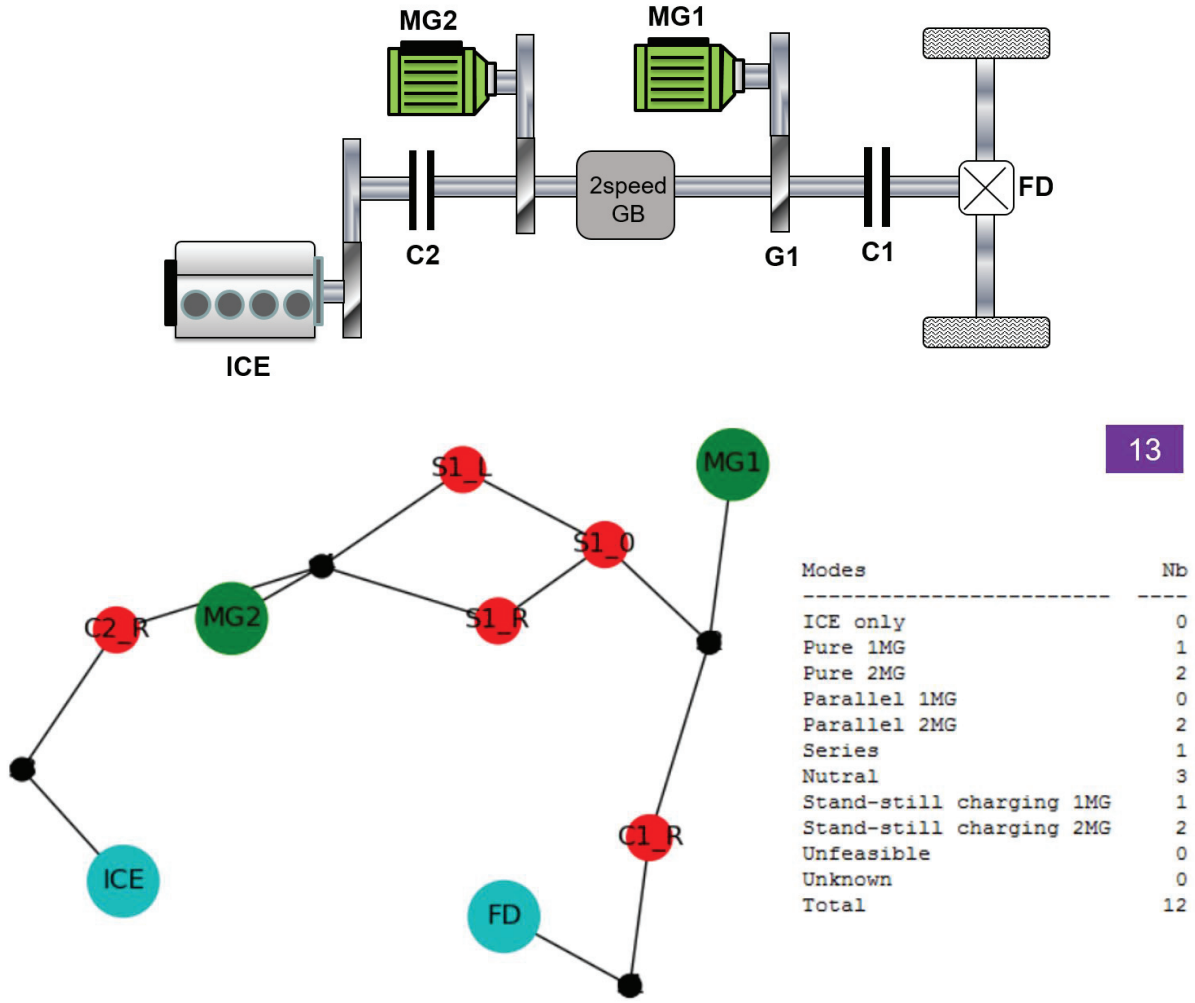


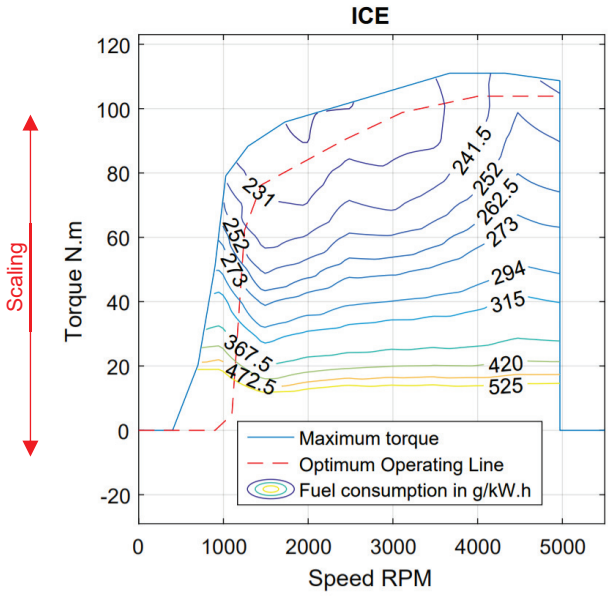
Figure 97: Architecture 13

5.7.2. Choice of components

For the powertrain assessment and optimization, the vehicle characteristics and components used correspond to a middle class HEV and can be found in Table 6. The reference maps of EM1, EM2 and ICE before sizing are shown in Figure 98. The sized maps will be generated by multiplying the torque and losses by the scaling factor.

Table 6: Components specification

Component	Specification
Engine	Gasoline, Atkinson
EM1	Permanent Magnet Synchronous Motor
EM2	Permanent Magnet Synchronous Motor
Battery	Nickel-metal hydride, (each module: 6.5 Ah capacity, 1kW maximum power)
Gears	Parallel axis helical gears with a 98% efficiency if their ratio is different than 1 and 100% efficiency if their ratio is 1
Final Drive	97% efficiency
Vehicle	Middle class HEV



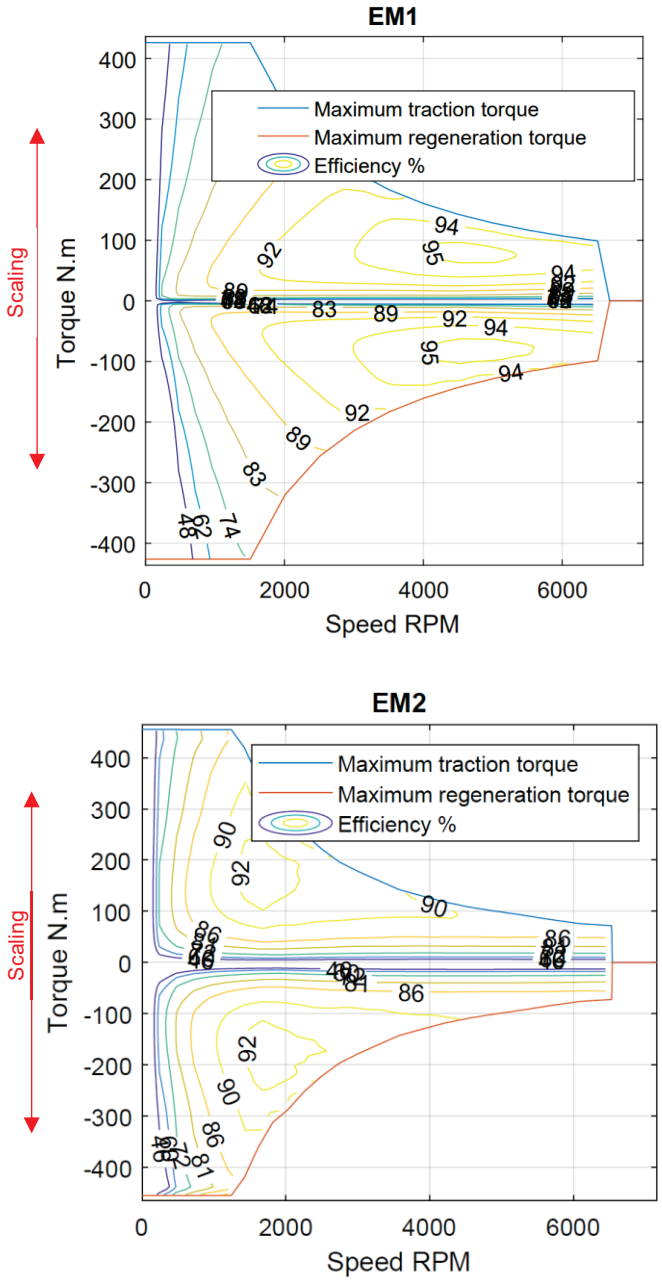


Figure 98: EM1, EM2 and ICE maps before sizing

The gears efficiency:

The components gear ratios are variables in the sizing optimization process. The optimization can choose to remove a gear by setting its value to 1. For this reason, the gear is given an efficiency of 1 in case its ratio is equal to 1 (no gear) and a constant value not equal to 1 in case its ratio is different than 1 (Figure 99).

In that way, if the gear presence is not so important for the powertrain, the optimization will tend to remove it in order to gain this increase in efficiency.

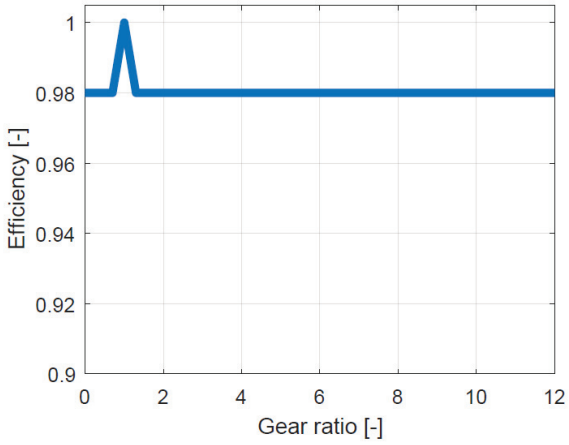


Figure 99: Components gears efficiency

The fuel consumption objective function (FCons) is evaluated in mixed driving condition. FCons is computed as a weighted average of the fuel consumption in charge sustaining mode in urban, rural road and highway conditions. The ARTEMIS European driving cycles [83] are used to simulate these three conditions and the control is optimized on each of them.

$$FCons = \alpha.FCons_{urban} + \beta.FCons_{rural} + \gamma.FCons_{highway} \quad (10)$$

Where the α, β and γ are coefficients calculated from the mean traveled distance by the French population in urban, rural road and highway conditions. These values are respectively 0.4, 0.3 and 0.3 [99].

5.7.3. Performance values

In the performance check, the following values are considered:

- Acceleration time from 0 to 100 km/h: $t0{-}{>}100 < 10.1$ s
- Acceleration time from 80 to 120 km/h: $t80{-}{>}120 < 7.5$ s
- Maximum speed of the vehicle on a flat road: $Vmax > 180km/h$

5.7.4. DP parameters

For DP solving the following parameters are chosen:

Table 7: DP chosen parameters

Initial SOC	60%
Final SOC	60%
SOC minimum	20%
SOC maximum	70%
Time step	1s
SOC discretization step	0.05%
Torque discretization step	5 N.m.
ICE speed discretization step in series mode	50 rad/s

5.7.5. NSGA parameters

NSGA is defined and parametrized as follows:

- Number of design variables: 10
  - Number of battery modules
  - Power of ICE
  - Power of EM1
  - Power of EM2
  - Gear ratio on ICE
  - Gear ratio on EM1
  - Gear ratio on EM2
  - FD gear ratio
  - Synchro unit gear 1
  - Synchro unit gear 2

- Number of objectives: 2
  - Charge sustaining fuel consumption:

$$FCons = \alpha.FCons_{urban} + \beta.FCons_{rural} + \gamma.FCons_{highway}$$

  - Number of battery modules in series

- Parametrization:

Table 8: NSGA chosen parameters

Population size	56
Number of generations	400
Tournament size	2
Size of the mating pool	28
Mutation probability	0.2
Crossover probability	0.8

5.8. RESULTS: COMPARISON IN TERMS OF FUEL CONSUMPTION AND BATTERY SIZE

The optimization methodology explained in Chapter 4 is applied on the general hybrid model that is called using the values of Modes Tables + of architecture 1. This will result in a Pareto front for architecture 1 showing the optimal tradeoff between the number of battery modules and the fuel consumption.

The Modes Table + is changed to the one of architecture 2, and the process is repeated. The Pareto front of architecture 2 is then obtained. The process is repeated for all the architectures until architecture 13. The Pareto fronts are presented in Figure 100.

A recall is made here to remember what these different architectures are. In an interest in exploring series-parallel architectures, the authors injected inside the proposed tool the components of a relatively simple series-parallel architecture. These components are 1 ICE, 1 FD, 2 MGs, 4 shafts, 2 clutches, and 1 synchro unit of 2 gears. The methodology generated 480 graphs of architectures, 63 non-redundant.

Architectures 12 and 13 were proposed to be studied by the authors as improved versions of the simple series-parallel architecture in [53], before the realization of the proposed methodology. One of these architectures has the same functionality of an architecture proposed by Denso Corporation in [98]. These 2 architectures are two of the 63 generated graphs.

Optimizing and simulating all the 63 architectures is time consuming. Therefore, the methodology filtered these 63 architectures. It selected 11 architectures that are believed to be the most promising or that at least include between them the most promising architecture. These are Architectures 1 until 11. Architectures 12 and 13 were not between those selected

architectures because they lack an 'ICE only' mode. They will be used in the comparison as reference architectures that would have been chosen if the proposed methodology is unavailable.

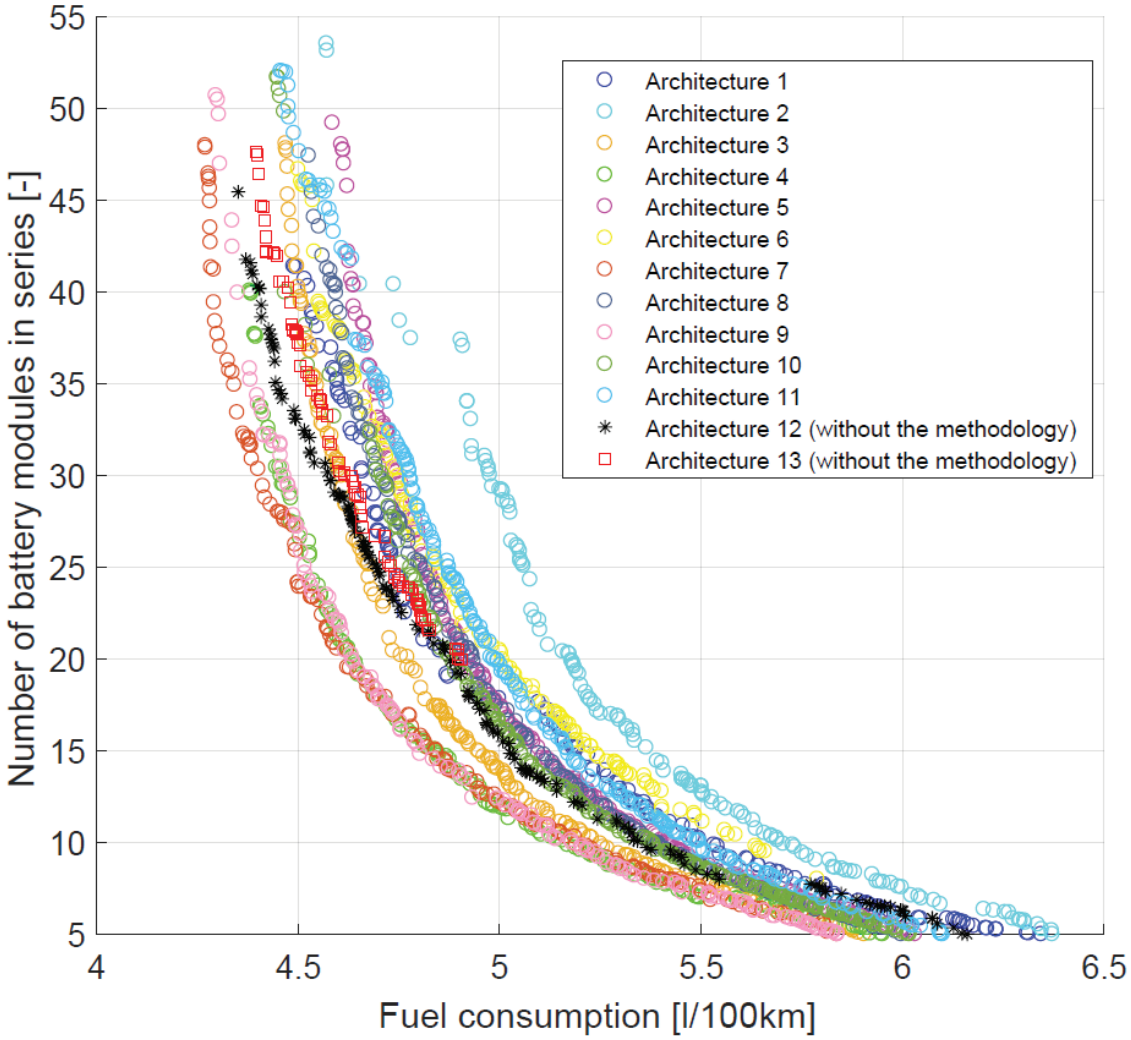


Figure 100: Pareto fronts of the 13 considered architectures

The resulted Pareto fronts show that between the 11 selected architectures, some are better than architectures 12 and 13 in terms of the 2 considered objective functions. In fact, for a same number of battery modules, architectures 4, 7 and 9 have lower charge sustaining fuel consumption than architectures 12 and 13. This is also true for architecture 3 for a number of battery modules less than 26. The other architectures are worse than architectures 12 and 13.

The best architecture in terms of the 2 considered objective functions is architecture 7.

Those results prove that the methodology succeeded in:

1- Automatically generating all the graphs of possible architectures, **including the 2 conventional architectures** proposed without the methodology.
2- Automatically filtering these architectures and selecting some of them for optimization and assessment. These selected architectures seem to be quite efficient, most of them have fuel consumptions close to the ones of architecture 12 and 13 (maximum 6% higher FC, except for architecture 2)
3- Finding an architecture better than the architectures that were proposed in the absence of the methodology (architecture 7 having a fuel consumption about 5% lower than architectures 12 and 13 for same number of batteries).

To evaluate the correctness of the filtering steps, some filtered architectures were simulated and compared with the selected architectures.

Assessment of the correctness of the filtering based on the number of paths from a component to the wheels:

6 architectures were dismissed in this filtering step because they have only one path (one gear choice) between the engine and the wheels. These architectures are optimized and compared to the 4 best architectures (3, 4, 7 and 9) and the engineer chosen architectures. They are found to be worse than the considered architectures. The Pareto of these 6 filtered architectures are shown in Figure 101.

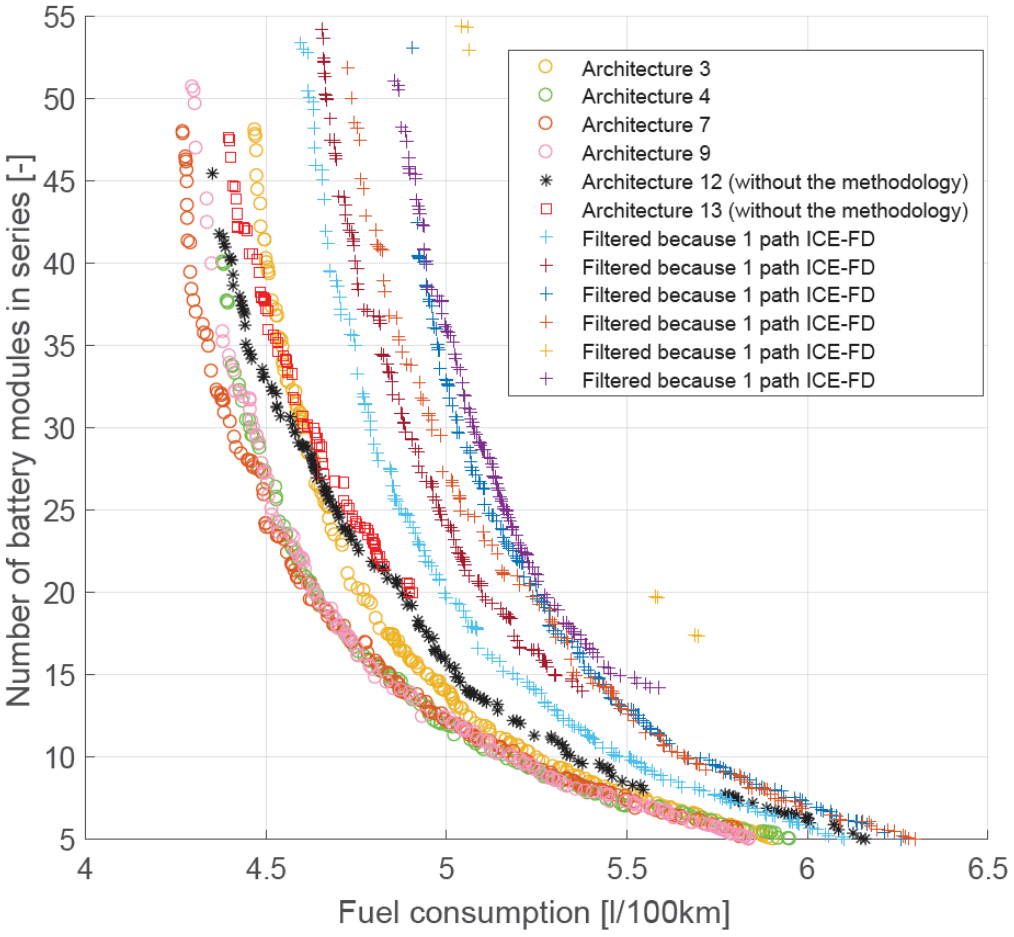


Figure 101: Filtered architectures because of 1 path ICE - FD

This confirms that this filtering step succeeding in removing architectures that will not be the best. This filtering step is to be kept.

Conclusion:

A case study was performed in this chapter on the proposed methodology. The first results were exposed in the aim of showing the capability of the entire methodology from its start till its end. Surely, more detailed analysis need to be done on the found results: analysis of the optimal sizing, analysis of the powertrain operation and its control, the importance of the gears or clutches and the effect of eliminating them… Due to time limitation, these questions were not answered now. They will be answered in the future works.

CHAPTER 6: CONCLUSION

Abstract

Abstract - In this chapter, a final global look is made on the presented work. Then, the main contributions are highlighted and the limitations are presented. Finally, projections are made on the future work.

6.1. FINAL GLOBAL LOOK

The thesis objective was to present a systematic methodology for the generation and design of hybrid vehicle powertrains.

To do this, the manuscript started by presenting in chapter 1 the context that is leading to vehicle hybridization and electrification. It also introduced the hybrid electric powertrains and their design problem.

This design problem was detailed in chapter 2 and was exposed in its optimization context. The state of the art was overviewed and the main methods used for this optimization were shown. This chapter also introduced the authors' proposed methodology to solve the design problem.

The detailed explanation of the proposed methodology was done in chapters 3 and 4. Chapter 3 was dedicated to the part of automatic generation of architectures. The graphical representation used to visualize the architectures was shown, the constraint satisfaction problem was described, and the problem solving and the graphs generation were presented. The techniques used to do automatic filtering for these graphs were also explained. The outcomes of this part of the methodology are the Modes Table + of the architectures. These tables include the information needed for the energetic assessment of the architectures. The automatic assessment and optimization of the generated architectures was the subject of chapter 4. Here it was seen that energetic models are needed and cannot be manually derived for all the generated architectures. To solve this, a general model for (P)HEV was developed and explained. It is inserted inside a bi-level optimization methodology relying on a Genetic Algorithm for the design and sizing optimization and on Dynamic Programming for the control optimization.

The capability of the entire methodology was verified in chapter 5, which was an application on a case study. The components that can lead to simple series-parallel architectures were considered and injected inside the tool. All the feasible architectures were generated and filtered. Eleven most promising architectures were selected. Two dismissed architectures were also selected: they are the first two architectures that might come to the mind of a design engineer if asked to imagine an architecture with the chosen components. The thirteen architectures were automatically optimized using the developed general model and were compared based on fuel

consumption and battery size. Results showed that the tool was capable of automating the design procedure. It also generated non-conventional architectures better than the ones that the design engineer might have thought of.

6.2. CONTRIBUTIONS

This work resulted in two main contributions:

1- The development of a tool that can automatically generate architectures for hybrid powertrains and filter them based on the manufacturer constraints and requirements. This tool was entirely done during this PhD.
2- The development of a general hybrid model that can simulate all the generated architectures, using a novel perspective in modelling.
   Parts of the general model were developed in this PhD: the performance models, the models used to calculate the limits of the DP graph, and the models used to calculate the cost of the edges of the DP graph. In these models, all the possible modes are evaluated.
   Other parts already existed in IFSTTAR VEHLIB library: the dynamic programming functions, the vehicle model, the components models, and genetic algorithm functions.

6.3. LIMITATIONS

The presented work has some weak points and limitations.

1- The tool components library does not include Planetary Gears. Therefore, the tool explores the Architecture design space while excluding from it the part corresponding to power-split architectures.
2- The bi-level optimization of 1 architecture takes around 8-10h

6.4. FUTURE WORK

The target in this PhD was to create a systematic methodology for generation and design of hybrid vehicle powertrains. The work succeeded in reaching a first complete prototype of this

imagined methodology. However, it still needs improvement and development. This will be done in the future and includes:

1- Reassessing the exactness of the components models and the accuracy of the sizing technique.
2- Including the component choice in the design optimization loop.
3- Decreasing the optimization calculation time: initially by improving the codes (removal of loops, removal of unnecessary calculations...), then by developing techniques for a rapid evaluation of the fuel consumption of the design candidates. Abandoning the DP optimal technique is to be considered.
4- Including Planetary Gears in the library of components in order to generate power-split architectures. This implies an increase in the number of possible modes and a need to develop power-split mode models to be included inside the general hybrid model.
5- Including additional aspects of the powertrain in the assessment (compactness, weight, cost, aging, pollutants emissions...).

LIST OF PUBLICATIONS

The work during those three PhD years led to the publication of articles in international conferences and in a journal.

International conferences:

- **Improvement of a Series-Parallel Hybrid Electric Vehicle Architecture** [89]

Bilal Kabalan, Emmanuel Vinot, Yuan Cheng, Rochdi Trigui, Clément Dumand. Article presented at the IEEE VPPC 2017 conference, December 2017, Belfort, France and published in the proceedings, available on IEEE Xplore digital library:

https://ieeexplore.ieee.org/document/8330943/similar#similar

- **Optimal Design and Sizing of Through-The-Road Hybrid Vehicle Powertrain** [90]

Bilal Kabalan, Yuan Cheng, Emmanuel Vinot, Clément Dumand, Rochdi Trigui, Wissam Bou Nader. Article presented at the SIA POWERTRAIN conference, May 2018, Rouen, France and published in the conference proceedings.

https://hal.archives-ouvertes.fr/hal-01907847

- **Sensitivity Analysis on the Sizing Parameters of a Series-Parallel HEV** [88]

Taha El Hajji, Bilal Kabalan, Yuan Cheng, Emmanuel Vinot, Clément Dumand. Article presented at the IFAC AAC conference, June 2019, Orleans, France and published on ScienceDirect.

https://www.sciencedirect.com/science/article/pii/S2405896319306871

Journal articles:

- **Efficiency Improvement of a Series-Parallel Hybrid Electric Powertrain by Topology Modification** [53]

Bilal Kabalan, Emmanuel Vinot, Yuan Cheng, Rochdi Trigui, Clement Dumand, Taha El Hajji. Article published in the journal IEEE Transactions on Vehicular Technology (5.339 impact factor), available on IEEE Xplore digital library:

https://ieeexplore.ieee.org/document/8894384

- **Methodology for the Automatic Generation and Filtering of Hybrid Powertrain Architectures.**

Bilal Kabalan, Emmanuel Vinot, Yuan Cheng, Rochdi Trigui, Clément Dumand. Article in preparation. To be submitted to the journal IEEE Transactions on Vehicular Technology in March 2020.

ANNEXE1

Implementation of the CSP problem

In this example, 2 variables “a” and “b” exist, each having a domain of [1,2,3], and they should take different values. All the solutions are required.

```
>>> problem = Problem()
>>> problem.addVariables(["a", "b"], [1, 2, 3])
>>> problem.addConstraint(AllDifferentConstraint())
>>> problem.getSolutions()
[{'a': 3, 'b': 2}, {'a': 3, 'b': 1}, {'a': 2, 'b': 3},
 {'a': 2, 'b': 1}, {'a': 1, 'b': 2}, {'a': 1, 'b': 3}]
```

In a similar manner, and based on a code initially done for Sudoku solving, the problem of graph generation was coded. Below are shown the main stages in the developed code.

Components definition:

```
# components
V0 = 1      #ICE
V1 = 1      #MG1
V2 = 1      #MG2
V3 = 1      #FD
# connetors_clutches
V4 = 1      #Clutch1_node1
V5 = 0      #Clutch1_node2 always =0
#----
V6 = 0      #Clutch2_node1
V7 = 0      #Clutch2_node2 always =0
#----
V8 = 0      #Clutch3_node1
V9 = 0      #Clutch3_node2 always =0
# connetors_synchros
V10 = 1     #Synch1_node0
V11 = 1     #Synch1_node1
V12 = 1     #Synch1_node2
#----
V13 = 0     #Synch2_node0
```

```
V14 = 0     #Synch2_node1
V15 = 0     #Synch2_node2
#----
V16 = 0     #Synch3_node0
V17 = 0     #Synch3_node1
V18 = 0     #Synch3_node2
# Shafts or connetor nodes
V19 = 1     #Connector node 1
V20 = 1     #Connector node 2
V21 = 1     #Connector node 3
V22 = 1     #Connector node 4
```

```
n_nodes_max = 23
ROWS = range(n_nodes_max)
COLS = range(n_nodes_max)
DIGITS = range(0, 2)
V =
[V0,V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,V2
1,V22]
VARS = ['V%s_V%s' %(row,col) for row in ROWS for col in COLS]
```

Problem definition:

```
problem = Problem()
```

Variables and domains definition:

```
# DECLARE THE VARIABLES IN THE CSP PROBLEM
for var in VARS:
    problem.addVariables([var], DIGITS)
```

Constraints definition:

```
#C000 SYMMETRIC MATRIX
for i in range(len(ADJACENTS)):
    for vargroup in [ADJACENTS[i]]:
        problem.addConstraint(AllEqualConstraint(),vargroup)
```

Solution:

```
sols = problem.getSolutions()
```

FINAL WORD

This thesis is lovingly dedicated to my parents, to my future family, to my sisters.
You are part of the motivations that led to this work.

For all your support, thank you Yuan, Emmanuel, Rochdi, Clément, Wissam.
Special thanks to Emmanuel for your help.
Thank you Charbel Mansour.

To my Spotify playlists, my friends in Lyon, my gym, Lavazza coffee, BU grange blanche, the coffee breaks with Davo, the cool places in Lyon, my parents in Lebanon, my sisters...
To Imane.
Thank you for making this journey easier and more beautiful.

Bilal Kabalan, Lyon, 12/01/2020

BIBBLIOGRAPHY

---

[1] "Tomorrow In Motion - Carlos Tavares | Média Groupe PSA." [Online]. Available: https://media.groupe-psa.com/fr/tomorrow-motion-carlos-tavares. [Accessed: 05-Jan-2020].

[2] Groupe PSA, "Intelligent technologies for plug-in vehicles," *Innovation Day 2016*, pp. 1–10, 2016.

[3] Renault-Nissan-Mitsubishi, "Alliance 2022: New plan targets annual synergies of €10 billion and forecasts unit sales of 14 million & combined revenues of $240 billion," pp. 1–4, 2017.

[4] Volkswagen, "The Volkswagen Group launches the most comprehensive electrification initiative in the automotive industry with 'Roadmap E,'" *Volkswagen Gr. Media Serv.*, no. 308, pp. 1–4, 2017.

[5] IEA-International Energy Agency, "CO2 emissions from fuel combustion," 2018.

[6] "COP21: The key points of the Paris Agreement." [Online]. Available: https://www.diplomatie.gouv.fr/en/french-foreign-policy/climate-and-environment/2015-paris-climate-conference-cop21/cop21-the-paris-agreement-in-four-key-points/. [Accessed: 05-Jan-2020].

[7] M. Baude, F.-X. Dussud, M. Ecoiffier, J. Duvernoy, and C. Vailles, "Chiffres clés du climat France, Europe et Monde - Edition 2018," *Commis. général au développement durable*, p. 80, 2018.

[8] CITEPA, "Gaz à effet de serre et polluants atmosphériques - Bilan des émissions en France de 1990 à 2017 - Extrait," p. 46, 2019.

[9] European Comission, "Europe accelerates the transition to clean mobility: Co-legislators agree on strong rules for the modernisation of the mobility sector," 2018. [Online]. Available: https://ec.europa.eu/clima/news/europe-accelerates-transition-clean-mobility-co-legislators-agree-strong-rules-modernisation_en. [Accessed: 15-Jan-2019].

[10] PFA, "Contribution des transports routiers à la réduction de la demande énergétique et des émissions de CO2 à l'horizon 2030 dans le monde," 2017.

[11] The Guardian, "The Volkswagen emissions scandal explained." [Online]. Available: https://www.theguardian.com/business/ng-interactive/2015/sep/23/volkswagen-

emissions-scandal-explained-diesel-cars. [Accessed: 06-Jan-2020].

[12] G. Hitchcock, D. Birchby, C. Bouvet, and D. Clarke, “Oxford Zero Emission Zone Feasibility and Implementation Study,” no. 1, 2017.

[13] Grenoble-Alpes Métropole, “La réglementation de la zone à faibles émissions dans la Métropole grenobloise.” [Online]. Available: https://www.lametro.fr/761-la-zone-a-faibles-emissions.htm. [Accessed: 11-Jan-2020].

[14] Reuters, “Greater Paris to ban old diesel cars from summer 2019.” [Online]. Available: https://www.reuters.com/article/us-france-paris-pollution/greater-paris-to-ban-old-diesel-cars-from-summer-2019-idUSKCN1NH2BC. [Accessed: 06-Jan-2020].

[15] P. K. Senecal and F. Leach, “Diversity in transportation: Why a mix of propulsion technologies is the way forward for the future fleet,” *Results Eng.*, vol. 4, Dec. 2019.

[16] Reuters, “France to uphold ban on sale of fossil fuel cars by 2040.” [Online]. Available: https://www.reuters.com/article/us-france-autos/france-to-uphold-ban-on-sale-of-fossil-fuel-cars-by-2040-idUSKCN1TC1CU. [Accessed: 06-Jan-2020].

[17] Norwegian Ministry of Transport and Communications, “National Transport Plan 2018-2029,” 2018.

[18] The Netherlands, “Draft Integrated National Energy and Climate Plan 2021-2030 (2018),” p. 120, 2018.

[19] A. Ghoshal, “Watch: India unveils ambitious plan to have only electric cars by 2030,” *Int. Bus. Times*, 2017.

[20] UK Government, “UK plan for tackling roadside nitrogen dioxide concentrations: An overview,” 2017.

[21] Groupe PSA, “Voiture électrique et hybride : technologie.” [Online]. Available: https://www.groupe-psa.com/fr/groupe-automobile/innovation/groupe-psa-lelectrification-en-marche/. [Accessed: 06-Jan-2020].

[22] Reuters, “Peugeot CEO: All PSA vehicles will electrified by 2025.” [Online]. Available: https://www.reuters.com/article/autoshow-detroit-peugeot/peugeot-ceo-all-psa-vehicles-will-electrified-by-2025-idUSL1N1PD030. [Accessed: 06-Jan-2020].

[23] “Groupe PSA and Punch Powertrain are negotiating terms of a Joint Venture agreement for the Group’s future electrified transmissions | Media Groupe PSA.” [Online]. Available: https://media.groupe-psa.com/en/groupe-psa-and-punch-powertrain-are-negotiating-terms-joint-venture-agreement-group’s-future?idtok=a343ebbb237. [Accessed: 06-Jan-2020].

[24] Renault, “E-TECH : la voiture hybride adaptée à tous.” [Online]. Available: https://www.renault.fr/decouvrez-renault/easylife/vehicules-hybrides-e-tech.html. [Accessed: 06-Jan-2020].

[25] IEA-International Energy Agency, “Global EV Outlook 2019,” 2019. [Online]. Available: https://www.iea.org/reports/global-ev-outlook-2019. [Accessed: 06-Jan-2020].

[26] “LAU | SOE | Industrial and Mechanical Engineering | B.E. in Mechanical Engineering.” [Online]. Available: http://soe.lau.edu.lb/ime/programs/be-mechanical/. [Accessed: 03-Jan-2020].

[27] “Powertrain Engineering | IFP School.” [Online]. Available: https://www.ifp-school.com/en/training/specialized-engineering-graduate-degree/powertrain-engineering. [Accessed: 03-Jan-2020].

[28] C. C. Chan, “The state of the art of electric, hybrid, and fuel cell vehicles,” *Proc. IEEE*, vol. 95, no. 4, pp. 704–718, 2007.

[29] C. C. Chan, A. Bouscayrol, and K. Chen, “Electric, hybrid, and fuel-cell vehicles: Architectures and modeling,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 589–598, 2010.

[30] K. Muta, M. Yamazaki, and J. Tokieda, “Development of new-generation hybrid system THS II-Drastic improvement of power performance and fuel economy,” 2004.

[31] U. D. Grebe and L. T. Nitz, “Voltec – The Propulsion System for Chevrolet Volt and Opel Ampera,” *MTZ worldwide*, vol. 72, no. 5, Springer Automotive Media, pp. 4–11, 15-May-2011.

[32] E. Vinot, “Comparison of different power-split hybrid architectures using a global optimization design method.,” *Int. J. Electr. Hybrid Veh.*, vol. 8, no. 3, pp. 225–241, 2016.

[33] R. Trigui, E. Vinot, and B. Jeanneret, “Backward Modeling and Energy Management Optimization of a Two Clutches Series-Parallel HEV for Efficiency Assessment,” in *Power plant and power system control symposium IFAC PPPSC), 2012*, 2012.

[34] H. Abdelbasset, “Exploration en conception mécanique préliminaire des compromis entre contraintes architecturales véhicule et performances vibro-acoustiques agrégées,” *Sci. l’ingénieur*, no. January 2008, 2008.

[35] E. Silvas, T. Hofman, N. Murgovski, P. Etman, and M. Steinbuch, “Review of Optimization Strategies for System-Level Design in Hybrid Electric Vehicles,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 57–70, 2017.

[36] W. van Harselaar, T. Hofman, and M. Brouwer, “Automated Dynamic Modeling of

Arbitrary Hybrid and Electric Drivetrain Topologies," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6921–6934, 2018.

[37] S. Onori, L. Serrao, and G. Rizzoni, *Hybrid Electric Vehicles, Energy Management Strategies*. 2016.

[38] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[39] E. Vinot, R. Trigui, Y. Cheng, C. Espanet, A. Bouscayrol, and V. Reinbold, "Improvement of an EVT-based HEV using dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 40–50, 2014.

[40] X. Zhang, C.-T. Li, D. Kum, and H. Peng, "Prius(+) and Volt(-): Configuration Analysis of Power-Split Hybrid Vehicles With a Single Planetary Gear," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3544–3552, 2012.

[41] L. . Pontryagin, V. . Boltyanskii, R. . Gamkrelidze, and E. . Mishchenko, *The mathematical theory of optimal processes*. New York: Inderscience Publishers, 1962.

[42] E. Vinot and B. Jeanneret, "Fuel Consumption vs Pollutant Emission Trade-Off for Hybrid Electric Vehicle: An Application of the Pontryagin's Minimum Principle," in *Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE*, 2014, pp. 1–6.

[43] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on Pontryagin's minimum principle," *Control Syst. Technol. IEEE Trans.*, vol. 19, no. 5, pp. 1279–1287, 2011.

[44] J. Zhao, "Design and Control Co-Optimization for Advanced Vehicle Propulsion Systems," 2017.

[45] T. Nüesch, P. Elbert, M. Flankl, C. Onder, and L. Guzzella, "Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs," *Energies*, vol. 7, no. 2, pp. 834–856, 2014.

[46] L. Serrao, S. Onori, and G. Rizzoni, "ECMS as a realization of Pontryagin's minimum principle for HEV control," in *Proceedings of the 2009 conference on American Control Conference*, 2009, pp. 3964–3969.

[47] X. Zhang, H. Peng, and J. Sun, "A near-optimal power management strategy for rapid component sizing of multimode power split hybrid vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 609–618, 2015.

[48] X. Zhang, S. E. Li, H. Peng, and J. Sun, "Efficient Exhaustive Search of Power-Split Hybrid Powertrains With Multiple Planetary Gears and Clutches," *J. Dyn. Syst. Meas. Control*, vol. 137, no. December, pp. 1–12, 2015.

[49] X. Zhang, S. E. Li, H. Peng, and J. Sun, “Design of Multimode Power-Split Hybrid Vehicles - A Case Study on the Voltec Powertrain System,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4790–4801, 2016.

[50] V. Reinbold, E. Vinot, and L. Gerbaud, “Joint Optimization of Control and Sizing of the Parallel HEV using SQP Algorithm,” in *Optimization & Inverse Problems in Electromagnetism (OIPE), 2014*, 2014.

[51] E. Vinot, “Contribution aux méthodes de gestion et de dimensionnement des véhicules hybrides.”

[52] N. Murgovski, L. Johannesson, J. Sjöberg, and B. Egardt, “Component sizing of a plug-in hybrid electric powertrain via convex optimization,” *Mechatronics*, vol. 22, no. 1, pp. 106–120, 2011.

[53] B. Kabalan, E. Vinot, Y. Cheng, R. Trigui, C. Dumand, and T. El Hajji, “Efficiency Improvement of a Series-Parallel Hybrid Electric Powertrain by Topology Modification,” *IEEE Trans. Veh. Technol.*, pp. 1–1, 2019.

[54] E. Silvas, E. Bergshoeff, T. Hofman, and M. Steinbuch, “Comparison of Bi-level Optimization Frameworks for Sizing and Control of a Hybrid Electric Vehicle,” pp. 0–5, 2014.

[55] S. Ebbesen, C. Dönitz, and L. Guzzella, “Particle swarm optimisation for hybrid electric drive-train sizing,” *Int. J. Veh. Des.*, vol. 58, no. 2–4, pp. 181–199, Jun. 2012.

[56] M. Pourabdollah, E. Silvas, N. Murgovski, M. Steinbuch, and B. Egardt, “Optimal sizing of a series PHEV: Comparison between convex optimization and particle swarm optimization,” in *IFAC-PapersOnLine*, 2015, vol. 28, no. 15, pp. 16–22.

[57] S. Gan, D. Chrenko, A. Kéromnès, and L. Le Moyne, “Development of a Multi-Architecture and Multi-Application Hybrid Vehicle Design and Management Tool,” *Energies*, vol. 11, no. 11, p. 3185, Nov. 2018.

[58] S. Gan, D. Chrenko, P. Bouillot, and L. Le Moyne, “Multi Architecture Optimization of a Hybrid Electric Vehicle Using Object-Oriented Programming,” in *2017 IEEE Vehicle Power and Propulsion Conference, VPPC 2017 - Proceedings*, 2018, vol. 2018-Janua, pp. 1–5.

[59] W. Gao and C. Mi, “Hybrid vehicle design using global optimisation algorithms,” *Int. J. Electr. Hybrid Veh.*, vol. 1, no. 1, pp. 57–70, 2007.

[60] L. Wu, Y. Wang, X. Yuan, and Z. Chen, “Multiobjective optimization of HEV fuel economy and emissions using the self-adaptive differential evolution algorithm,” *IEEE*

Trans. Veh. Technol., vol. 60, no. 6, pp. 2458–2470, Jul. 2011.

[61] M. Le Guyadec, "Dimensionnement multi-physique des véhicules hybrides, de leurs composants et de la commande du système," *Univ. Grenoble Alpes*, 2018.

[62] E. Silvas, E. Bergshoeff, T. Hofman, and M. Steinbuch, "Comparison of Bi-Level Optimization Frameworks for Sizing and Control of a Hybrid Electric Vehicle," in *Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE*, 2014, pp. 1–6.

[63] T. Hofman, S. Ebbesen, and L. Guzzella, "Topology Optimization for Hybrid Electric Vehicles With Automated Transmissions," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2442–2451, 2012.

[64] W. Zhuang *et al.*, "Simultaneous Optimization of Topology and Component Sizes for Double Planetary Gear Hybrid Powertrains," *Energies*, vol. 9, no. 6, p. 411, 2016.

[65] W. Lhomme, A. Bouscayrol, and P. Barrade, "Simulation of a series hybrid electric vehicle based on energetic macroscopic representation," in *IEEE International Symposium on Industrial Electronics*, 2004, vol. 2, pp. 1525–1530.

[66] X. Zhang, "Design of Power Split Hybrid Powertrains with Multiple Planetary Gears and Clutches," 2015.

[67] X. Zhang, H. Peng, J. Sun, and S. Li, "Automated Modeling and Mode Screening for Exhaustive Search of Double-Planetary-Gear Power Split Hybrid Powertrains," in *ASME 2014 Dynamic Systems and Control Conference*, 2014, p. V001T15A002--V001T15A002.

[68] E. Silvas, T. Hofman, A. Serebrenik, and M. Steinbuch, "Functional and Cost-Based Automatic Generator for Hybrid Vehicles Topologies," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1561–1572, 2015.

[69] J. Wijkniet and T. Hofman, "Modified Computational Design Synthesis Using Simulation-Based Evaluation and Constraint Consistency for Vehicle Powertrain Systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8065–8076, 2018.

[70] S. Masfaraud, F. Danes, P.-E. Dumouchel, F. De Vuyst, and N. Vayatis, "Automatized gearbox architecture design exploration by exhaustive graph generation," *WCCM XII*, no. July, 2016.

[71] N. Fremau, A. Villeneuve, A. Ketfi-Cherif, and A. Vignon, "Transmission pour vehicule automobile a propulsion hybride et procédé de commande associé," WO2015107275A1, 2015.

[72] N. Fremau, A. Villeneuve, A. Ketfi-Cherif, and A. Vignon, "Hybrid transmission with

offset electric machine and method for controlling gear changes," WO2015197927A1, 2015.

[73] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems : Algorithms and applications," *Eur. J. Oper. Res.*, vol. 119, pp. 557–581, 1999.

[74] A. Bejan, Y. W. Kwon, H. Bang, and L. R. Davis, *Enumeration of Kinematic Structures Fundamentals of Environmental Discharge Modeling*. 2001.

[75] S. Dasgupta, C. . Papadimitriou, and U. . Vazirani, *Algorithms*, McGraw-Hil. 2008.

[76] J. WIELEMAKER, T. SCHRIJVERS, M. TRISKA, and T. LAGER, "SWI-Prolog," *Theory Pract. Log. Program.*, vol. 12, no. 1–2, pp. 67–96, Jan. 2012.

[77] G. Niemeyer, "Python-Constraint," *Constraint Solving Problem resolver for Python*. [Online]. Available: https://labix.org/python-constraint.

[78] E. Hebrard, E. O'Mahony, and B. O'Sullivan, "Constraint Programming and Combinatorial Optimisation in Numberjack," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems: 7th International Conference, CPAIOR 2010, Bologna, Italy, June 14-18, 2010. Proceedings*, A. Lodi, M. Milano, and P. Toth, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 181–185.

[79] X. Olive, "facile Documentation," 2018.

[80] "OR-Tools | Google Developers." [Online]. Available: https://developers.google.com/optimization/. [Accessed: 10-Nov-2019].

[81] A. Hagberg, D. Schult, and P. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 2008.

[82] E. Vinot, J. Scordia, R. Trigui, B. Jeanneret, and F. Badin, "Model simulation, validation and case study of the 2004 THS of Toyota Prius," *Int. J. Veh. Syst. Model. Test.*, vol. 3, no. 3, pp. 139–167, 2008.

[83] M. André, "The ARTEMIS European driving cycles for measuring car pollutant emissions," *Sci. Total Environ.*, vol. 334–335, pp. 73–84, Dec. 2004.

[84] V. Reinbold, E. Vinot, L. Garbuio, and L. Gerbaud, "Optimal sizing of an electrical machine using a magnetic circuit model: application to a hybrid electrical vehicle," *IET Electr. Syst. Transp.*, vol. 6, no. 1, pp. 27–33, 2016.

[85] ORNL, "Evaluation of 2004 Toyota Prius hybrid electric drive system," 2006.

[86] M. Le Guyadec, L. Gerbaud, E. Vinot, and B. Delinchant, "Sensitivity analysis using

Sobol indices for the thermal modelling of an electrical machine for sizing by optimization," *COMPEL - Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 38, pp. 965–976, 2019.

[87] J. Goos, C. Criens, and M. Witters, "Automatic Evaluation and Optimization of Generic Hybrid Vehicle Topologies using Dynamic Programming," 2017.

[88] T. El Hajji, B. Kabalan, Y. Cheng, E. Vinot, and C. Dumand, "Sensitivity Analysis on the Sizing Parameters of a Series-Parallel HEV," in *IFAC Symposium on Advances in Automotive Control AAC*, 2019, vol. 52, no. 5, pp. 405–410.

[89] B. Kabalan, E. Vinot, Y. Cheng, R. Trigui, and C. Dumand, "Improvement of a Series-Parallel Hybrid Electric Vehicle Architecture," in *2017 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2017, pp. 1–6.

[90] B. Kabalan, Y. Cheng, E. Vinot, C. Dumand, R. Trigui, and W. Bou Nader, "Optimal Design and Sizing of Through-The-Road Hybrid Vehicle Powertrain," in *SIA Powertrain 2018*, 2018, pp. 1–7.

[91] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm : NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[92] U.S.DepartmentofEnergy, S. Rogers, and S. Boyd, "Overview of the DOE VTO Electric Drive Technologies R&D Program," 2016, p. 8.

[93] luxresearch, "Energy storage : Li - ion battery pack prices will fall to $ 172 / kWh by 2025 for the best - in - class players," 2015.

[94] E. Vinot, "Time reduction of the dynamic programming computation in the case of hybrid vehicle," in *Optimization & Inverse Problems in Electromagnetism (OIPE), 2014*, 2014, pp. S213–S227.

[95] O. Sundstrom, D. Ambuhl, and L. Guzzella, "On implementation of dynamic programming for optimal control problems with final state constraints," *Oil Gas Sci. Technol. l'Institut Fran{ç}ais du P{é}trole*, vol. 65, no. 1, pp. 91–102, 2010.

[96] E. Vinot, V. Reinbold, and R. Trigui, "Global Optimized Design of an Electric Variable Transmission for HEVs," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6794–6798, 2016.

[97] N. Higuchi and H. Shimada, "Efficiency enhancement of a new two-motor hybrid system," *World Electr. Veh. J.*, vol. 6, no. 2, pp. 325–335, 2013.

[98] S. Washino, T. Saito, and Y. Jia, "New 3 Mode Hybrid System Concept," *24th Aachen Colloq. Automob. Engine Technol. 2015*, pp. 1123–1139, 2015.

[99] Eurostat, “Eurostat statistic on population displacement, http://epp.eurostat.ec.europa.eu/portal/page/portal/transport/data/database http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=road_tf_road&lang=fr.” .

N°d'ordre NNT : xxx

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

Ecole Doctorale N° 160
Électronique, Électrotechnique, Automatique (EEA)

Spécialité de doctorat : Génie électrique
Discipline : Sciences pour l'ingénieur

Soutenue publiquement le 10/03/2020, par :

Bilal KABALAN

Méthodologie de génération systématique et de conception des chaines de traction de véhicules hybrides

---

Devant le jury composé de :

M. Hofman Theo, Professeur Associé, Eindhoven University of Technology (TU/e)	Rapporteur
Mme Chrenko Daniela, Maître de Conférences, Université de Technologie de Belfort-Montbéliard	Rapporteuse
M. Sari Ali, Professeur des Universités, Université Claude Bernard Lyon1	Examinateur
Mme Semail Betty, Professeure des Universités, Université de Lille	Examinatrice
M. Trigui Rochdi, Directeur de Recherche, IFSTTAR	Directeur de thèse
M. Vinot Emmanuel, Chargé de recherche, IFSTTAR	Examinateur
M. Dumand Clément, Responsable Industriel, Groupe PSA	Invité

Université Claude Bernard – LYON 1

Président de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-Président du Conseil d'Administration	M. Didier REVEL
Vice-Président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALLIER
Vice-Président de la Commission de Recherche	M. Jean-François MORNEX
Directeur Général des Services	M. Damien VERHAEGHE

COMPOSANTES SANTE

Faculté de Médecine Lyon-Est – Claude Bernard	Doyen : M. Gilles RODE
Faculté de Médecine et Maïeutique Lyon Sud Charles. Mérieux	Doyenne : Mme Carole BURILLON
UFR d'Odontologie	Doyenne : Mme Dominique SEUX
Institut des Sciences Pharmaceutiques et Biologiques	Directrice : Mme Christine VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. Xavier PERROT
Département de Formation et Centre de Recherche en Biologie Humaine	Directrice : Mme Anne-Marie SCHOTT

COMPOSANTES & DEPARTEMENTS DE SCIENCES & TECHNOLOGIE

UFR Biosciences	Directrice : Mme Kathrin GIESELER
Département Génie Electrique et des Procédés (GEP)	Directrice : Mme Rosaria FERRIGNO
Département Informatique	Directeur : M. Behzad SHARIAT
Département Mécanique	Directeur M. Marc BUFFAT
UFR - Faculté des Sciences	Administrateur provisoire : M. Bruno ANDRIOLETTI
UFR (STAPS)	Directeur : M. Yannick VANPOULLE
Observatoire de Lyon	Directrice : Mme Isabelle DANIEL
Ecole Polytechnique Universitaire Lyon 1	Directeur : Emmanuel PERRIN
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur : Gérard PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. Christophe VITON
Institut de Science Financière et d'Assurances	Directeur : M. Nicolas LEBOISNE
ESPE	Administrateur Provisoire : M. Pierre CHAREYRON

Synthèse en Français

RESUME

Méthodologie de génération systématique et de conception des chaines de traction de véhicules hybrides

Pour répondre aux objectifs de consommation des flottes de véhicules, au normes d'émissions de polluants et aux nouvelles demandes de l'usager, les constructeurs automobiles doivent développer des motorisations hybrides et électriques. Réaliser une chaine de traction hybride reste cependant une tâche difficile. Ces systèmes sont complexes et possèdent de nombreuses variables réparties sur différents niveaux : architecture, technologie des composants, dimensionnement et contrôle/commande. L'industrie manque encore d'environnements et d'outils pouvant aider à l'exploration de l'ensemble de l'espace de dimensionnement et à trouver la meilleure solution parmi tous ces niveaux. Cette thèse propose une méthodologie systématique pour répondre au moins partiellement à ce besoin. Partant d'un ensemble de composants, cette méthodologie permet de générer automatiquement tous les graphes d'architectures possibles en utilisant la technique de programmation par contraintes. Une représentation dédiée est développée pour visualiser ces graphes. Les éléments de boites de vitesse (embrayages, synchroniseurs) sont représentés avec un niveau de détails approprié pour générer de nouvelles transmission mécaniques sans trop complexifier le problème. Les graphes obtenus sont ensuite transformés en d'autres types de représentation : 0ABC Table (décrivant les connections mécaniques entre les composants), Modes Table (décrivant les modes de fonctionnement disponibles dans les architectures) et Modes Table + (décrivant pour chaque mode le rendement et le rapport de réduction global des chemins de transfert de l'énergie entre tous les composants). Sur la base de cette représentation, les nombreuses architectures générées sont filtrées et seules les plus prometteuses sont sélectionnées. Elles sont ensuite automatiquement évaluées et optimisées avec un modèle général spécifiquement développé pour calculer les performances et la consommation de toute les architectures générées. Ce modèle est inséré dans un processus d'optimisation à deux niveaux ; un algorithme génétique GA est utilisé pour le dimensionnement des composants et la programmation dynamique est utilisée au niveau contrôle (gestion de l'énergie) du système. Un cas d'étude est ensuite réalisé pour montrer le potentiel de cette méthodologie. Nous générons ainsi automatiquement toutes les architectures qui incluent un ensemble de composants défini à l'avance, et le filtrage automatique élimine les architectures présupposées non efficaces et sélectionnent les plus prometteuses pour l'optimisation. Les résultats montrent que la méthodologie proposée permet d'aboutir à une architecture meilleure (consommation diminuée de 5%) que celles imaginées de prime abord (en dehors de toute méthodologie).

Mots clé: dimensionnement de chaine de traction, véhicules électriques hybrides, optimisation, algorithmes génétiques, programmation dynamique, programmation par contraintes, génération automatique d'architectures.

Table des matières

1. CONTEXTE DES TRAVAUX DE RECHERCHE

1.1 La motivation

Dans un contexte de réglementation automobile visant à réduire les émissions de gaz à effet de serre notamment (nouvelle normes européennes visant à réduire de 37.5 % les émissions des flottes de voitures particulières des constructeur en 2030 par rapport à 2020 [1]), la conception optimale des chaines de traction des véhicules hybrides et hybrides rechargeables est un sujet qui intéresse les chercheurs et les constructeurs automobiles, dont le Groupe PSA partenaire de ce travail.

Ces chaines de traction sont des systèmes complexes de par la multiplicité des sources et des chemin d'énergie nécessaire au mouvement du véhicule. Pour réaliser une optimisation globale, les leviers d'optimisation à considérer peuvent être séparés sur 3 niveaux :

O Niveau 1 : Le choix de l'architecture

O Niveau 2 : Le choix des composants et leur dimensionnement

O Niveau 3 : Le contrôle du système (chois des modes électriques ou hybrides et du courant de la batterie).

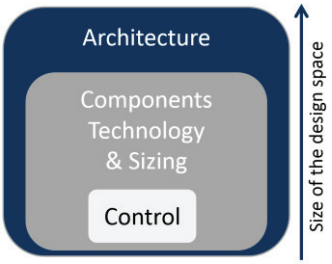


Figure 1: Niveau de conception des chaines de tractions pour (P)HEV

Ces 3 niveaux sont très dépendants et donc on ne peut pas optimiser le système sur un seul niveau en figeant les autres. Une optimisation globale est nécessaire. Les verrous sont : la grande dimension de l'espace d'optimisation, la complexité du contrôle (gestion de l'énergie), et l'exploration de l'espace de toutes les architectures possibles.

Cette conception doit prendre en compte des contraintes (comme la performance du véhicule) des objectifs pouvant être liés aux caractéristiques du système (coût, volume...) et/ou au comportement du système sur un cycle de conduite (consommation cumulée ou pollution cumulée).

1.2.Objectif de la thèse

L'objectif du travail de thèse est de Créer une méthodologie systématique qui aidera à la conception optimale des chaines de traction hybrides, tout en répondant aux demandes et aux objectifs du constructeur, avec une focalisation sur le problème du choix de la meilleure architecture.

Classiquement, le développement d'une chaine de traction se déroule selon un cycle en V représenté en Figure 2 avec le positionnement de cette thèse.

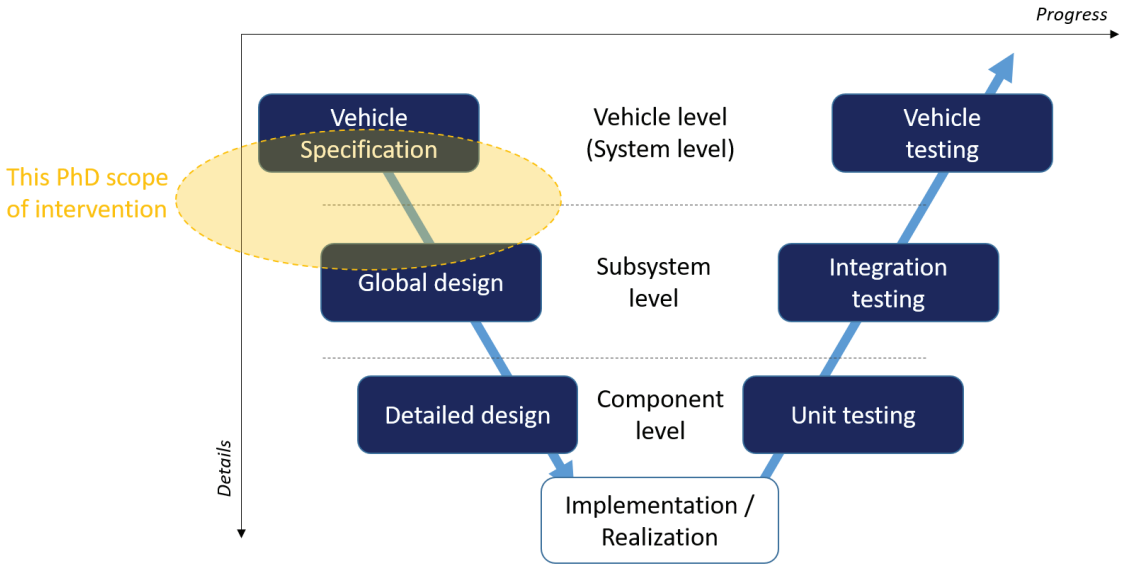


Figure 2: Cycle en V pour le développement d'une chaine de traction

Le processus de design part de la définition des spécifications du véhicule (segment, performances requises, objectifs de consommation et d'émissions…). Dans la phase de pré-dimensionnement où les ingénieurs choisissent les variables de la chaine de traction qui respectent les spécifications du véhicule, les simulations numériques sont largement utilisées

pour évaluer et optimiser le dimensionnement. Pour calculer les performances, la consommation et les émissions, des modèles énergétique quasi-statiques sont traditionnellement utilisés.

La complexité du problème de dimensionnement des véhicules hybrides (incluant les hybrides rechargeable et noté (P)HEV dans le reste du document) est souvent augmentée par rapport aux chaines de tractions conventionnelles ou purement électriques, car, au moins deux systèmes de propulsion existent et rajoutent des dégrées de liberté au système.

Le premier degré de liberté ajouté est le choix de l'architecture : la manière dont sont connecté les deux systèmes de traction. En effet, les chaines de traction hybrides combinent des systèmes de traction électrique (alimenté par batterie) avec un système conventionnel basée sur un moteur thermique. Ces deux systèmes peuvent être connectés de différentes manières : séries, parallèles et séries-parallèles qui sont les principales familles d'architectures [2].

Une fois l'architecture choisie, différentes technologies de composants peuvent être sélectionnée et dimensionnée (taille batterie, puissance des composants, rapports de réduction …). Le fonctionnement de la chaine de traction et la consommation sur un cycle de conduite donnée dépendront des composants choisis, de leur dimensionnement et enfin des lois de gestion de l'énergie. Une interaction forte existe entre ces différentes variables.

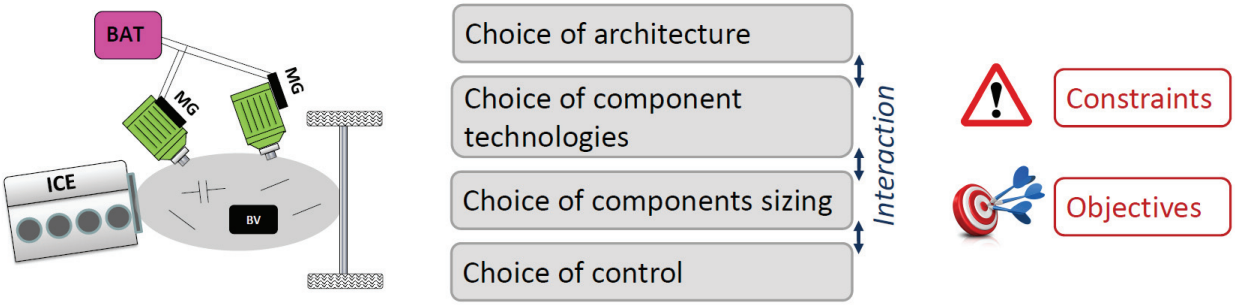


Figure 3: Le problème de design des (P)HEV

Pour choisir ces variables, il faut donc regarder le système dans sa globalité et respecter les contraintes au niveau composant (vitesse et couple max. par exemple) et système (vitesse et accélération max. du véhicule…). Cela en minimisant des objectifs qui peuvent être par exemple la consommation, les émissions sur un cycle, la compacité de la chaine de traction …

Cette thèse propose de donner des éléments de réponse à cette problématique en créant une méthodologie de pré-dimensionnement permettant de répondre à cette tâche complexe de dimensionnement multi-variables et multi-niveaux.

1.3.Contributions de cette thèse

1- Intégration de la génération automatique d'architecture dans le processus d'optimisation en explorant tout l'espace possible au lieu de présélectionner quelques architectures sur des critères "experts". A noter que les architectures à dérivation de puissance ne sont pour l'instant pas prises en compte dans la méthodologie.

2- Développement d'un modèle général de véhicules hybrides permettant l'évaluation de toutes les architectures générées et retenues et ainsi de connecter le niveau exploration des architectures et celui de l'optimisation des composants et de la gestion.

2. ETAT DE L'ART

On retrouve dans la littérature beaucoup de travaux sur l'optimisation couplée du système et de sa commande (Niveaux 2 et 3) pour une architecture hybride sélectionnée. Les méthodes optimales pour le niveau de commande sont principalement la Programmation Dynamique (DP) [3] et le Principe du Minimuim de Pontryagin (PMP) [4]. D'autres méthodes non-optimales sont parfois utilisées pour leur simplicité et leur vitesse de calcul (ECMS, Rule-based,…). Au niveau de l'algorithme d'optimisation des paramètres du système (niveau 2) on retrouve des travaux qui se font en utilisant des algorithmes de type génétique (GA) [5][6], Particle Swarm Optimisation (PSO) [7], Dividing Rectangles [8], ou bien des méthodes du type SQP (Sequential Quadratic Programming). Au laboratoire Eco7, une méthodologie d'optimisation a déjà été mise en place lors de travaux précédents (thèse de V.Reinbold [9] et M. Le-Guyadec [10] notamment). Cette méthode utilise un algorithme génétique pour le dimensionnement du système et des composants et la programmation dynamique pour la gestion du système.

Concernant l'exploration du niveau 1, on trouve assez peu de travaux. Des premiers travaux de génération automatique d'architecture ont déjà débuté à l'Université du Michigan et

à l'université d'Eindhoven [11][12]. Sur l'ensemble du procédé d'optimisation incluant les 3 niveaux, les méthodologies restent donc à concevoir sinon à consolider et à valider.

On trouve dans la littérature plusieurs exemples de travaux qui partent d'un petit nombre d'architectures présélectionnées puis les optimisent sur les niveaux (2) et (3) afin de les comparer (Table 1).

Table 1: Exemples de méthodologies d'optimisation sur les niveaux (2) et (3)

	IFSTTAR [5][10]	IFSTTAR [13]	IFPEN [14]	IFPEN [14]	TU Eindhoven [8][15]	U Michigan [16]
Components technology and Sizing	GA	SQP	DIRECT	FACE	GA; DIRECT; PSO; SQP; ES	ES
Control	DP		PMP; GRAB-ECO		DP	PEARS+
Coordination approach	Nested	Simultaneous	Nested	Simultaneous	Nested	Nested

Quelques travaux commencent également à créer des plateformes permettant d'évaluer et de comparer plusieurs véhicules, cycles de conduites, architectures, taille de composants et gestion de l'énergie [17] and [18].

3. *PRESENTATION DE LA METHODOLOGIE DEVELOPPEE*

Pour répondre au problème de dimensionnement des (P)HEV ces travaux proposent la méthodologie présentée Figure 4.

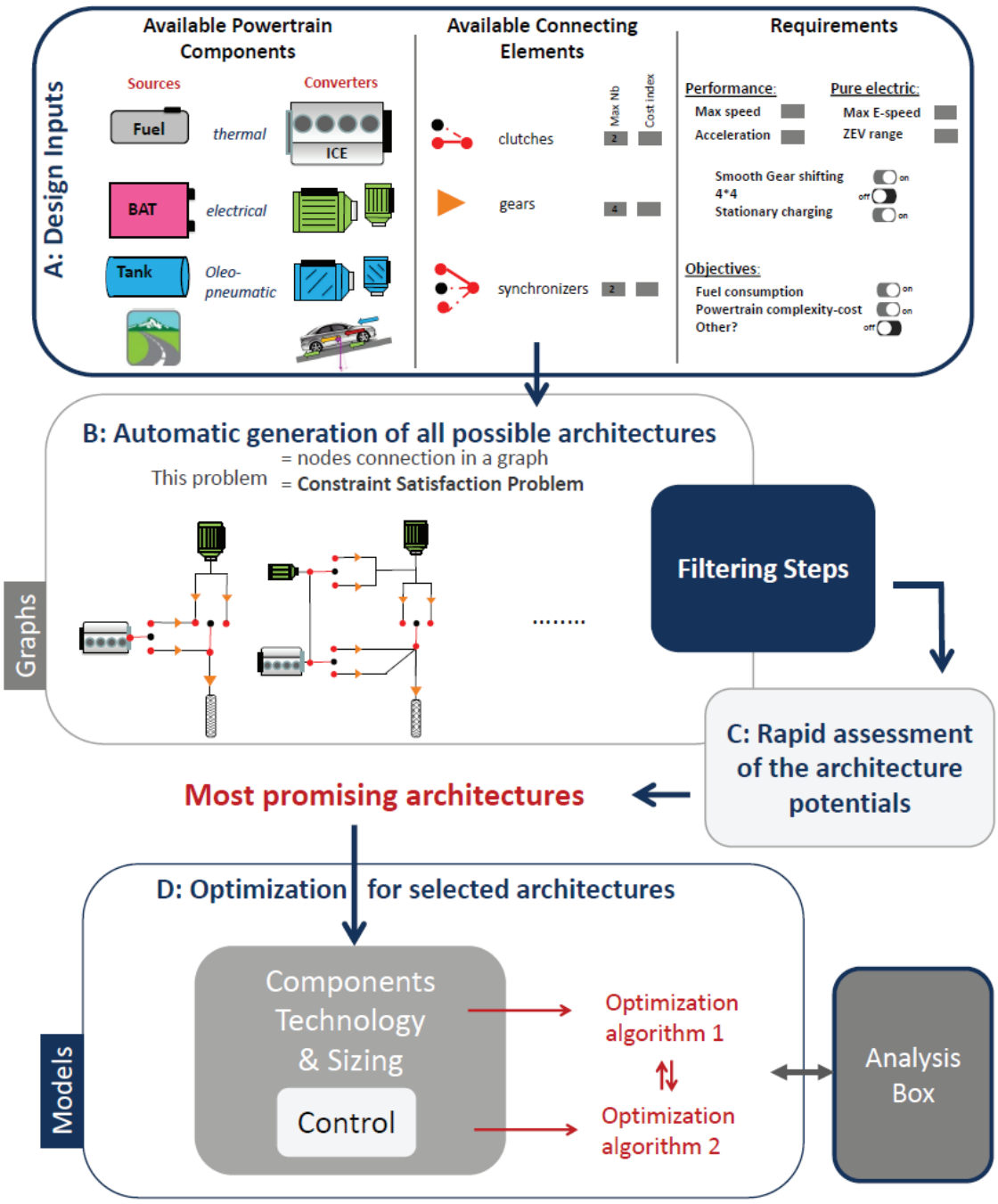


Figure 4: La méthodologie proposée

Elle se décompose en quatre parties principales (A, B, C et D).

A: Grandeurs d'entrées

Cette partie définie les composants disponibles pour réaliser la chaine de traction qui peuvent provenir de différents domaines de la physique : thermique, électrique, pneumatique (moteurs thermiques et électriques disponibles, batterie…). En plus, les éléments de connexions

de ces composants sont sélectionnés (embrayages, réducteurs, synchroniseurs …). Cette partie définie aussi ce que l'on souhaite : performance du véhicule (vitesse max, accel. Max, autonomie en mode électrique …) et les objectifs à minimiser (consommation sur différents cycle, coût …).

B: Génération automatique de toutes les architectures possibles :

Le but de cette partie est de générer toutes les architectures possibles avec les composants et éléments de connexions définis en A. Ces architectures sont représentées sous formes de graphe et la méthode tente de trouver toutes les possibilités de connections entre les nœuds de ce graphe. Le problème est écrit sous forme d'un problème de satisfaction de contrainte (CSP, voir partie 5). Les nombreux graphes générés doivent ensuite être filtrés pour ne conserver que ceux qui sont réalisables.

C: Evaluation rapide du potentiel de ces architectures:

Certains des graphes générés peuvent être réalisables mais ne présentent pas un bon potentiel pour être parmi les meilleures à cause, par exemple, de l'absence de certains modes de fonctionnement. Cela peut être détecté dans cette partie et ces architectures sont éliminées et ne seront pas évaluées par la suite.

D: Optimisation des architectures sélectionnées:

Les architectures les plus prometteuses doivent être comparées de façon à sélectionner l'architecture optimale pour les objectifs fixés. Des modèles numériques sont alors nécessaires. Développer un modèle pour chaque architecture étant quasi irréalisable, le but de cette partie est de développer un modèle « générique » de simulation des architectures sélectionnées. Ces architectures seront optimisées sur les deux niveaux ; technologie et dimensionnement des composants et contrôle du système. Deux algorithmes d'optimisation sont utilisés pour cela (cf. partie 8). Un exemple d'application complète de cette méthodologie est présenté dans la partie 9.

Une représentation graphique (Figure 5) est choisie et permet de représenter les chaines de tractions hybrides sous forme de graphe : un ensemble de nœuds (composants) connecté par des arcs (connexions). Les nœuds sont de deux types (graph bipartie) : les composants (cercles de couleurs) et les connecteurs représentants les arbres mécaniques du système (petit cercle en noir). Aucune connexion directe ne peut être réalisée entre composants. Elles doivent être réalisées à travers les arbres.

Les composants des nœuds sont:

- Les moteurs thermiques (ICE) et électriques (MGs), et le rapport de réduction final du différentiel (FD)
- Les composants de connexions : boites de vitesse, embrayages, composants des unités de synchronisation (Synchroniseur + 2 rapports de réduction)

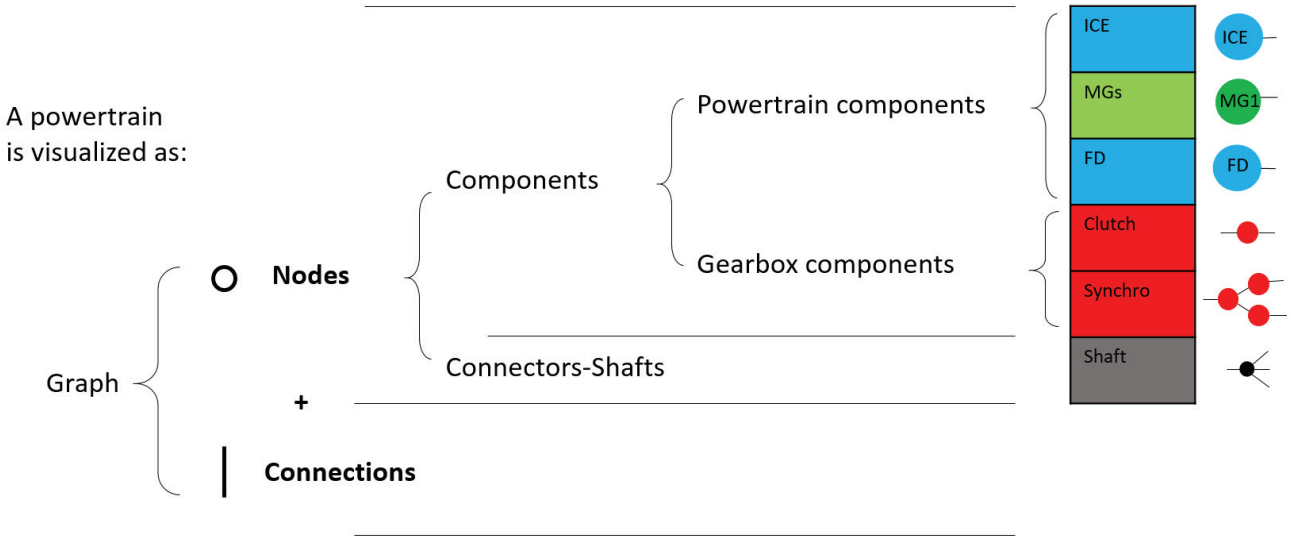
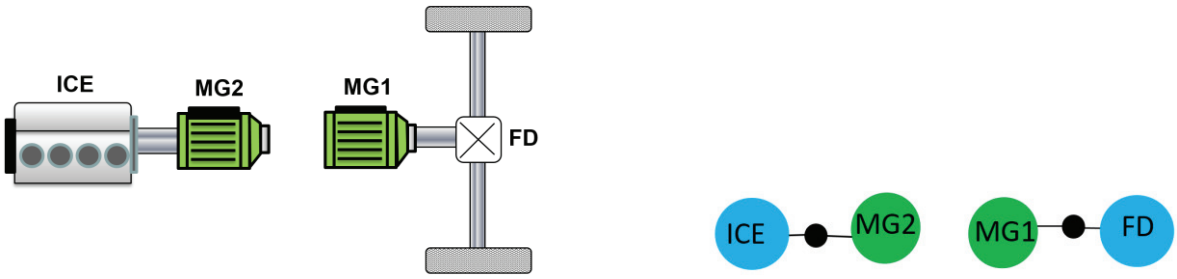


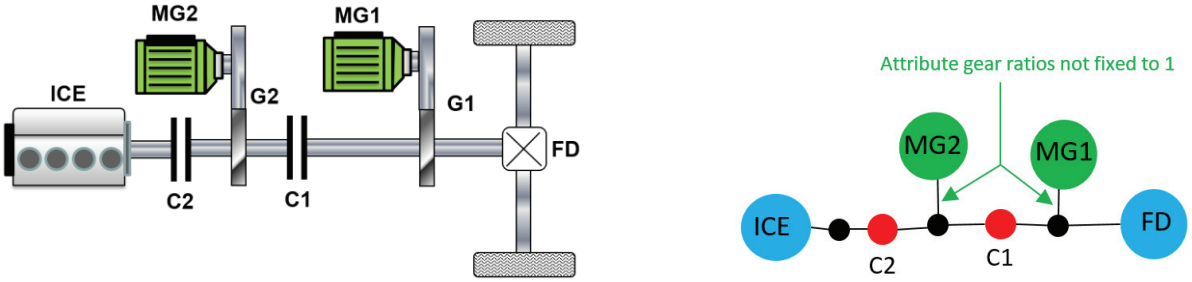
Figure 5: La représentation graphique proposée

Ci-dessous quelques exemples de représentation d'architectures traditionnelles :

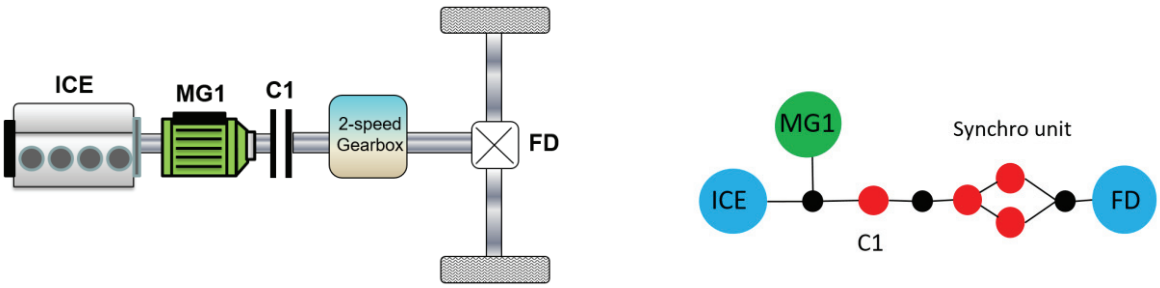
- Architecture série:



- Architecture serie-parallèlle avec réducteurs sur les machines:



- Architecture parallèle avec boîte de vitesse à deux rapports:



5. PROGRAMMATION PAR CONTRAINTES

Un problème de satisfaction de contraintes (CSP) est un problème défini comme suit [19]:

- Un ensemble de variables $X = \{ x_1,\dots, x_n\}$,
- Chaque x_i a un domaine de valeurs possible D_i
- Un ensemble de contraintes $C_{i,j,k..}$ limitant le nombre de valeurs possibles que les variables x_i, x_j, x_k, ... peuvent prendre simultanément

Trouver une solution possible à ce problème consiste à assigner une valeur à chaque variable dans son domaine de validité qui respectent les contraintes spécifiées. On peut choisir de rechercher une solution, toute les solutions ou une solution optimale si un objectif est spécifié.

Le problème de génération automatique d'architectures hybrides peut être formulé comme un problème de satisfaction de contraintes (CSP) :

Variables du problèmes et domaines de validité:

Comme expliqué précédemment une architecture de chaine de traction hybride peut être vue comme un graphe : un ensemble de nœuds connectés par des arcs. Ce graph est connecté et non-orienté puisque les connexions n'ont pas de directions.

Soit un graph G composé d'un ensemble de nœuds $\boldsymbol{V}$ connectés par un ensemble d'arcs $\boldsymbol{E}$. Dans les logiciels de création de graph, G peut être géré comme dans [20], [21] :

- Une liste adjacente:
  Listes des n nœuds $\boldsymbol{V} = \{ V_0, \dots V_{n-1}\}$ et des arcs $\boldsymbol{E} = \{ \{V_0, V_1\}, \dots \{V_{n-2}, V_{n-1}\} \}$
- Une matrice adjacente:
  Matrice n par n ou les colonnes sont les nœuds $V_0, \dots V_{n-1}$, les lignes sont les noeuds $V_0, \dots V_{n-1}$, et chaque cellule à l'index (i, j) prend la valeur 0 ou 1 selon l'absence (0) ou la présence (1) de connexion entre les nœuds V_i and V_j.

Dans ce travail nous considérons que la chaine de traction hybride peut contenir au maximum:

- 1 moteur thermique (ICE)
- 2 machines électriques (MGs)
- 1 réducteur final (FD)
- 4 arbres (X1 à X4 Figure 6)
- 3 embrayages (C1_R à C3_L)
- 3 synchroniseurs (S1_0 à S3_L)

Pour résoudre le problème de génération d'architecture le cas d'une chaine de traction comprenant tous ces composants est considéré. Etant donnée l'absence de contraintes de taille de mémoire, l'option de matrice adjacente est choisie. Le nombres de nœuds est alors de 23. La matrice adjacente est alors présentée Figure 6.

						clutches						synchro									shafts			
		ICE	MG1	MG2	FD	C1_R	C1_L	C2_R	C2_L	C3_R	C3_L	S1_0	S1_R	S1_L	S2_0	S2_R	S2_L	S3_0	S3_R	S3_L	X1	X2	X3	X4
		V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
ICE	V0	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
MG1	V1	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
MG2	V2	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
FD	V3	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C1_R	V4	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C1_L	V5	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C2_R	V6	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C2_L	V7	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C3_R	V8	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
C3_L	V9	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_0	V10	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_R	V11	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S1_L	V12	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_0	V13	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_R	V14	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S2_L	V15	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_0	V16	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_R	V17	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
S3_L	V18	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X1	V19	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X2	V20	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X3	V21	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]
X4	V22	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]	[0,1]

Figure 6: Matrice adjacente de la chaine de traction comprenant tous les composants

Les lignes et les colonnes correspondent au nœuds de tous les composants $V_0, \dots V_{22}$. En bleu et vert on trouve les composants de traction et le réducteur final (ICE, MGs, FD), en rouge les nœuds composants des « boites de vitesse » (embrayages et synchroniseurs), et en gris les nœuds des arbres. La matrice a alors $23 * 23 = 529$ cellules qui peuvent prendre les valeurs 1 ou 0 selon que les éléments correspondants à la ligne et à la colonne sont connectés ou pas. Le nombre initial de solutions est alors de :

$$2^{529} = 1.7574\, e + 159 \quad (1)$$

Générer les graphes de toutes les architectures possibles avec les composants définis et en respectant les contraintes est alors équivalent à trouver toutes les matrices adjacentes qui respectent ces contraintes. Autrement dit, le problème de génération de graphes est un CSP où les variables sont les 529 cellules de la matrice 23*23, le domaine de chacune de ces variables est {0, 1}, et l'objectif est de trouver toute les solutions possibles.

Un certain nombre de contraintes listées (tableau 1) ont été ajoutées. Ces contraintes sont la transposition de « contraintes » mécaniques de faisabilité de la chaine de traction. Elles ont permis de réduire le nombre de solution possible à :

$$2^{76} = 7.5558\text{e+}22$$

Tableau 1: Les contraintes considérées

C000	Symmetric matrix (undirected graph)
C001	Nodes can be connected to 'shafts' only
C002	[Min, Max] # connections
C003	ICE and FD cannot be connected to same shaft
C004	MG1 and MG2 cannot be connected to same shaft
C005	Clutches nodes can't be connected alone to 1 same shaft; Same for synchro nodes
C006	Connectors should be connected to at least 1 (ICE-MG-FD) component
C007	Synchro S_0 and S_L or S_R cannot be connected to same node
C008	2 clutches can 't be in parallel
C009	1 clutch and 1 synchro cannot be in parallel

De façon à résoudre ce problème, le module Python-Constraint [22] a été sélectionné. Ce choix a été réalisé en tenant compte de la simplicité d'implémentation et des possibilité offertes par Python de disposer facilement de module en accès libre (open source) de visualisation de graphe notamment.

6. *EXEMPLE DE GENERATION DE GRAPH*

Le module NetworkX [23] de Python est utilisé pour transformer les matrices adjacentes en graphes visualisables par l'utilisateur. Un exemple avec un moteur thermique, une machine électrique ; deux synchroniseurs, trois arbres et un réducteur final est compilé. Parmi les 132 solutions générées par l'algorithme 6 exemples d'architectures sont donnés ci-dessous (Figure 7).

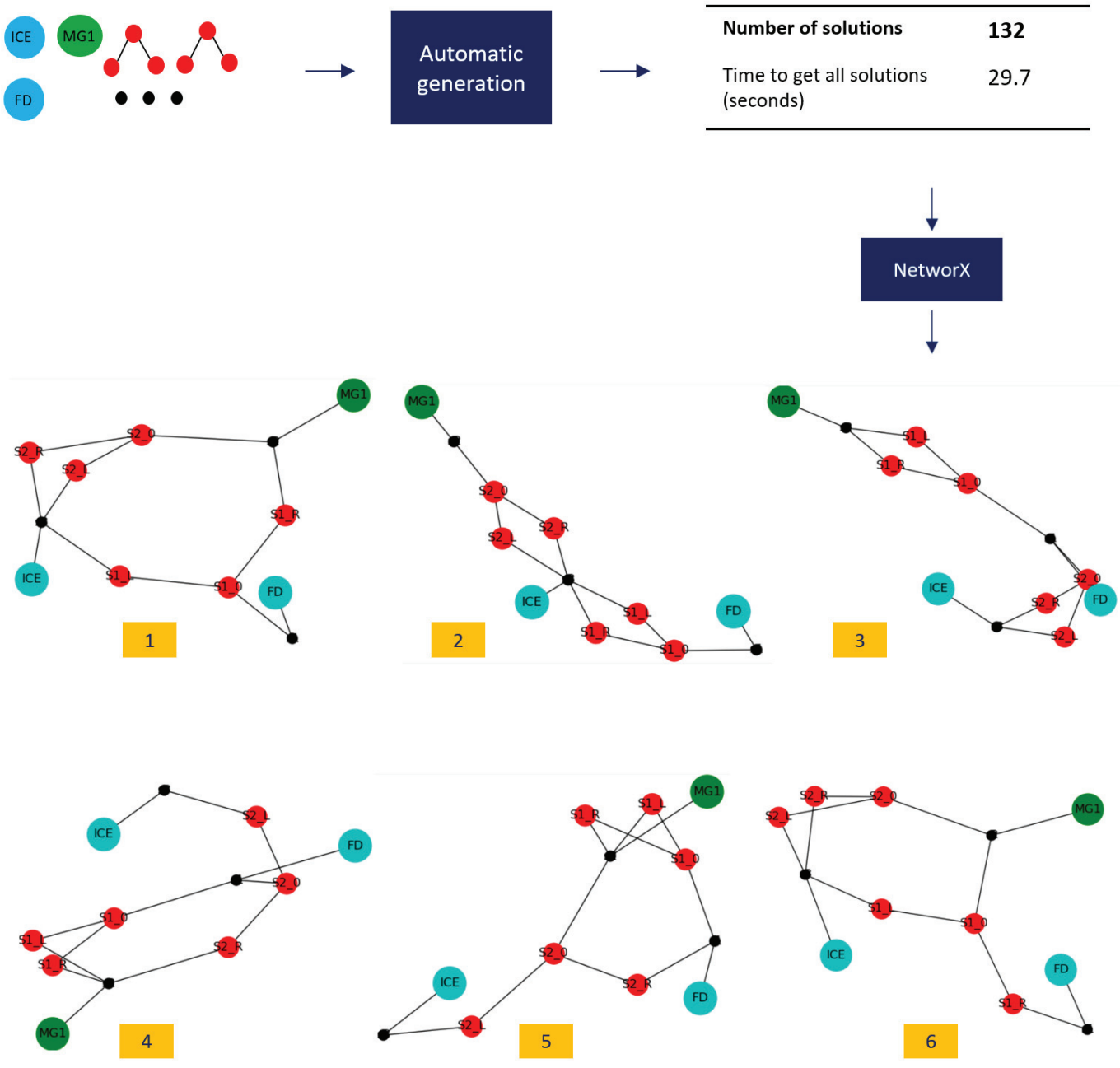


Figure 7: Exemple de génération de graphes

7. FILTRAGE AUTOMATIQUE ET ANALYSE DES GRAPHES

7.1. Tableau 0ABC Table

Dans l'exemple de la Figure 7, 6 solutions sont représentées où le placement des synchroniseur et des composants est différents. En fait ce n'est pas le cas pour les 132 solutions trouvées initialement. Certains graphes sont redondants ou isomorphes à cause de problème de symétrie. Les différences entre certains graphes sont uniquement des différences entre l'orientation des composants (synchroniseur par exemple) mais conduisent en fait à la même chaine cinématique et au même modèle énergétique.

Une représentation appelée '0ABC Table' a été développée pour détecter ces redondances. C'est un tableau décrivant le type et le nombre de connexions entre les éléments de la chaine de traction (Figure 9).

Ces connexions peuvent être classées en 4 types:

- 0 : pas de connexion entre les composants
- A: connexion directe, les deux composants sont connectés au même arbre.
- B: connexion par synchroniseur placé entre 2 arbres.
- C: connexion par embrayage placé entre 2 arbres.

La Figure 8 fournie un exemple dans un cas très simple (1 moteur et deux synchroniseurs) de quatre graphes ayant la même table 0ABC (Figure 9):

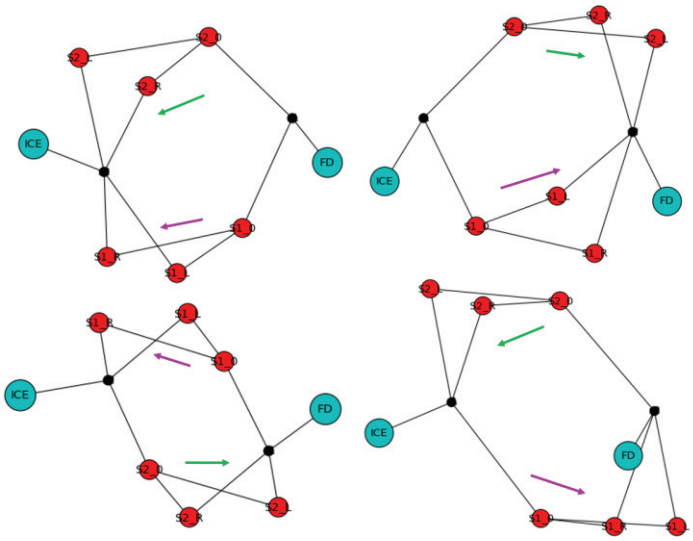


Figure 8: Exemple de redondance

	0	A	B	C
ICE<->FD	0	0	4	0
MGs<->FD	0 0	0 0	0 0	0 0
ICE<->MGs	0 0	0 0	0 0	0 0
MGs<->MGs	0	0	0	0

Figure 9: 0ABC table des 4 graphes en Figure 8

Si plusieurs architectures on la même 0ABC table, on conserve la première. Cette première étape de filtrage permet de réduire grandement le nombre d'architecture retenue. Par exemple dans le cas de la Figure 7 on passe de 132 graphes à 12.

7.2. Tableau des modes

Dans un premier temps les différents états de fonctionnement que pourra prendre chaque architecture est déterminé automatiquement. En effet, un embrayage peut prendre deux positions (ouvert ou fermé) et un synchroniseur trois (déconnecté, connecté sur le rapport 1, connecté sur le rapport 2). Chaque architecture possédant un nombre d'embrayage $N\_clutches$ et un nombre

de synchroniseur $N\_synchros$ a donc $2^{N\_clutches} \times 3^{N\_synchros}$ états. Chacun de ces états correspond à un mode de fonctionnement différent ou des rapports de réduction différents.

L'outil développé sous Python considère chaque graphe comme un 'Graphe Parent'. Pour chaque Graphe Parent, $2^{N\_clutches} \times 3^{N\_synchros}$ 'Graphes d'état' sont déduits. Pour chacun de ces graphes les modes de fonctionnement sont automatiquement détectés (Figure 10).

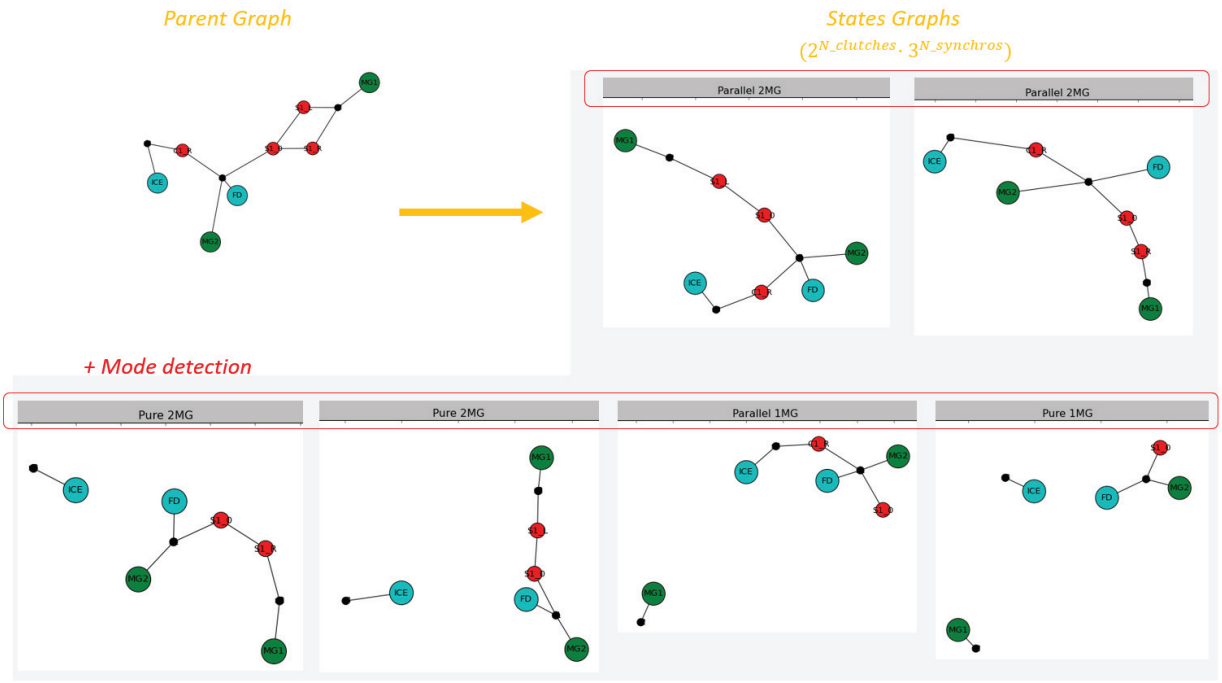


Figure 10: Du 'Graphe Parent' au 'Graphes d'état'

On détecte neuf modes de fonctionnement :

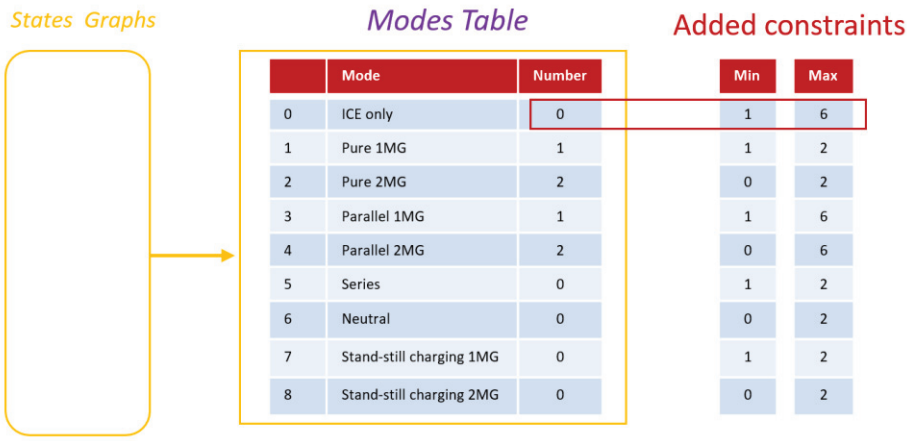
- **ICE only:** seul le moteur thermique est connecté aux roues.
- **Pure 1MG:** seul une machine électrique est connectée aux roues.
- **Pure 2MG:** les deux machines électriques sont connectées aux roues.
- **Parallel 1MG:** ICE et une MG sont connectés aux roues.
- **Parallel 2MG: :** ICE et deux MG sont connectés aux roues.
- **Series:** 1 MG est connectée aux roues et ICE est connecté à une autre MG
- **Neutral:** les roues ne sont connectées à aucun composants et il n'y as pas de connexions entre les autres composants.

- **Stand-still charging 1MG**: **:** les roues ne sont connectées à aucun composants, ICE est connecté à 1 MG.
- **Stand-still charging 2MG:** les roues ne sont connectées à aucun composants, ICE est connecté à 2 MGs.

A partir de ces modes, un tableau de modes est créé pour chaque Graph (Figure 11). Ce tableau donne la liste des différents modes avec une valeur 1 si le mode existe et 0 sinon. A ce stade les graphes peuvent donc être comparés sur la base de leurs modes de fonctionnement.

De plus, des contraintes peuvent être ajoutées comme par exemple le nombre minimum et maximum d'instances de chaque mode.

Dans l'exemple de la Figure 11, on peut décider qu'une architecture ne possédant pas au moins un mode « pure ICE » est supprimée. Ce qui est le cas de l'architecture de la figure 10 vu la comparaison de l'encadré rouge sur la figure 11.



	Mode	Number	Min	Max
0	ICE only	0	1	6
1	Pure 1MG	1	1	2
2	Pure 2MG	2	0	2
3	Parallel 1MG	1	1	6
4	Parallel 2MG	2	0	6
5	Series	0	1	2
6	Neutral	0	0	2
7	Stand-still charging 1MG	0	1	2
8	Stand-still charging 2MG	0	0	2

Figure 11: Modes table du graphe parent de la Figure 10

7.3. Modes Table +

Pour comparer et évaluer (cf. partie 8) plus précisément les architectures et finalement les optimiser, nous devons disposer d'informations supplémentaires sur les chemins de transfert d'énergies entre les composants et leurs efficacités.

Chaque chemin de transfert de l'énergie entre les composants est décrit par deux valeurs:

- Son rendement global
- Son rapport de réduction global

Un rendement et un rapport de réduction sont assignés à chaque nœuds du graphe d'état (Figure 12).

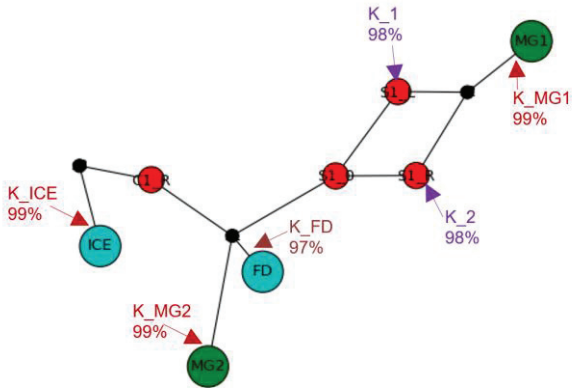


Figure 12: Rapports assignés aux nœuds

Ces informations sont alors retranscrites dans un tableau appelé Modes Table + (Figure 13). On retrouve dans ce tableau tous les modes possibles pour une architecture et pour chacun de ces modes le rendement et le rapport de réduction global pour chaque chemin de transfert de l'énergie. A noter que les rapports de réduction ne sont, à ce stade, pas fixés et le seront dans la phase de dimensionnent (cf. partie 8.2).

		Description
	Mode	1st instance
1	ICE only	K_ICE_FD, η_ICE_FD
2	Pure 1MG	MG = 1 or 2 K_MG_FD, η_MG_FD
3	Pure 2MG	K_MG1_FD, η_MG1_FD K_MG2_FD, η_MG2_FD K_MG1_MG2, η_MG1_MG2
4	Parallel 1MG	MG = 1 or 2 K_ICE_FD, η_ICE_FD K_MG_FD, η_MG_FD K_ICE_MG, η_ICE_MG
5	Parallel 2MG	K_ICE_FD, η_ICE_FD K_MG1_FD, η_MG1_FD K_MG2_FD, η_MG2_FD K_ICE_MG1, η_ICE_MG1 K_ICE_MG2, η_ICE_MG2 K_MG1_MG2, η_MG1_MG2
6	Series	MG= 1 or 2 K_MG_FD, η_MG_FD K_ICE_MG, η_ICE_MG
7	Nutral	-
8	Stand-still charging 1MG	MG=1 or 2 K_ICE_MG, η_ICE_MG
9	Stand-still charging 2MG	K_ICE_MG1, η_ICE_MG1 K_ICE_MG2, η_ICE_MG2 K_MG1_MG2, η_MG1_MG2

Figure 13: Modes Table + avec le rendement et le rapport de réduction global pour chaque chemin de transfert de l'énergie, dans chaque mode

8. *PROCEDURE D'EVALUATION AUTOMATIQUE*

8.1. Modélisation de la chaine de traction

Des modèles énergétiques des chaines de traction sont développés en utilisant la bibliothèque VEHLIB [24] de modélisation de véhicule hybrides. Une approche backward est utilisée pour calculer la consommation, et une approche forward pour calculer les performances du véhicule.

Chaque composant da la chaine de traction est modélisée par des modèles classiques :

- Les moteurs électriques par des cartographies de pertes en fonction du couple et de la vitesse de rotation

- Le moteur thermique par une cartographie de consommation en fonction du couple et de la vitesse de rotation
- La batterie par des modules en séries modélisés par des circuits électriques (OCV-R)
- Les réducteurs par des rapports de réduction et des rendements constants

Le dimensionnement des composants sera fixé par un processus d'optimisation avec les variables suivantes :

- Les puissances max. du moteur thermique et des machines électriques (facteur d'échelle sur les cartographies)
- Le nombre de modules de la batterie en séries
- Les rapports de réduction des réducteurs
- Les rapports de réduction des boites de vitesse

La gestion de l'énergie est déterminée de façon optimale par la programmation dynamique [25] ce qui permet de calculer la consommation sur des cycles de fonctionnement qui sont une donnée d'entré du problème.

8.2. Méthodologie de dimensionnement

Pour chaque architecture sélectionnée, un processus d'optimisation à deux niveaux est appliqué (Figure 14). Un algorithme génétique NSGA-II [26] est utilisé pour l'optimisation des variables de dimensionnement.

Pour chaque candidat évalué par cet algorithme les contraintes sur les performances sont vérifiées. Si elles sont respectées les fonctions objectifs sont évaluées. Dans ce travail nous considérons généralement 500 générations pour arrêter le processus.

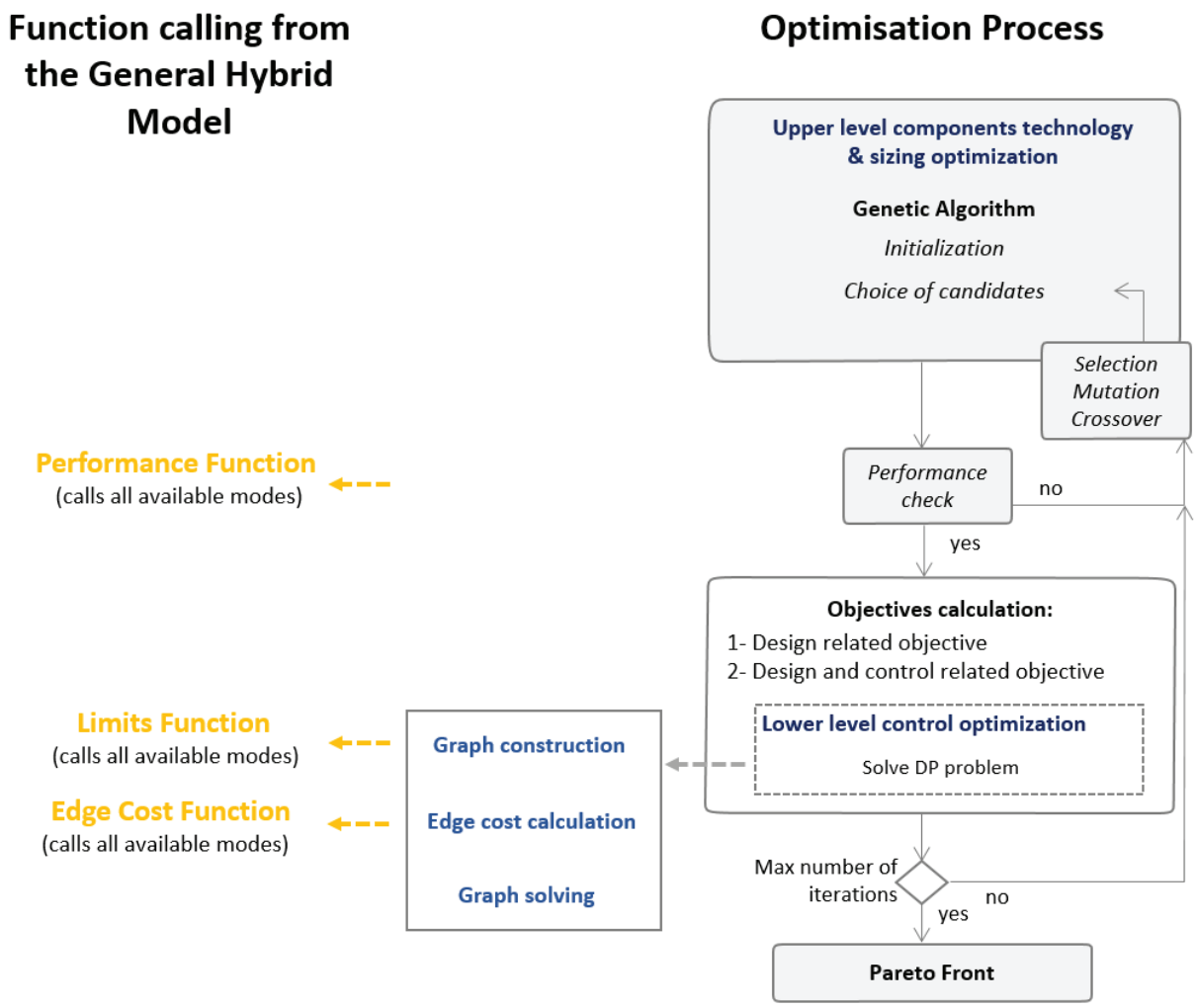


Figure 14: Processus d'optimisation à deux niveaux

Quatre contraintes de performances sont considérées :

- L'accélération de 0 à 100 km/h:
  $t_{0->100} < value$
- L'accélération de 80 à 120 km/h:
  $t_{80->120} < value$
- La vitesse maximum du véhicule sur plat : $V_{max} > value$
- L'autonomie en mode tout électrique > *value*

Pour calculer les trois premières contraintes (2 accélérations et la vitesse maximum), une 'fonction performance ' est appelée (Figure 15). Le véhicule part d'une position à l'arrêt puis à

chaque instant, la fonction calcule, pour chacun des modes disponibles, le couple maximum qui serait fourni aux roues. Le mode qui maximise ce couple est alors choisi. En mode parallèle les embrayages peuvent patiner (mode glissement) si nécessaire.

L'autonomie en mode tout électrique est calculée séparément sur cycle WLTC.

Pour évaluer la consommation sur cycle d'usage, la programmation dynamique est utilisée comme méthode de gestion [3], [27], [28]. Un pas de temps de 1 seconde est utilisé et le SOC est discrétisé avec environ 1000 points ente le SOC minimum et le SOC maximum voir [29] pour plus de détails.

La résolution du problème de control peut être représenté comme un graphe de SOC de la batterie en fonction du temps (Figure 15). Ce graphe est échantillonné de façon régulière et limité par les SOC min et max autorisés par la batterie. Les points de deux colonnes consécutives sont connectés par des arcs associés à des couts ; dans notre cas une consommation de carburants.

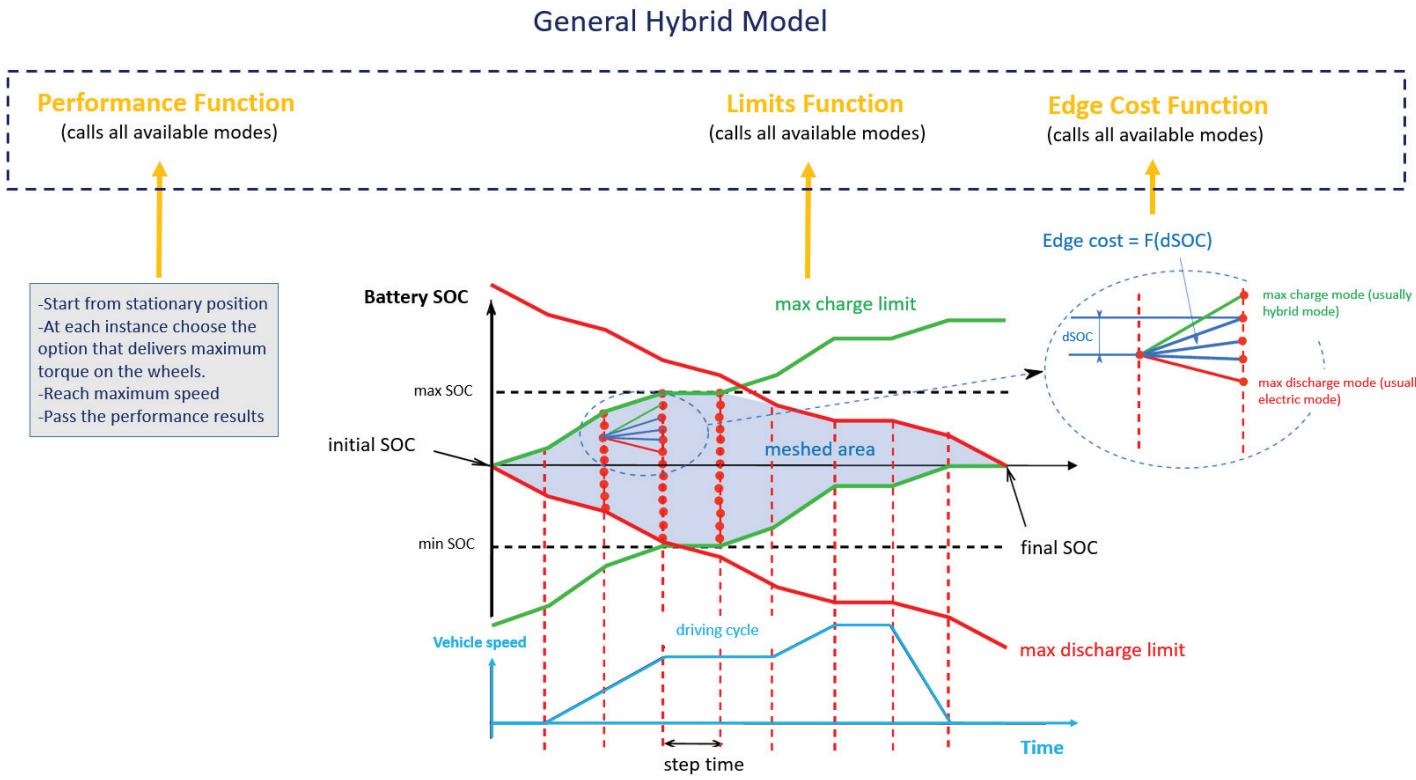


Figure 15: le calcul par programmation dynamique inséré dans le modèle général

La méthode DP commence par construire le graph en déterminant les limites de charge et

de décharge max. de la batterie en vert et en rouge (Figure 15). Une fonction 'Limites' du modèle général de véhicule hybride est utilisée ici. Dans cette fonction, tous les modes possibles sont appelés et ceux correspondant à une charge maximale et minimale sont retenue. Le graph est ainsi limité à une surface entre ces deux limites (Figure 15).

La seconde étape est le calcul du coût des arcs. Une fonction coût est utilisée. Elle calcule pour chaque mode et pour chaque instance de ces modes disponibles la consommation du véhicule sur chaque arc en optimisant le cas échéant les variables de contrôle pour chacun de ces modes. Le coût des arcs est alors le plus petit parmi les consommations calculées.

Une fois les coûts de chaque arcs calculé, DP détermine la trajectoire qui minimise la consommation sur le cycle entre le SOC initial et le SOC final.

Cette consommation est « envoyée » au niveau supérieur où l'algorithme génétique continu le processus d'évaluation jusqu'à un nombre maximum de génération. Un front de Pareto, consommation en fonction du nombre de modules de la batterie en série est alors obtenu (cf. partie 9).

9. APPLICATION ET RESULTATS

Un exemple d'application de ce processus est présenté en utilisant les composants de départ suivants:

- 1 ICE
- 1 FD
- 2 MG
- 4 arbres
- 2 embrayages
- 1 synchroniseur

Le processus de génération automatique d'architecture génère 480 graphes possibles. Après filtrage, en utilisant les tables 0ABC et tableau de modes, 63 architecture réalisables et non redondantes subsistent.

Le choix est alors fait de ne considérer que les architectures ayant au moins :

- 1 mode tout thermique
- 1 mode pure électrique (avec une ou deux machines)
- 1 mode série
- 1 mode parallèle (avec une ou deux machines)
- 1 mode de recharge (Series or Stand-still charging 1MG or 2MG)

46 architectures ne répondent pas à cette spécification et 17 passent cette nouvelle étape de filtrage.

6 de ces architectures ne possèdent qu'un seul choix de rapport de réduction entre le moteur thermique et les roues, ce qui peut limiter l'efficacité en mode tout thermique ou hybride parallèle. Elles sont retirées pour ne retenir dans un premier temps que 11 architectures.

Pour des fins de comparaison, 2 architectures (Figure 16 : architectures 12 et Figure 17: architectures 13) sont ajoutées à ces 11 car correspondant à des architectures existantes ou précédemment évaluées dans d'autres études [30][6] :

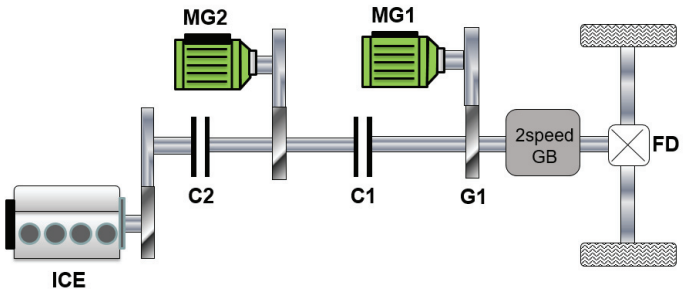


Figure 16: Architecture 12

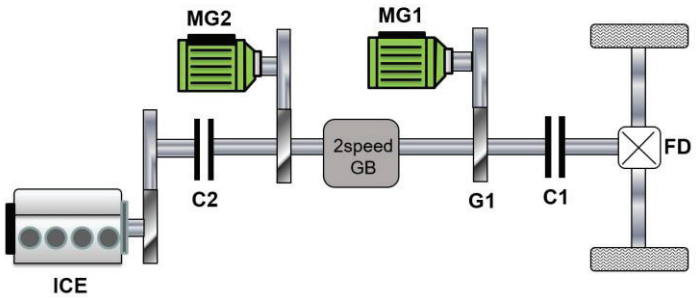


Figure 17: Architecture 13

Les résultats obtenue pour ces 13 architectures et les fronts de Pareto associés sont représentés Figure 18 et montrent que l'architecture 7 semble être la meilleure pour les objectifs considérés.

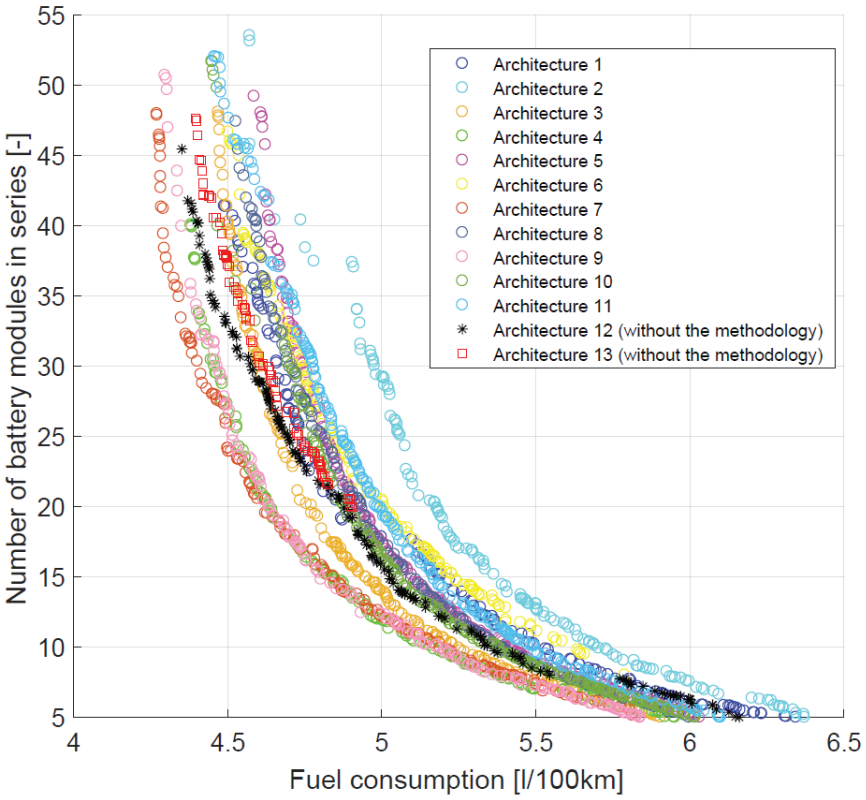


Figure 18: Fronts de Pareto des architectures étudiées

10. CONCLUSION

L'objectif de cette thèse était de présenter une méthodologie systématique de génération et de dimensionnement des chaines de traction de véhicule hybride.

Ce travail se décline en deux contributions principales :

1- Le développement d'un outil de génération automatique et de filtrage d'architectures hybrides basé sur un certain nombre de contraintes et de spécifications issues par exemple d'un constructeur.
2- Le développement d'un modèle général de véhicules hybrides permettant d'évaluer les performances et la consommation de ces véhicules, et notamment les fonctions permettant le calcul des limites du problème de programmation dynamique et du calcul du coût des arcs.

A ce jour, le travail présenté ne permet pas de simuler les architectures à dérivation de puissance à base de trains épicycloïdaux. Le temps de calcul est une autre piste d'amélioration puisqu'une architecture est simulée (optimisée) en 8 à 10h.

11. PERSPECTIVES

1- Réévaluer la précision des modèles de composants et de la technique de dimensionnement.
2- Inclure le choix de la technologie des composants dans la boucle d'optimisation
3- Réduire le temps de calcul en améliorant le code (supprimer les boucles et les calculs inutiles …) et en explorant des techniques de gestion de l'énergies plus rapides (abandon de DP).
4- Inclure les trains épicycloïdaux dans ces travaux et générer et évaluer des architectures à dérivation de puissance (prise en compte dans le modèle général d'évaluation des contraintes et des objectifs).
5- Inclure d'autres objectifs dans le processus (compacité, poids, coût, vieillissement batteries, émissions de polluant...).

[1] European Comission, “Europe accelerates the transition to clean mobility: Co-legislators agree on strong rules for the modernisation of the mobility sector,” 2018. [Online]. Available: https://ec.europa.eu/clima/news/europe-accelerates-transition-clean-mobility-co-legislators-agree-strong-rules-modernisation_en. [Accessed: 15-Jan-2019].

[2] C. C. Chan, A. Bouscayrol, and K. Chen, “Electric, hybrid, and fuel-cell vehicles: Architectures and modeling,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 589–598, 2010.

[3] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[4] L. . Pontryagin, V. . Boltyanskii, R. . Gamkrelidze, and E. . Mishchenko, *The mathematical theory of optimal processes*. New York: Inderscience Publishers, 1962.

[5] E. Vinot, “Comparison of different power-split hybrid architectures using a global optimization design method.,” *Int. J. Electr. Hybrid Veh.*, vol. 8, no. 3, pp. 225–241, 2016.

[6] B. Kabalan, E. Vinot, Y. Cheng, R. Trigui, C. Dumand, and T. El Hajji, “Efficiency Improvement of a Series-Parallel Hybrid Electric Powertrain by Topology Modification,” *IEEE Trans. Veh. Technol.*, pp. 1–1, 2019.

[7] S. Ebbesen, C. Dönitz, and L. Guzzella, “Particle swarm optimisation for hybrid electric drive-train sizing,” *Int. J. Veh. Des.*, vol. 58, no. 2–4, pp. 181–199, Jun. 2012.

[8] E. Silvas, E. Bergshoeff, T. Hofman, and M. Steinbuch, “Comparison of Bi-Level Optimization Frameworks for Sizing and Control of a Hybrid Electric Vehicle,” in *Vehicle Power and Propulsion Conference (VPPC), 2014 IEEE*, 2014, pp. 1–6.

[9] V. Reinbold, “Méthodologie de dimensionne- ment d ’ un moteur électrique pour véhicules hybrides Optimisation conjointe des composants et de,” 2014.

[10] M. Le Guyadec, “Dimensionnement multi-physique des véhicules hybrides, de leurs composants et de la commande du système,” *Univ. Grenoble Alpes*, 2018.

[11] E. Silvas, T. Hofman, A. Serebrenik, and M. Steinbuch, “Functional and Cost-Based Automatic Generator for Hybrid Vehicles Topologies,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1561–1572, 2015.

[12] J. Wijkniet and T. Hofman, “Modified Computational Design Synthesis Using Simulation-Based Evaluation and Constraint Consistency for Vehicle Powertrain Systems,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8065–8076, 2018.

[13] V. Reinbold, E. Vinot, and L. Gerbaud, “Joint Optimization of Control and Sizing of the Parallel HEV using SQP Algorithm,” in *Optimization & Inverse Problems in Electromagnetism (OIPE), 2014*, 2014.

[14] J. Zhao, “Design and Control Co-Optimization for Advanced Vehicle Propulsion Systems,” 2017.

[15] T. Hofman, S. Ebbesen, and L. Guzzella, “Topology Optimization for Hybrid Electric Vehicles With Automated Transmissions,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2442–2451, 2012.

[16] W. Zhuang *et al.*, “Simultaneous Optimization of Topology and Component Sizes for Double Planetary Gear Hybrid Powertrains,” *Energies*, vol. 9, no. 6, p. 411, 2016.

[17] S. Gan, D. Chrenko, P. Bouillot, and L. Le Moyne, “Multi Architecture Optimization of a Hybrid Electric Vehicle Using Object-Oriented Programming,” in *2017 IEEE Vehicle Power and Propulsion Conference, VPPC 2017 - Proceedings*, 2018, vol. 2018-Janua, pp. 1–5.

[18] S. Gan, D. Chrenko, A. Kéromnès, and L. Le Moyne, “Development of a Multi-Architecture and Multi-Application Hybrid Vehicle Design and Management Tool,” *Energies*, vol. 11, no. 11, p. 3185, Nov. 2018.

[19] S. C. Brailsford, C. N. Potts, and B. M. Smith, “Constraint satisfaction problems : Algorithms and applications,” *Eur. J. Oper. Res.*, vol. 119, pp. 557–581, 1999.

[20] A. Bejan, Y. W. Kwon, H. Bang, and L. R. Davis, *Enumeration of Kinematic Structures Fundamentals of Environmental Discharge Modeling*. 2001.

[21] S. Dasgupta, C. . Papadimitriou, and U. . Vazirani, *Algorithms*, McGraw-Hil. 2008.

[22] G. Niemeyer, “Python-Constraint,” *Constraint Solving Problem resolver for Python*. [Online]. Available: https://labix.org/python-constraint.

[23] A. Hagberg, D. Schult, and P. Swart, “Exploring network structure, dynamics, and function using NetworkX,” in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 2008.

[24] E. Vinot, J. Scordia, R. Trigui, B. Jeanneret, and F. Badin, “Model simulation, validation and case study of the 2004 THS of Toyota Prius,” *Int. J. Veh. Syst. Model. Test.*, vol. 3, no. 3, pp. 139–167, 2008.

[25] D. P. Bertsekas, *Dynamic Programming and optimal control*. Belmont, MA: Athena Scientific, 1995.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A Fast and Elitist Multiobjective

Genetic Algorithm : NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[27] E. Vinot, "Time reduction of the dynamic programming computation in the case of hybrid vehicle," in *Optimization & Inverse Problems in Electromagnetism (OIPE), 2014*, 2014, pp. S213–S227.

[28] O. Sundstrom, D. Ambuhl, and L. Guzzella, "On implementation of dynamic programming for optimal control problems with final state constraints," *Oil Gas Sci. Technol. l'Institut Fran{ç}ais du P{é}trole*, vol. 65, no. 1, pp. 91–102, 2010.

[29] E. Vinot, R. Trigui, Y. Cheng, C. Espanet, A. Bouscayrol, and V. Reinbold, "Improvement of an EVT-based HEV using dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 40–50, 2014.

[30] S. Washino, T. Saito, and Y. Jia, "New 3 Mode Hybrid System Concept," *24th Aachen Colloq. Automob. Engine Technol. 2015*, pp. 1123–1139, 2015.