



Systematic methodology for architecture generation and design optimization of hybrid powertrains

Bilal Kabalan, Emmanuel Vinot, Rochdi Trigui, Clément Dumand

► To cite this version:

Bilal Kabalan, Emmanuel Vinot, Rochdi Trigui, Clément Dumand. Systematic methodology for architecture generation and design optimization of hybrid powertrains. IEEE Transactions on Vehicular Technology, 2021, 69 (12), pp.14846 - 14857. 10.1109/TVT.2020.3041501 . hal-03184562

HAL Id: hal-03184562

<https://hal.science/hal-03184562>

Submitted on 29 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Systematic methodology for architecture generation and design optimization of hybrid powertrains

B. Kabalan, E. Vinot, R. Trigui, and C. Dumand

Abstract—Designing a hybrid powertrain remains a complex task. It is an intricate system involving numerous variables that are spread over different levels: architecture, component technologies, sizing, and control. There is currently a lack of frameworks or tools that help in exploring the entire design space and in finding the global optimal solution throughout these levels. This article 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 the graphs. They are then transformed into other types of representation (tables describing the connections and the powertrain modes). Based on these representations, the architectures are automatically filtered and the most promising ones are selected. They are automatically assessed and optimized using a specifically developed general hybrid model that calculates the performance and fuel consumption of all the generated architectures. This model is inserted inside a bi-level optimization process: a Genetic Algorithm is used on the sizing and components level, while the Dynamic Programming is used on the control level. A case study is performed and the capability of the methodology is proven.

Index Terms—Powertrain design, hybrid electric vehicles, optimization, sizing, genetic algorithms, dynamic programming, constraint programming, architecture generation.

I. INTRODUCTION

THE vehicle fleet-wide average CO₂ emission targets keep on decreasing. For instance, the EU made an objective of 37.5% CO₂ reduction in 2030 compared to 2021 [1] for passenger cars. Such numbers cannot be reached with conventional diesel and gasoline powertrains.

To meet these CO₂ targets, the stringent pollutant emissions standards and the users' new demands, the automakers realized the inevitable need to propose more hybrid and electric powertrains. Designing in an optimal way a hybrid powertrain remains however a complex task.

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This paper was submitted for review on April 20, 2020.

B. Kabalan (e-mail: bilal.kabalan01@gmail.com), E. Vinot (e-mail: emmanuel.vinot@univ-eiffel.fr), and R. Trigui (e-mail: rochdi.trigui@univ-eiffel.fr) are with Gustave Eiffel University, IFSTTAR / AME / ECO7, 25 Av. Francois Mitterrand, 69675 Bron, France.

C. Dumand (e-mail: clement.dumand@mpsa.com) is with Groupe PSA, Centre Technique Vélizy A, case courrier VV1415, Route de Gizey, 78943 Vélizy-Villacoublay Cedex, France.

i. (P)HEV design problem

(P)HEV (referring to Hybrid and Plug-in Hybrid Electric Vehicle) powertrains combine a battery electric traction system with a conventional engine-based traction system. Different architectures exist and can be split into 3 main categories: series, parallel and series-parallel [2]. Within a category, different variants can be made by changing the location of the powertrain components and by adding gears or gearboxes.

Once the architecture is chosen, different component technologies can be selected, and different sizing can be made. The vehicle operation and fuel consumption will depend on all these choices and on the energy management during the operation.

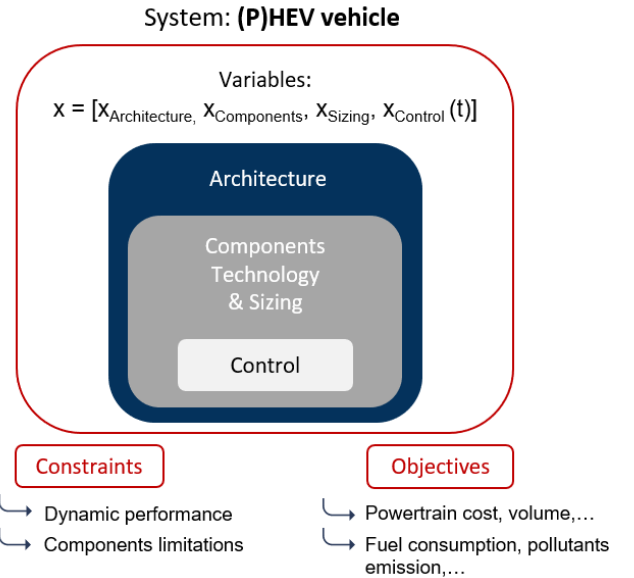


Figure 1: (P)HEV design problem in its optimization context

In view of this, the design of (P)HEV can be seen as a multi-objective optimization problem that is spread over multiple levels [3], [4] (Figure 1). The involved variables can be divided into 3 levels: (1) Architecture level, (2) Components technology and sizing level, (3) Control level.

The variables are to be optimized while respecting constraints coming from a system level (as the vehicle performance) and components level (components limitations). The objective functions to optimize can be design related functions (powertrain cost, volume) or design and control related functions (fuel consumption, pollutants emissions).

This design problem is highly challenging due to the high number of involved variables, the numerous conflicting

considerations, and the multi-objective, non-linear, non-convex character of the problem. In addition, the assessment of all the combinations of variables (brute force or exhaustive search) is not time feasible.

ii. State of the art

In the literature, this design problem is traditionally tackled by optimizing level 2 (Components technology & sizing) and 3 (Control) without including the level of architecture in the optimization. Few benchmark architectures are arbitrarily selected, manually modeled, optimized on the two levels (2) and (3), and then compared [5][3]. Some manual topology modifications in the aim of efficiency improvement are also done. In [6] a gearbox is inserted between the Engine (ICE) and the Motor Generator (MG). Different topologies of series-parallel HEV made by adding gears and gearboxes are studied in [7]. The location of the MG can also be changed and improved as in [8]. Different power-split topologies can also be considered as in [9].

Further, platforms are proposed to compare combinations of vehicle application, architectures, components choice, sizing, and control [10][11]. Sometimes, a general model is used to simulate the few studied architectures. For example, in [12] a general forward Simulink model is developed to assess different architectures by connecting or not some components by means of coefficients equal to a gear ratio or to zero if not connected. General models of power-split architectures are also used in [13][14][15], see below. In [16], a “matrix” based method with 2 clutches and four possible MGs positions allow to simulate a large class of hybrid vehicles with one gearbox.

In the present paper an original general hybrid model based on the operating modes will be proposed. It will be able to simulate all the hybrid architectures with one engine, two electrical machines and complex or multiple gearboxes.

The architecture level has not been sufficiently explored in the literature. This is because listing and modeling manually all the architectures is infeasible. In addition, it is because of the model complexity and computation time burden associated with the architecture’s evaluation and optimization, in case of exhaustive search. Nevertheless, some works started to explore the architecture level by performing automatic generation of architectures instead of sticking to a few known architectures. In [13][15][17], a generation and comparison of power-split architectures is performed based on system matrix generation and analysis. The principle is to use one or two planetary gears (PG) linked to the other components by clutches which can be closed or open. It allows the assessment of different topologies and operating modes depending on the state of the clutches. However, only a brute force comparison is performed with the components and gear-ratio of the Volt-2 and only some downsizing of the MGs is presented. In [15] a new model of tracked vehicle with three PG is developed and optimized sizing is performed using genetic algorithm (GA) and near optimal control method. In most of these studies, the efficiency of the gears seems not to be taken into account, thus energetic comparison could not be completely fair.

In [18], the automatic generation of graphs of hybrid architectures is proposed using constraint-programming, without an evaluation of the architectures. Automated physical modeling and filtering is then added in [19] without component

technology & sizing optimization, and while neglecting the transmission losses. These works will be recalled later and compared to the proposed methodology.

In this paper, the authors propose a complete methodology allowing the optimal design of HEV architectures from topology level to the optimal sizing and control. Using a certain amount of components, all the possible topologies are generated, filtered and optimally sized (components and gears). An original model allows assessing the fuel consumption while taking into account the efficiency of all the components and without the need to develop a new model when changing the involved components.

iii. Proposed methodology

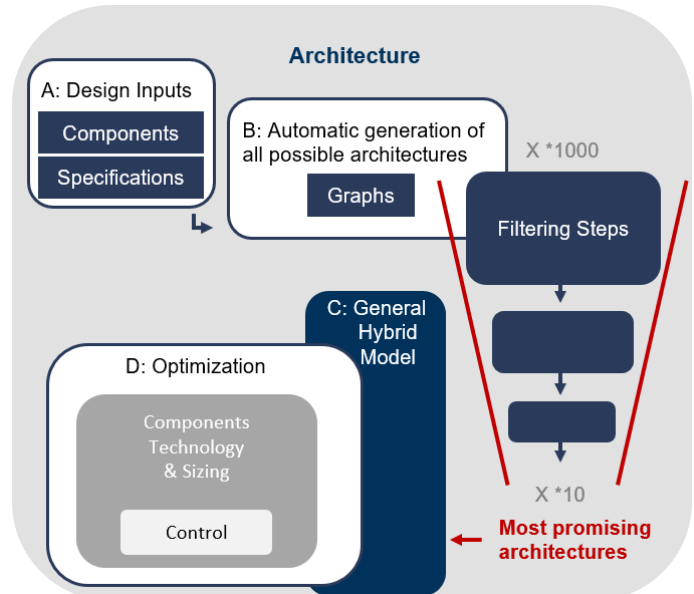


Figure 2: Overview of the proposed methodology

The proposed methodology (Figure 2) starts by a definition of the design inputs (A part in Figure 2): the components that can be used, and the specifications (vehicle requirements, objective functions).

Then an automatic generation of all the possible architectures (B part in Figure 2) is done using constraint-programming. This method is similar to the one used in [18]; however, in this work the problem is differently defined, more conveniently implemented using Python environment, and easily solved (chapter III). In this step, the architectures are represented as graphs using a novel graphical representation (chapter II) suitable for new-trend hybrid gearboxes that cannot be generated and studied using the methodologies in the previously cited works [13][17][18][19]. Thousands of generated graphs are commonly expected. Some filtering steps could be added in order to select tens of graphs which correspond to the most promising architectures. This filtering (chapter IV) is done based on novel representations developed in this work.

The most promising architectures should be evaluated and optimized. A General Hybrid Model (C part in Figure 2) is developed for this purpose. It is inserted inside a bi-level optimization technique (D part in Figure 2) on the levels of

control and component technology & sizing. To solve the linked problem of sizing and control (level 2 and 3, figure 1) the most used method seems to be a joined optimization of component sizes and control. This is often performed using an outer loop of optimization based on stochastic algorithm and an inner control loop which can be based on optimal control as dynamic programming (DP) method or on near optimal control method [15]. For example, in [20] and [21] GA and DP are used. In [22] particle swarm optimization are tried with DP. Other techniques are also tried, using Pontryagin's minimum principle for sizing and control [23] or even gradient based technique for the outer loop [24]. However, due to the non-convexity of the problem, GA associated to DP seems to be the most robust method to tackle the sizing and control problem [20], and performs a fair comparison while guaranteeing the convergence and the optimal control.

The developed General Hybrid Model is inserted inside a bi-level optimization technique (GA and DP) and is able to automatically assess and optimize all the generated architectures, an achievement not yet seen in the literature. The architectures are finally compared based on the chosen objective functions.

iv. Contribution and outline

The contributions of this paper are:

- 1- Exploration of the architecture level instead of selection of few known architectures for comparison. This is done thanks to the automatic generation of architectures.
- 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.
- 3- Combining the exploration of the architecture level with the automatic multi-objective optimization. This was not previously achieved in the literature.

In the following, the representation used to visualize the architectures is revealed in chapter II. The automatic generation of architectures is described in chapter III and the automatic filtering in IV. The automatic optimization done using the general hybrid model is explained in chapter V. Finally, a case study is performed in VI to prove the capability of the methodology before concluding.

II. PROPOSED REPRESENTATION

The methodology proposes to generate all the possible architectures that can be made when connecting a certain number of defined powertrain components and defined connecting elements. For that, these components and generated architectures need to be visualized.

In the literature, two remarkable representations are found in [18], and [25]. Inspired by these two works, a graphical representation is proposed to be used in this article (Figure 3).

i. Overview of the graphical representation

A powertrain is visualized as a graph. This graph is made of nodes and connections. The nodes are of two types (bipartite

graph) [25]: components (circles in colors) and connectors - i.e. shafts - (small circles in black). No direct 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)

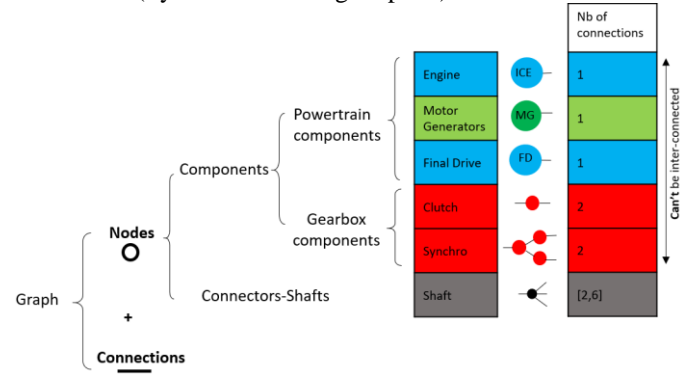


Figure 3: The proposed representation

ii. The 'synchro' unit in details

More details are given here on the synchro unit concept because it gives the representation its uniqueness and appropriate level of details. Such units are seen in manual gearboxes and consist of a synchronizer and 2 pairs of gears (Figure 4). In each pair of gears, one gear (the green) is fixed to a shaft and one (the blue) is free spinning and connects to another shaft through the synchronizer. The synchronizer can be in 3 states: (1) in the center and not connected to any free-spinning gear: position Neutral, (2) to the left and connected to the left gear: connection with gear ratio 1, (3) to the right and connected to the right gear: connection with gear ratio 2.

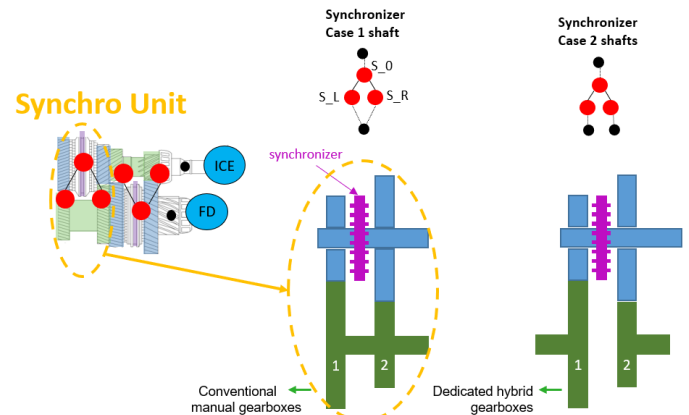


Figure 4: The synchro unit

This 'synchro' unit is visualized as an entity of 3 red nodes (in Figure 4: S_0: central node, S_R: right node, S_L: left node) and has three external connections. One connection is to the input shaft and the two other connections can be to a same output shaft (case of conventional manual gearboxes) or to 2

output shafts (case of dedicated hybrid gearboxes newly seen, as in [26][27][28][29]).

This gives the proposed representation a level of details appropriate to generate these new-trend dedicated hybrid gearboxes that cannot be generated by [18] without making the problem too complex like in [25]. In fact, in [18] the gearbox is considered as one node that can have two connections (1 input, 1 output) and that can be used once (as in conventional powertrains). There is no possibility to innovate inside the gearbox. In [25], all the elementary gearbox components are visualized (the synchronizer and every single gear). This gives a great degree of freedom inside the gearbox, but makes the problem more complex if one is only interested in the energetic aspect of the gearbox and is not in its volumetric and compactness aspect.

iii. Example of architectures

Below are examples on architectures visualized using the proposed representation:

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

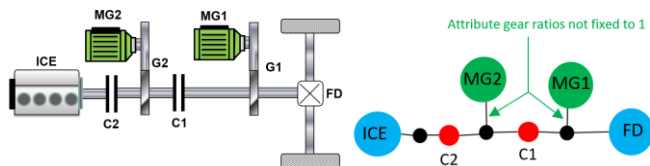


Figure 5: Example 1

Gears are not shown in our representation. However, each component edge and each of the right and left nodes of the ‘synchro’ unit could have an attribute gear ratio.

- A parallel architecture with a 2-speed gearbox:

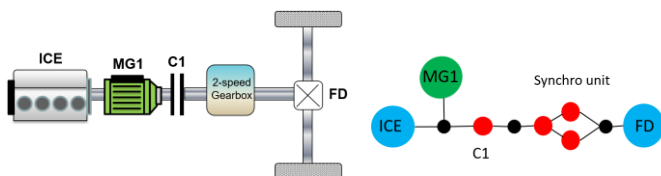


Figure 6: Example 2

It should be mentioned that in the proposed representation no planetary gear component was used at the time, but it can be easily added in the future. The general model (section V.i) will be adapted.

III. AUTOMATIC GENERATION

In this section, the automatic generation of architectures is explained. This involves solving a Constraint Satisfaction Problem (CSP) presented hereafter.

i. *Problem definition*

1) *Variables and domains:*

An architecture is visualized as a graph: a set of nodes V connected by a set of edges E . The graph can be described using an adjacency matrix: n by n matrix where the columns and the

rows are the node V_0, \dots, V_{n-1} , and each cell at index (i, j) takes a 0 or 1 value describing the absence (0) or presence (1) of connection between the node V_i and V_j .

In this work, the powertrain is considered to include a maximum of: 1 ICE, 2 MGs, 1 FD, 4 shafts, 3 clutches, and 3 synchro units. With these components quite all new compact parallel or series-parallel architectures for passenger vehicles can be generated, such as presented in [26] and [27]. Topologies with multiple gearboxes can be generated including architectures where the MGs are connected to some internal gears of the gearbox and not just to the input or the output shafts of the gearbox. Architectures with more components can be encountered in heavy vehicles but are not relevant (cost issue) for passenger vehicles. The method can be adapted to more components but will be more time consuming.

The corresponding adjacency matrix is shown in Figure 7. In blue and green are the powertrain components' nodes (ICE, MGs, FD), in red are the mechanical transmission components' nodes (clutches, synchronizers), and in grey the shafts nodes (X1 to X4).

The matrix has $23 \times 23 = 529$ cells that can take 2 values (0 or 1). The values of the variables are related between themselves through some constraints. The involved CSP can be defined as follows:

- Number of variables: 529
- Domain of all variables: $\{0, 1\}$
- Constraints: see next section
- Target: get all the feasible solutions

The number of solutions without constraints is:

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

2) *Constraints:*

Any solution should respect some constraints regarding the components connection, mechanical feasibility, powertrain functionality and the non-repeatability of the components. A summary of the used constraints is shown in Table 1.

Table 1: Summary of the problem constraints

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

These constraints reduce the design space. Before adding these constraints, the variables are 529 (violet rectangle in Figure 7) and have unknown values. The number of solutions is 2^{529}

The C000 constraint (of Table 1) imposes the matrix to be 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. C001 imposes the nodes to be connected to the shaft nodes only, enforcing the variables of other connections to be 0 (except some values kept 1 to visualize the mandatory connection between the nodes of the synchro unit). This is seen in Figure 7 visualizing how the C000 and C001 are affecting the adjacency matrix. The variables with unknown value are reduced from 529 to 76 and the number of solutions from 2^{529} to 2^{76} . The other constraints are also reducing the design space. They are functionality and non-repeatability constraints (for example: in C008 it is useless to have 2 clutches in parallel). Another case of constraint is C003 which imposes that, in order to forbid direct connection between ICE and FD, the value at index (0, j) and the value at index (3, j) cannot be both equal to 1, for a column index j between 19 and 22.

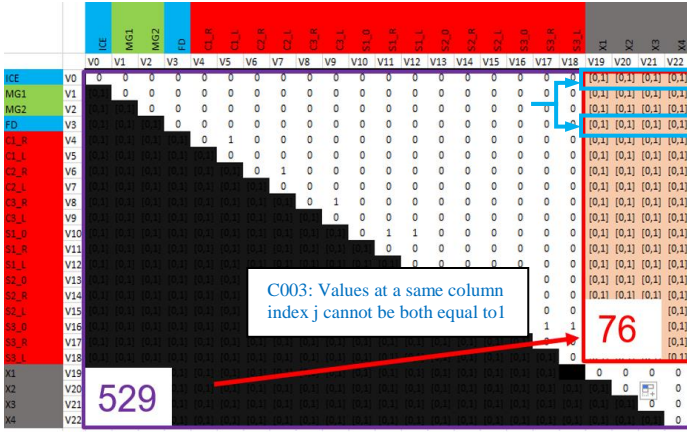


Figure 7: The adjacency matrix after the addition of constraints

ii. Problem implementation and solving

The problem needs to be implemented in an environment and solved. Constraint programming techniques will be used. Python-Constraint module [30] on Python environment was selected. It offers backtracking capabilities and forward checking capabilities (constraint propagation). It succeeded in solving our problem in a relatively fast way. Figure 8 presents an example with a reduced possible number of components.

The output solutions are represented by their adjacency matrices. They need to be visualized according to our proposed representation. For this, the python NetworkX[31] module was used. Between the 132 generated graphs in this example, some are unique but some are redundant. Two examples of redundant graphs are shown in Figure 9. The 2 graphs are the same except a detected interchangeability between the nodes of the synchro unit (S1_R and S1_L).

All the redundant graphs but one need to be removed. This is done in the automatic filtering steps.

IV. AUTOMATIC FILTERING

Even after considering the previously presented constraints (Table 1), the number of possible solutions remains very high and all the generated architectures are not necessarily efficient or different. Therefore, different steps of filtering are needed. They are explained in this chapter.



Figure 8: Example of solving

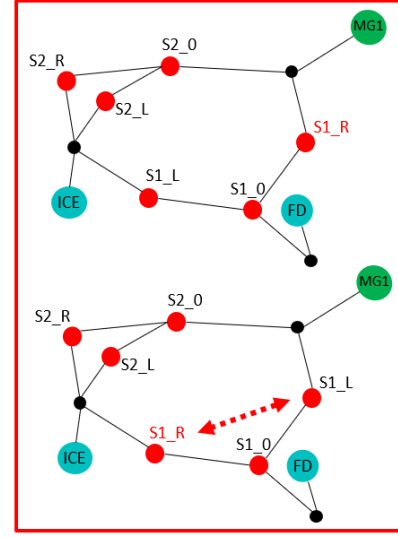


Figure 9: Example of redundant graphs

i. Filtering Step 1: 0ABC Table

Some graphs were found to be redundant or isomorph due to a symmetry problem: orientation of the synchro unit, interchangeability between the nodes of the synchro unit and interchangeability between the electric machines. The 2 graphs in Figure 9 are examples of the interchangeability between the nodes of the synchro unit (S1_R and S1_L). The 2 graphs have the same energetic model leading to same energy efficiency. There is no need to consider both graphs in the methodology and hence one of them is to be removed.

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 exists between the 2 considered components. Figure 10 shows the '0ABC Table' of the redundant graphs of Figure 9. The 1st line of the 0ABC Table indicates that the ICE can be connected to FD through a synchro. The 2nd line: between MG1 and FD, 1 connection through synchro is possible. The 3rd line describes the connection between MG2 and FD, but it is set to 0 since there is no MG2 in this architecture. 4th line: 2 connections through synchro can be done between ICE and MG1. 5th and 6th lines are 0 since MG2 is not present.

Graphs having the same 0ABC table are redundant graphs. The decision is to keep the 1st graph and remove the others.

	0	A	B	C	
ICE <-> FD	0	0	1	0	4 types
MGs <-> FD	0	0	1	0	
ICE <-> MGs	0	0	2	0	
MGs <-> MGs	0	0	0	0	
					0 : no connection
					A : direct connection
					B : connection through synchro
					C : connection through clutch

Figure 10: 0ABC table of the redundant graphs in Figure 9

ii. Filtering Step 2: Modes Table

Each of the kept non-redundant graphs has different states caused by the different combination of states of the actuators. A graph equipped with $N_{clutches}$ and $N_{synchros}$ will have $2^{N_{clutches}} \times 3^{N_{synchros}}$ states. The developed tool on Python generates these states graphs and automatically detects the corresponding powertrain mode, based on the components present and which components are connected. Nine modes are possible:

- (1) ICE only: ICE alone is connected to the wheels
- (2) Pure 1MG: electric mode, only 1 MG connected to the wheels
- (3) Pure 2MG: electric mode, 2 MGs connected to the wheels
- (4) Parallel 1MG: ICE and 1 MG connected to the wheels
- (5) Parallel 2MG: ICE and 2 MGs connected to the wheels
- (6) Series: 1 MG is connected to the wheels and the ICE is connected to another MG
- (7) Neutral: the wheels are connected to none of the components, and no connection between the other powertrain components
- (8) Stand-still charging 1MG: the wheels are connected to none of the components; the engine is however connected to 1 MG
- (9) Stand-still charging 2MG: the wheels are connected to none of the components; the engine is however connected to 2 MGs

These modes were imagined to cover all the possibilities of energy paths. The option of ‘undetected mode’ was added in case a state graph did not fit to the 9 possible modes. However, this case was never encountered proving that the 9 possible modes do cover all the possible modes.

For each graph, a ‘Modes Table’ is created and lists the number of each of the 9 possible modes available in the graph (Figure 11).

Modes Table			Added constraints	
	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

At this level, the graphs can be compared based on their Modes Table. In addition, constraints can be added on the minimum and maximum number of modes required. An example is shown in Figure 11. The Modes Table of the considered architecture of the example has 0 “ICE only” mode. This could be used for example as a constraint and will dismiss this architecture as the minimum number of “ICE only” modes required in this example is 1.

iii. Toward assessment: Modes Table +

More information is needed to assess the architectures and is linked to the efficiency of the powertrain in each of the listed modes. For this reason, a Modes Table + is created (Table 2) and will be used in the assessment (part V.i). For each available

mode, a description of the paths between all the components is done by calculating:

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

Each node in the graph was already assigned an attribute ratio and efficiency. The global gear ratio or efficiency of a path is calculated as a multiplication or division of the attributes of the nodes that are on the path. An example of how some modes are described in Modes Table + is shown in Table 2.

For instance, the ICE only mode (first line of Table 2) is described using 2 values: K_{ICE_FD} and η_{ICE_FD} . K_{ICE_FD} is the global ratio of the ICE to FD path and is calculated as a multiplication of all the ratios of the nodes that exist on this path. η_{ICE_FD} is the global efficiency of the ICE to FD path and is also calculated as a multiplication of the efficiencies of the nodes on the path. For the other modes, all the paths between the involved components are described in a similar way. When only 1 MG is involved in a mode (example Pure 1MG), a variable MG is given a value of 1 or 2 to define which MG is connected to the wheels.

Table 2: Example of modes description in Modes Table +

	Mode	Description
1	ICE only	K_{ICE_FD} , η_{ICE_FD}
2	Pure 1MG	MG = 1 or 2 K_{MG_FD} , η_{MG_FD}
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}

The Modes Table + 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.

V. AUTOMATIC OPTIMIZATION

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 required. 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 modeling all these generated architectures becomes infeasible.

To solve this, automatic modeling [32] or generic transmission models [4] can be used. In this work, a specific general model for (P)HEV is developed.

i. General hybrid model

This model includes the 9 possible modes that the generated architectures can have. A visualization of the model is shown in Figure 12. No real transmission exists in this model but it can represent all the generated architectures. Even if the energy path is not real, it is perfectly equivalent to the “real” generated topologies and allows assessing the vehicle performance or the fuel consumption in an exact manner. All the components are connected by simplified virtual links that have 1 gear with a ratio and efficiency. These are the global ratio and efficiency of the considered power path. This information is found in the Modes Table + of the architecture.

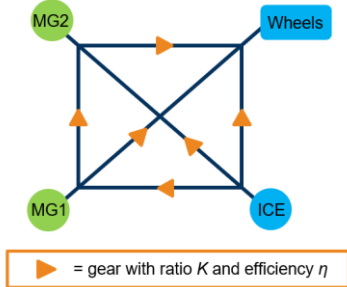


Figure 12: General Hybrid Model

Depending on the selected mode, the energy path is activated or not. Examples of 2 modes are shown in Figure 13. Each of the possible modes in the architecture is a function called using the description of the mode found in Modes Table+.

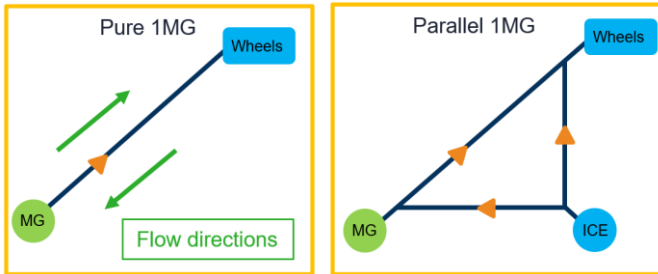


Figure 13: Example of modes inside the general hybrid model

The general hybrid model includes a function for the dynamic performance calculation of the vehicle and a function for the fuel consumption calculation using Dynamic Programming (DP), see part ii. In each of these 2 scripts, the calculation requires an instantaneous calling of all the available modes. All the available modes in Modes Table + are called with the specific information that is also found in Modes Table +.

ii. Bi-level optimization process

The general hybrid model has components & sizing variables that need to be decided when the powertrain is being designed, and control variables that need to be decided instantaneously when the powertrain is being operated. To optimize these variables, the model is inserted inside a bi-level optimization technique following the nested approach [3]: the design (components technology & sizing) in an upper level and the control in a lower level (Figure 14).

1) Upper level design optimization:

A Genetic Algorithm (GA) manages the design level and makes the choice of design candidates to assess (set of values for the design variables). Each design candidate will pass a set of performance tests (for example: acceleration 0 to 100 km/h, acceleration 80 to 120 km/h, and maximum speed). For that, the performance function is called. Inside this function, the vehicle starts from stationary position and at each time step we search for the maximum torque that can be delivered to the wheels. For this, all the available modes are called using their specific information found in Modes Table+. The mode that can deliver the maximum torque on the wheels is selected.

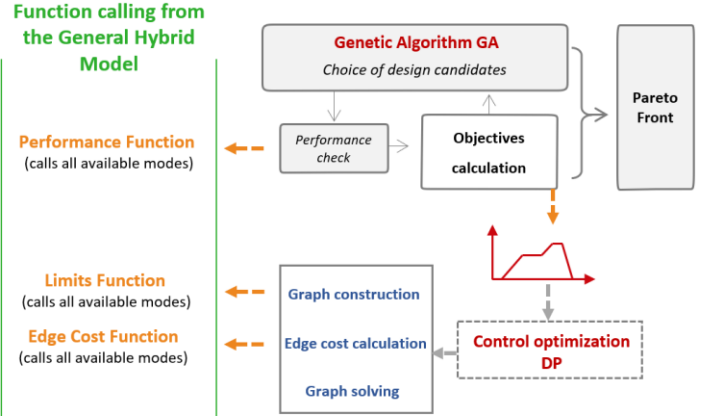


Figure 14: Bi-level optimization process and function calling

For the candidates that pass the performance test, a calculation of the objectives is done. GA iterates this process and performs an exploration of this design space using an evolution process (Selection, Crossover, and Mutation) to produce better generations. This process yields to a Pareto front presenting the tradeoff between the objective functions.

2) Lower level control optimization:

A calculation of the objectives is needed on the upper level of design optimization. When the objective function is only design related (powertrain cost, volume, sizing), no vehicle operation is needed and no control optimization is needed. However, in the case of an objective like the fuel consumption of the vehicle, the vehicle should be simulated and its control is to be optimized.

The vehicle is considered to operate on a driving cycle and Dynamic Programming DP [33][34] solves the control problem. This problem can be represented as a graph of battery State Of Charge SOC versus time (the SOC being the only state variable in the problem). The objective is to go from an initial SOC to a same final SOC (charge sustaining) while choosing the control that ensures the minimum cumulative fuel consumption at the end of the driving cycle. DP solves this problem in 3 steps: graph construction (space limit of the problem), edge cost calculation and finally graph solving. For the first 2 steps, the limits function and the edge cost function of the general hybrid model are called. The final SOC constraint is ensured by the construction of the searching space. The instantaneous fuel consumption cost of the edges is assessed by means of the general model presented (part i), and DP finds the

optimal control to minimize the global fuel consumption on the cycle. For more information on DP applied to HEVs see for example [6]. Inside these functions, all the available modes are to be called instantaneously. Based on Modes Table +, all the available modes are called using their specific information. Unlike conventional ways, DP is operating here with no previous knowledge of the architecture.

The GA, DP and all the scripts are fixed. Changing Modes Table + and repeating the process is enough to simulate and optimize all the generated architectures.

VI. CASE STUDY

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

i. Input components and generated graphs

The starting components that are injected in the automatic generation tool are the components of a series-parallel architecture that has a 2-speed gearbox (Figure 15): 1 ICE, 1 FD, 2 MGs, 4 shafts (black dots), 2 clutches (red dots) and 1 synchro unit. We will be interested in architectures having at least 1 ICE only, 1 electric, 1 series and 1 parallel mode. In addition, the architectures should allow battery charging at standstill (cf. VI.ii).

The tool generated 480 graphs of architectures in 807.1 seconds. The eleven architectures finally selected (see below) are presented in Figure 16.

ii. Automatic filtering and selection of architectures

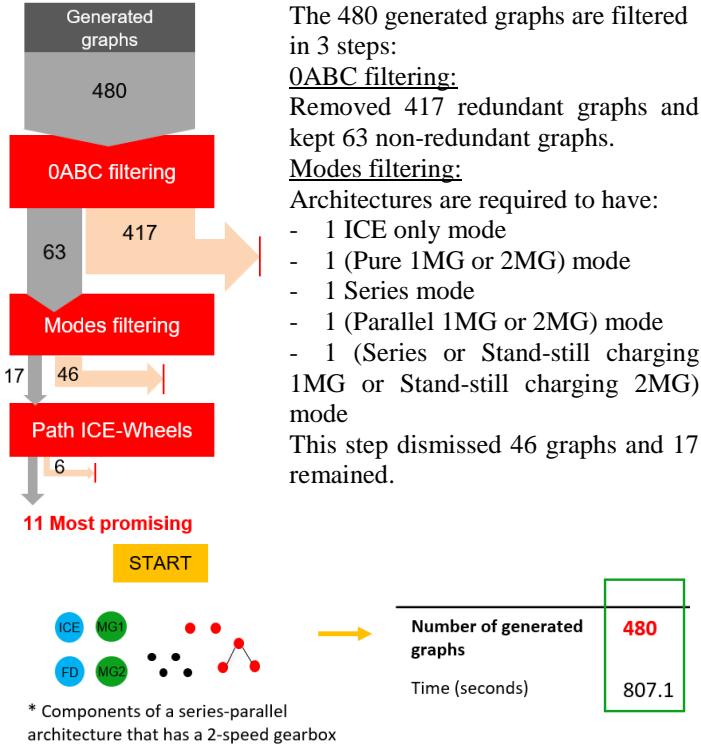


Figure 15: Input components and solving

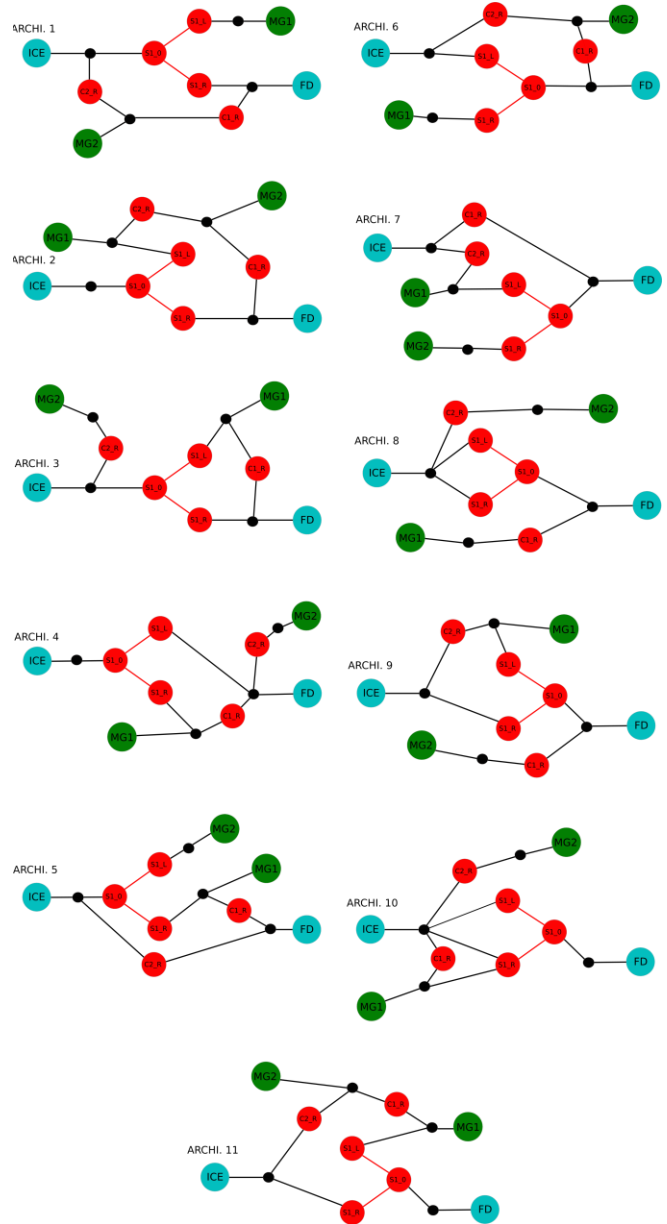


Figure 16: The 11 selected architectures

Path ICE-Wheels:

For dynamic and efficiency reasons, architectures are required to have a minimum of 2 paths from the ICE to the wheels. 6 graphs are dismissed.

At the end of the filtering steps, 11 graphs of architectures remained (Figure 16). These are the most promising architectures that are optimized and compared in the next section.

iii. Automatic assessment and comparison

The Modes Tables + are passed to the general hybrid model.

1) Simulation parameters

The components used correspond to a middle class HEV [7]. The components are sized through a scaling technique [7][35]. Their initial main characteristics are presented Table 3. The engine specific fuel consumption and MGs efficiencies maps can be found in [7].

Table 3: Main initial characteristics of the components

Vehicle Weight		1360 kg
Engine	weight	100 kg
	max power	57kW @ 5000rpm
MG1 and MG2	weight	55 kg
	max power	50 kW
	max torque	400Nm @ 1200 rpm
Battery	Module weight	1.75 kg
	Module energy	46.7 Wh

The sizing variables are scaling factors applied to the maximum power of the ICE, MG1 and MG2, the modules number in the battery and all the involved gear ratios (6). The scaling factor also impacts the weight of the components (with the assumption of weight proportional to maximum torque). The initial weights are presented in Table 3. The battery is modeled by a classical equivalent electric serial model: open circuit voltage in series with a resistance. The resistance value changes depending on charge or discharge operations.

Two objective functions are considered:

- (1) Number of battery modules in series (N_{batt}),
- (2) Charge sustaining fuel consumption:

$$FCons = \alpha.FCons_{urban} + \beta.FCons_{rural} + \gamma.FCons_{highway} \quad (1)$$

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 [36].

The fuel consumption will be a good indication of the CO₂ emission. Further works can also consider other pollutants emissions (CO, NO_x, HC). However it will complicate the control problem if a tradeoff between CO₂ and pollutant is also considered as for example in [37]. The battery number is a good indicator of the onboard electrical power and energy and of the cost of the hybridization. A global cost can also be considered [38][15] even if it is difficult to evaluate with accuracy.

The multi-objectives NSGA-II algorithm will then provide the Pareto front of the optimization problem formulated as follows:

$$\begin{aligned} & \begin{cases} \min(FCons) \\ \min(Nbatt) \end{cases} \\ & \text{subject to the following constraint} \\ & C1: t_{0 \rightarrow 100} < 10.1 \text{ s;} \\ & C2: t_{80 \rightarrow 120} < 7.5 \text{ s;} \\ & C3: V_{max} > 180 \text{ km/h} \end{aligned}$$

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 are used to simulate these three conditions and the control is optimized on each of them using DP. One DP problem is solved (part V.ii.2) for each cycle of FCons.

2) Results

The 11 architectures are optimized. The Pareto fronts representing the performance of each architecture according to the 2 objectives are shown in Figure 17.

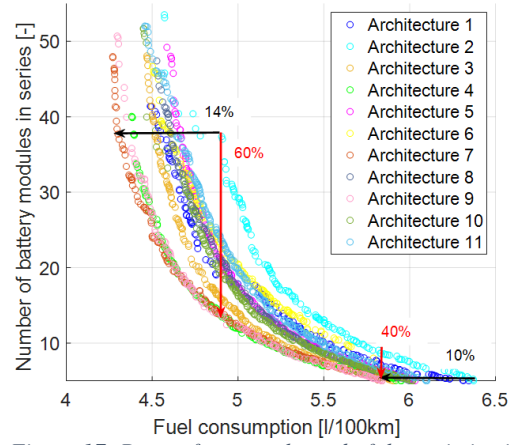


Figure 17: Pareto fronts at the end of the optimization

Those results prove that the methodology succeeded in:

- 1- Automatically generating all the graphs of possible architectures.
- 2- Automatically filtering these architectures and selecting some of them for optimization and assessment.
- 3- Automatically optimizing these architectures and comparing them based on the 2 chosen objective functions. The Pareto fronts shown in Figure 17 prove that for a same number of battery modules, the architectures can have a 10% to 14% difference in fuel consumption. For a same fuel consumption, the architectures can have 40% to 60% difference in the number of battery modules in series.
- 4- Another output of the methodology is the possibility of examining closely the optimal sizing and the powertrain operation (energy management, components operation, modes choice...) for all the architectures on all the simulated driving cycles. An example is presented hereafter.

	ARCH 9	ARCH 1	ARCH 2
Nb Battery Modules	28		
FC MIXED (l/100km)	4.49	4.69	5.07
urban	4.39	4.65	5.15
rural	3.67	3.81	4.19
highway	5.44	5.64	5.86
Power MG1 (kW)	86.6	50.5	98.8
Power MG2 (kW)	100	29	99
Power ICE (kW)	64.2	88.9	69.9
k_FD	3.4	3.95	1.23
k_ICE	1	1.2	2.55
k_MG1	1	2.8	2.94
k_MG2	1.11	1	2.86
S_L	1.49	0.8	1.52
S_R	0.73	0.46	0.63

Table 4: The sizing of the Pareto points for 28 battery modules

The Pareto points corresponding to 28 Number of battery modules for architectures 9 (one of the best), 1 (intermediate) and 2 (worse) are considered in this example. The sizing is presented in Table 4, along with the fuel consumption in each of the driving conditions.

Architecture 9 has the least fuel consumption in all driving conditions. It has also downsized MG1 and ICE compared to architecture 2. However, architecture 1 presents the most downsized electric powertrain components (MG1 and MG2), with a slightly upsized engine.

The operation of architecture 9 on the rural driving cycle is examined in Figure 18 and Figure 19. The charge sustaining operation is evident and frequent engine power cut is visible.

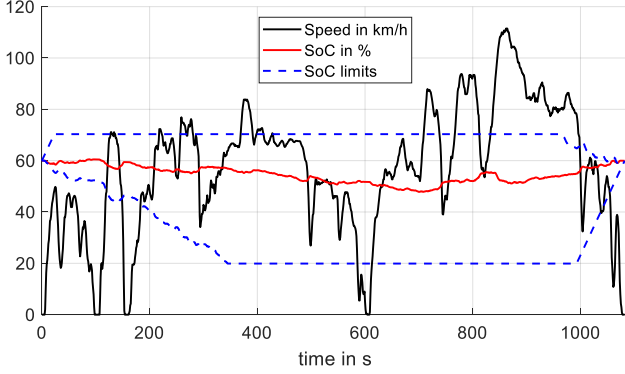


Figure 18: SOC variation in rural driving

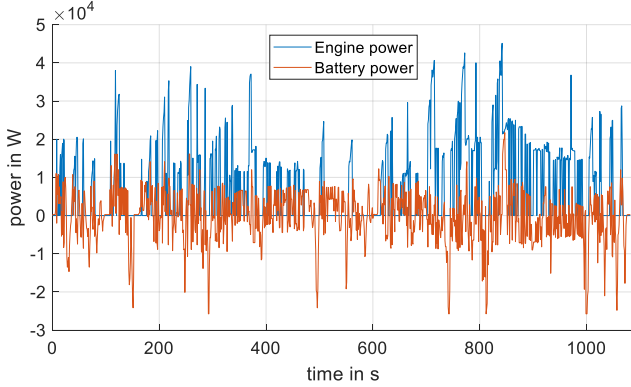


Figure 19: Power variation in rural driving

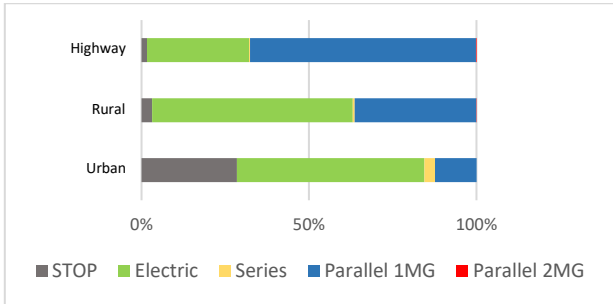


Figure 20: Mode choice in percentage of time in urban, rural and highway driving

The percentage of time choice of each mode in each driving condition is presented in Figure 20. The electric driving has the major shares in urban and rural, with less occurrence in highway. The parallel 1MG modes usage increases from urban to rural and highway. The parallel 2MG modes are absent in urban and minorly used in rural and highway. The series mode is also minorly used in all the driving conditions. The importance of having a series mode is to be questioned in later analysis.

VII. CONCLUSION

The paper proposes a systematic methodology for the generation and design of hybrid powertrains. The representation used and the different parts of the methodology were explained: the automatic generation of architectures, the automatic filtering and the automatic optimization. A case study was also done and proved the capability of the methodology. The main contributions of this work are (1) the developed tool that can generate and filter hybrid architectures based on the manufacturer constraints and requirements, and (2) the general hybrid model that can simulate and optimize all the generated architectures. The main limitations are the exclusion of power-split architectures and the optimization time for 1 architecture (8 to 10 hours). These issues will be addressed in the future. Improving the codes has already started and could reduce the optimization time by half. The component models, sizing technique and choice will also be reviewed and improved. Other objectives will be also considered (compactness, pollutants, battery aging...). An example of comprehensive analysis of the results (optimal sizing, powertrain operation, modes choice) has been given in this article, but more deep analysis will be carried out in the future in order to assess in detail the effectiveness of the best proposed solutions.

ACKNOWLEDGMENT

The authors acknowledge the contribution of Yuan CHENG (chengyuan.hit@gmail.com) to this work when he was with Groupe PSA.

REFERENCES

- [1] European Commission, "Europe accelerates the transition to clean mobility: Co-legislators agree on strong rules for the modernisation of the mobility sector," 2018. https://ec.europa.eu/clima/news/europe-accelerates-transition-clean-mobility-co-legislators-agree-strong-rules-modernisation_en (accessed Jan. 15, 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, doi: 10.1109/TVT.2009.2033605.
- [3] 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, doi: 10.1109/TVT.2016.2547897.
- [4] 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, doi: 10.1109/TVT.2018.2834537.
- [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] 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, Jan. 2014, doi: 10.1109/TVT.2013.2271646.
- [7] B. Kaban, 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, doi: 10.1109/TVT.2019.2952190.
- [8] 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, doi: 10.1109/TVT.2012.2196299.
- [9] 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, doi: 10.1109/TVT.2012.2208210.

- [10] 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, doi: 10.3390/en11113185.
- [11] 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*, Apr. 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/VPPC.2017.8330989.
- [12] K. Chen, A. Bouscayrol, A. Berthon, P. Delarue, D. Hissel, and R. Trigui, “Global modeling of different vehicles,” *Veh. Technol. Mag. IEEE*, vol. 4, no. 2, pp. 80–89, 2009.
- [13] 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, doi: 10.1109/TVT.2016.2531740.
- [14] S. E. L. Xiaowu Zhang, Huei Peng, Jing Sun, 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, doi: 10.1115/DSCC2014-6028.
- [15] Z. Qin, Y. Luo, W. Zhuang, Z. Pan, K. Li, and H. Peng, “Simultaneous optimization of topology, control and size for multi-mode hybrid tracked vehicles,” *Appl. Energy*, vol. 212, pp. 1627–1641, Feb. 2018, doi: 10.1016/j.apenergy.2017.12.081.
- [16] N. Stroe, G. Colin, K. Ben-Cherif, S. Olaru, and Y. Chamaillard, “Towards a generic control-oriented model for HEV predictive energy management,” *IFAC-PapersOnLine*, vol. 49, no. 11, pp. 259–264, Jan. 2016, doi: 10.1016/j.ifacol.2016.08.039.
- [17] 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.
- [18] 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, doi: 10.1109/TMECH.2015.2405473.
- [19] 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, doi: 10.1109/TVT.2018.2844024.
- [20] 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.
- [21] Y. Zou, F. Sun, X. Hu, L. Guzzella, and H. Peng, “Combined optimal sizing and control for a hybrid tracked vehicle,” *Energies*, 2012, doi: 10.3390/en5114697.
- [22] 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*, Sep. 2015, vol. 28, no. 15, pp. 16–22, doi: 10.1016/j.ifacol.2015.10.003.
- [23] Z. Song, X. Zhang, J. Li, H. Hofmann, M. Ouyang, and J. Du, “Component sizing optimization of plug-in hybrid electric vehicles with the hybrid energy storage system,” *Energy*, vol. 144, pp. 393–403, Feb. 2018, doi: 10.1016/j.energy.2017.12.009.
- [24] M. Poline, L. Gerbaud, J. Pouget, and F. Chauvet, “Simultaneous optimization of sizing and energy management—Application to hybrid train,” *Math. Comput. Simul.*, vol. 158, pp. 355–374, Apr. 2019, doi: 10.1016/j.matcom.2018.09.021.
- [25] 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.
- [26] N. Fremau, A. Villeneuve, A. Ketfi-Cherif, and A. Vignon, “Transmission pour vehicule automobile a propulsion hybride et procédé de commande associé,” WO2015107275A1, 2015.
- [27] N. Fremau, A. Villeneuve, A. Ketfi-Cherif, and A. Vignon, “Hybrid transmission with offset electric machine and method for controlling gear changes,” WO2015197927A1, 2015.
- [28] Renault, “E-TECH: la voiture hybride adaptée à tous.”

<https://www.renault.fr/decouvrez-renault/easylife/vehicules-hybrides-e-tech.html> (accessed Jan. 06, 2020).

- [29] M. Nakagawa, U. Schwarz, and J. Andert, “DENSO’s Contribution for the Future E-mobility Through New Technology (Innovative approach for development),” 2013.
- [30] G. Niemeyer, “Python-Constraint,” *Constraint Solving Problem resolver for Python*. <https://labix.org/python-constraint>.
- [31] A. Hagberg, D. Schult, and P. Swart, “Exploring network structure, dynamics, and function using NetworkX,” 2008, Accessed: Nov. 07, 2019. [Online]. Available: https://www.researchgate.net/publication/242724947_Exploring_net_work_structure_dynamics_and_function.
- [32] J. Goos, C. Criens, and M. Witters, “Automatic Evaluation and Optimization of Generic Hybrid Vehicle Topologies using Dynamic Programming,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10065–10071, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1778.
- [33] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
- [34] 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.
- [35] 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.
- [36] Eurostat, “Eurostat statistic on population displacement.” [Online]. Available: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=road_tf_road&lang=fr.
- [37] A. Guille des Buttes, B. Jeanneret, A. Kéromnès, L. Le Moyne, and S. Péliissier, “Energy management strategy to reduce pollutant emissions during the catalyst light-off of parallel hybrid vehicles,” *Appl. Energy*, vol. 266, 2020, doi: 10.1016/j.apenergy.2020.114866.
- [38] 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.



B. Kabalan received his B.E. in Mechanical Engineering from the Lebanese American University, Lebanon in 2015. He received a M.S. in Powertrain Engineering from IFP School, France in 2017.

In March 2020 he finished a PhD that took place in the Advanced Research Department of Groupe PSA and in IFSTTAR. He then joined IFSTTAR as a researcher. His work is focused on developing optimal methodologies for the design of hybrid electric vehicle powertrains.



E. Vinot aged 48, received the engineering degree from the Grenoble-INP Institute (1997), the M.S. degree in electrical engineering from the Laval University, Québec, Canada (1998) and the Ph. D degree from the Grenoble-INP Institute in 2000. Currently he is working in Gustave Eiffel University in ECO7 (ECO-gestion des Systèmes Énergétiques Pour les Transports) team. His main interests are systemic model of vehicle and components, system management optimisation, and system and electrical machine design.



R. Trigui received the PhD in Electrical Engineering in 1997. Since 1998, he is full researcher (senior since 2012) in the French Institute of Science and Technology for Transport, Development and Networks (IFSTTAR) in the field of modelling and energy management of electric and hybrid vehicles. He is now

deputy head of AME department of Gustave Eiffel University (former IFSTTAR). He is member of the French MEGEVH network. He was co-chair of IEEE VPPC 2010.



C. Dumand received his Engineering degree of Polytech'Orléans (2002) and the PhD degree from ENSMA of Poitiers, France, in 2005.

Currently, he is team leader in the Advanced Research Department of Groupe PSA. His team is in charge of advanced research in powertrain, new energy converter and energy storage technologies,

new energy pathways and advanced Computer Aided Design using Artificial Intelligence and automatic optimization.